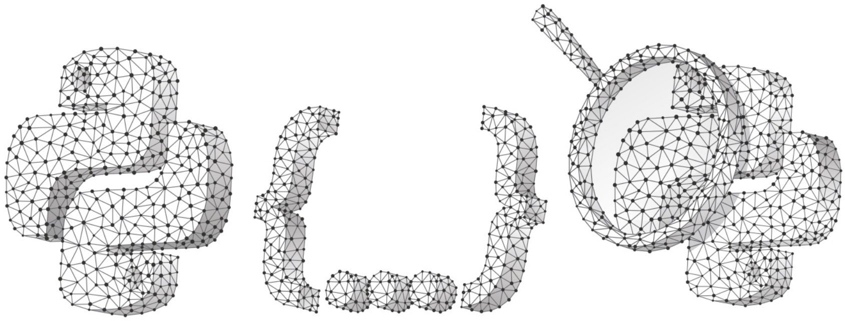
INTERNET OF THINGS AND MACHINE LEARNING

Machine Learning Algorithms Using Python Programming



Gopal Sakarkar • Gaurav Patil • Prateek Dutta

NOVA

INTERNET OF THINGS AND MACHINE LEARNING

MACHINE LEARNING ALGORITHMS USING PYTHON PROGRAMMING

INTERNET OF THINGS AND MACHINE LEARNING

Additional books and e-books in this series can be found
on Nova’s website under the Series tab.

INTERNET OF THINGS AND MACHINE LEARNING

MACHINE LEARNING ALGORITHMS USING PYTHON PROGRAMMING

GOPAL SAKARKAR
GAURAV PATIL
AND
PRATEEK DUTTA

nova
science publishers
New York

NOTICE TO THE READER

The Publisher has taken reasonable care in the preparation of this book, but makes no expressed or implied warranty of any kind and assumes no responsibility for any errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of information contained in this book. The Publisher shall not be liable for any special, consequential, or exemplary damages resulting, in whole or in part, from the readers' use of, or reliance upon, this material. Any parts of this book based on government reports are so indicated and copyright is claimed for those parts to the extent applicable to compilations of such works.

Independent verification should be sought for any data, advice or recommendations contained in this book. In addition, no responsibility is assumed by the Publisher for any injury and/or damage to persons or property arising from any methods, products, instructions, ideas or otherwise contained in this publication.

This publication is designed to provide accurate and authoritative information with regard to the subject matter covered herein. It is sold with the clear understanding that the Publisher is not engaged in rendering legal or any other professional services. If legal or any other expert assistance is required, the services of a competent person should be sought. FROM A DECLARATION OF PARTICIPANTS JOINTLY ADOPTED BY A COMMITTEE OF THE AMERICAN BAR ASSOCIATION AND A COMMITTEE OF PUBLISHERS.

Additional color graphics may be available in the e-book version of this book.

Library of Congress Cataloging-in-Publication Data

ISBN: 978-1-53619-615-3 (eBook)

Published by Nova Science Publishers, Inc. † New York

“Artificial intelligence will be the ultimate version of Google. The ultimate search engine that would understand everything on the web. It would understand exactly what you wanted, and it would give you the right thing. We’re nowhere near doing that now. However, we can get incrementally closer to that, and that is basically what we work on.”

-*Larry Page (Co-founder and CEO, Alphabet Inc. ‘Google’)*

“I’m increasingly inclined to think that there should be some regulatory oversight, maybe at the national and international level, just to make sure that we don’t do something very foolish. I mean with the artificial intelligence we are summoning the demon.”

-*Elon Musk (South African inventor, investor, CEO & CTO of SpaceX, CEO of Tesla Inc.)*

CONTENTS

PREFACE

The machine learning field is concerned with the question of how to create computer programs that automatically improve information. In recent years many successful electronic learning applications have been made, from data mining systems that learn to detect fraudulent credit card transactions, filtering programs that learn user readings, to private cars that learn to drive on public highways. At the same time, there have been significant developments in the concepts and algorithms that form the basis for this field. Machine learning is programming computers to optimize a performance criterion using example data or past experience.

The goal of this textbook is to present the key concepts of Machine Learning which includes Python concepts and Interpreter, Foundation of Machine Learning, Data Pre-processing, Supervised Machine Learning, Unsupervised Machine Learning, Reinforcement Learning, Kernel Machine, Design & analysis of Machine Learning experiment and Data visualization. We are covering the theoretical concepts along with coding implementation. This book aims to pursue a middle ground between a theoretical textbook and one that focuses on applications. The book concentrates on the important ideas in machine learning.

Given the wide variety of features, this book makes a few thoughts about the student’s background. Instead, it introduces basic concepts from mathematics, artificial intelligence, the concept of knowledge, and other areas where necessary, focusing on those concepts that are most relevant to machine learning. This book is designed for undergraduate and graduate students in fields such as computer science, engineering, mathematics and social sciences, and as a reference for software experts and operators.

The principle behind the writing of this book is that we should introduce doctrinal equality and practice. The study of machine learning attempts to answer questions such as "How does learning performance differ from the number of training examples provided?" and "Which learning algorithms are most suitable for different types of learning activities?" The practice of machine learning is overshadowed by introducing major algorithms in the field, as well as clues that demonstrate their effectiveness.

Chapter 1

PYTHON CONCEPT AND INTERPRETER

1.1. PYTHON

1.1.1. What Is Python

Python is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library. Two major versions of Python are currently in active use: Python 3.x is the current version and is under active development. Python 2.x is the legacy version and will receive only security updates until 2020. No new features will be implemented. Note that many projects still use Python 2, although migrating to Python 3 is getting easier.

1.1.2. Installation of Python

On Windows

If you are using Windows, you will probably have to install Python and configure certain settings correctly before you can get to grips with the examples in this book. For that, you will need to refer to the specific instructions for your operating system on the following Python web pages:

- http://wiki.python.org/moin/BeginnersGuide/Download
- http://www.python.org/doc/faq/windows/
- http://docs.python.org/dev/3.0/using/windows.html

First, you need to download the official installer; alternative versions for Itanium and AMD machines are available from http://www.python.org/download/. You should save this file, which will have a .msi extension, somewhere you'll be able to find again easily. You can then double-click this file to start the Python installation wizard, which will take you through the installation. It's fine to accept the default settings if you're not sure of any answer.

Installing on Other Systems

You may choose to install Python on other systems, particularly if you want to take advantage of newer versions. For Linux and other Unix-like systems, the installation instructions are here:

- http://docs.python.org/dev/3.0/using/unix.html

If you're using OS X, your instructions are here:

- http://www.python.org/download/mac/
- http://docs.python.org/dev/3.0/using/mac.html

Choosing the Right Python Version

You will find the different installers include a number after the word Python. This is the version number. When I started writing this book, those numbers ranged from 2.3.7 (old but still usable) through 2.5.2 (the previous stable version) to 3.0 (the new version about to be released). At the same time as version 3.0 was released, the Python team also put out version 2.6, which is an upgraded version of Python version 2 for people who want (or need) to stick with the old way of doing things but still want to benefit from general fixes and some of the new features introduced in version 3.0. The Python language is continuously evolving; version 3.0 has become the norm and has evolved into version 3.1.1. The new version, which I'll refer to as version 3.0 because all 3.x versions are simply refinements on the original plan of 3.0, includes several changes to the programming language that are incompatible with version 2.x (x is any number you like), which I will refer to in the rest of this book as the old version. Most of the language is the same, however, so the differences between the versions of Python will be noted in the text as those subjects are covered. Examples in this book are for Python 3.0 except where noted. There may be some differences running Python on other operating systems, which I will do my best to point out where relevant. Otherwise, the examples of code will work the same way. This is one of the many good points of Python. For the most part, this book will concentrate on the fun part—learning how to write programs using Python. The official Python documentation is plentiful, free, and well written, and you should read it alongside this book. It is available on at http://www. python.org/doc/.

1.2. INTERPRETER

1.2.1. IDLE

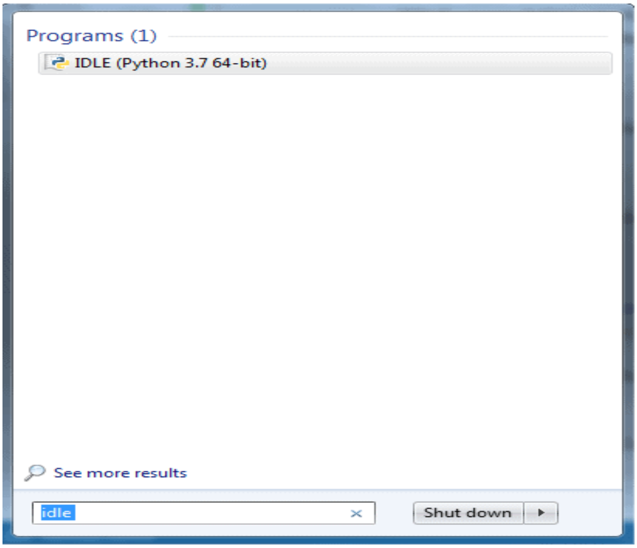
What Is IDLE?

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default.

IDLE can be used to execute a single statement just like Python Shell and also to create, modify and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts that includes features like syntax highlighting, autocompletion and smart indent. It also has a debugger with stepping and breakpoints features.

How to Use IDLE?

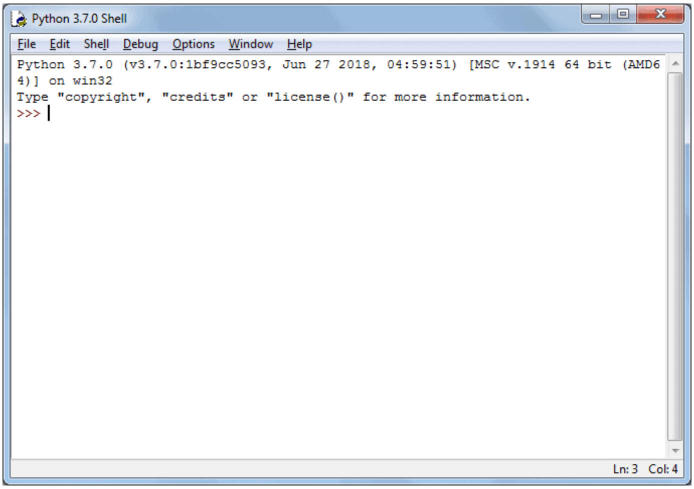
- To start IDLE interactive shell, search for the IDLE icon in the start menu and double click on it.



idle

Figure 1.1. Get start with IDLE.

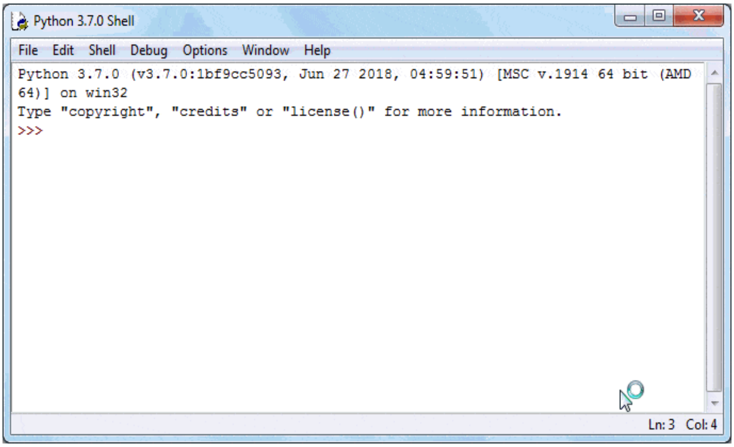
- This will open IDLE, where you can write Python code and execute it as shown below.



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD6
4)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> |
```

Figure 1.2. IDLE interface.

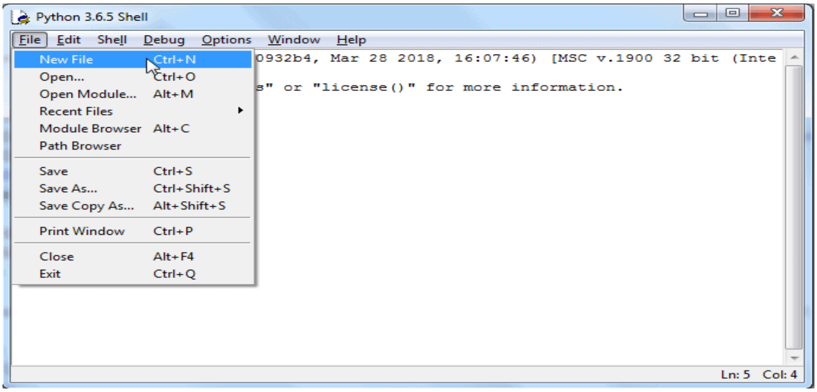
- Now, you can execute Python statements the same as in Python Shell as shown below.



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD
64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Figure 1.3. IDLE Environment.

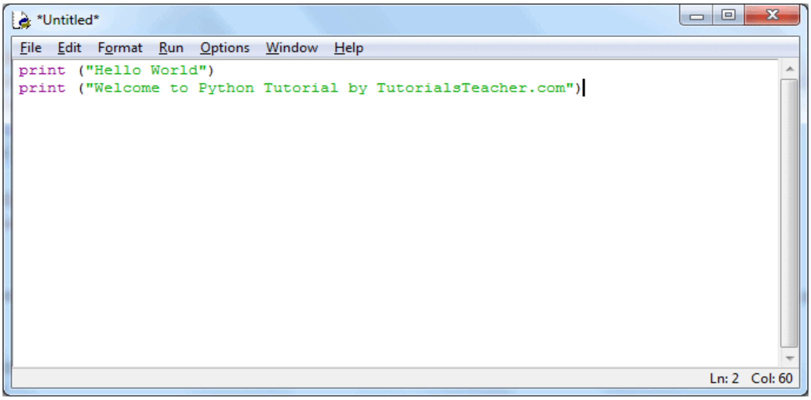
- To execute a Python script, create a new file by selecting File -> New File from the menu.



```
0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Inte

s" or "license()" for more information.
```

Figure 1.4. New file creation in IDLE.

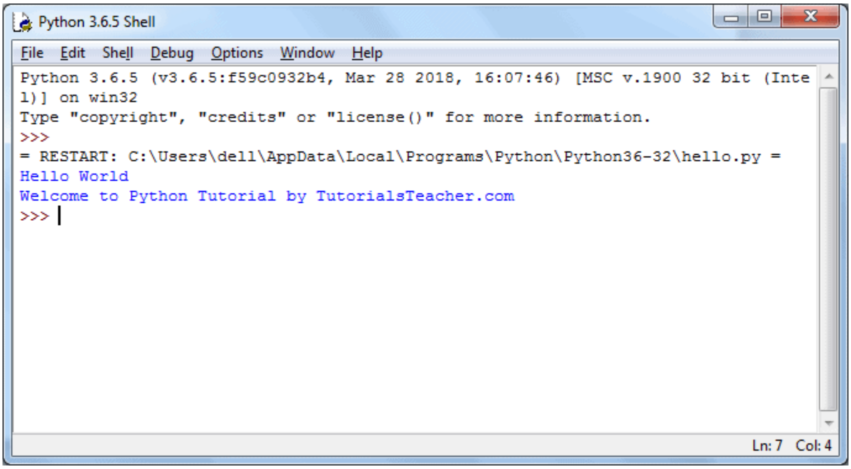
- Enter multiple statements and save the file with extension .py using File -> Save. For example, save the following code as hello.py



```
print ("Hello World")
print ("Welcome to Python Tutorial by TutorialsTeacher.com")
```

Figure 1.5. Program execution in IDLE.

- Now, press F5 to run the script in the editor window. The IDLE shell will show the output.



```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Inte
l)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\AppData\Local\Programs\Python\Python36-32\hello.py =
Hello World
Welcome to Python Tutorial by TutorialsTeacher.com
>>> |
```

Figure 1.6. Output interface of IDLE.

- Python Script Execution Result in IDLE.

Thus, it is easy to write, test and run Python scripts in IDLE.

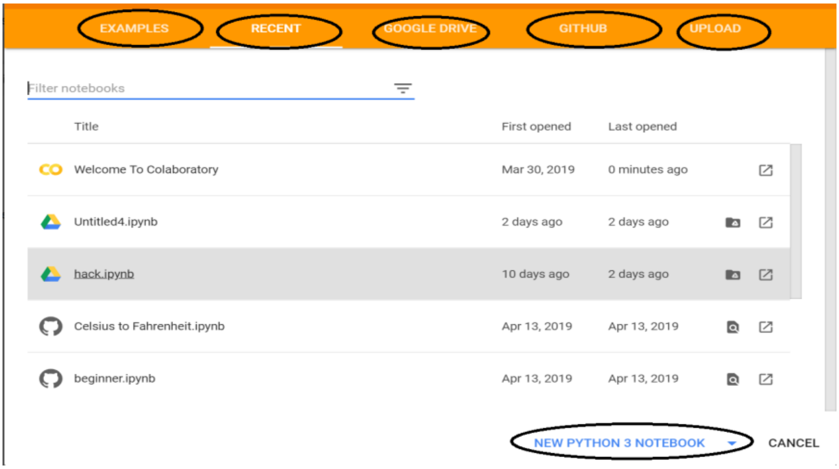
1.2.2. Google Colab

Colab is a free notebook environment that runs entirely in the cloud. It lets you and your team members edit documents, the way you work with Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

How to Use Google Colab?

To start working with Colab you first need to log in to your google account, then go to this link https://colab.research.google.com.

On opening the website you will see a pop-up containing following tabs –



Title	First opened	Last opened
Welcome To Colaboratory	Mar 30, 2019	0 minutes ago
Untitled4.ipynb	2 days ago	2 days ago
hack.ipynb	10 days ago	2 days ago
Celsius to Fahrenheit.ipynb	Apr 13, 2019	Apr 13, 2019
beginner.ipynb	Apr 13, 2019	Apr 13, 2019

Figure 1.7. Getting start with colab

- EXAMPLES: Contain a number of Jupyter notebooks of various examples.
- RECENT: Jupyter notebook you have recently worked with.
- GOOGLE DRIVE: Jupyter notebook in your google drive.
- GITHUB: You can add Jupyter notebook from your GitHub but you first need to connect Colab with GitHub.
- UPLOAD: Upload from your local directory.

Else you can *create a new Jupyter notebook* by clicking New Python3 Notebook or New Python2 Notebook at the bottom right corner.

Notebook's Description

On creating a new notebook, it will create a Jupyter notebook with Untitled.ipynb and save it to your google drive in a folder named Colab Notebooks. Now as it is essentially a Jupyter notebook, all commands of Jupyter notebooks will work here. Though, you can refer to the details in Getting started with Jupyter Notebook.

Type the code in the cell and click on the run button (ctrl + enter) located at the left of the cell.

As soon as running is completed your resultant outcome will get displayed at the bottom of the cell.

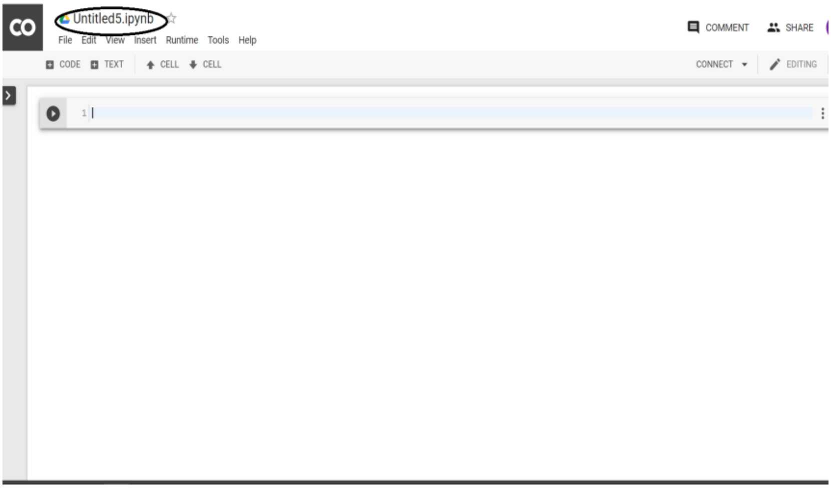


Figure 1.8. Colab Program interface.

1.2.3. Jupyter

What Is Jupyter Notebook?

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

How to Install Jupyter Notebook?

Jupyter Notebook can be installed by using either of the two ways described below:

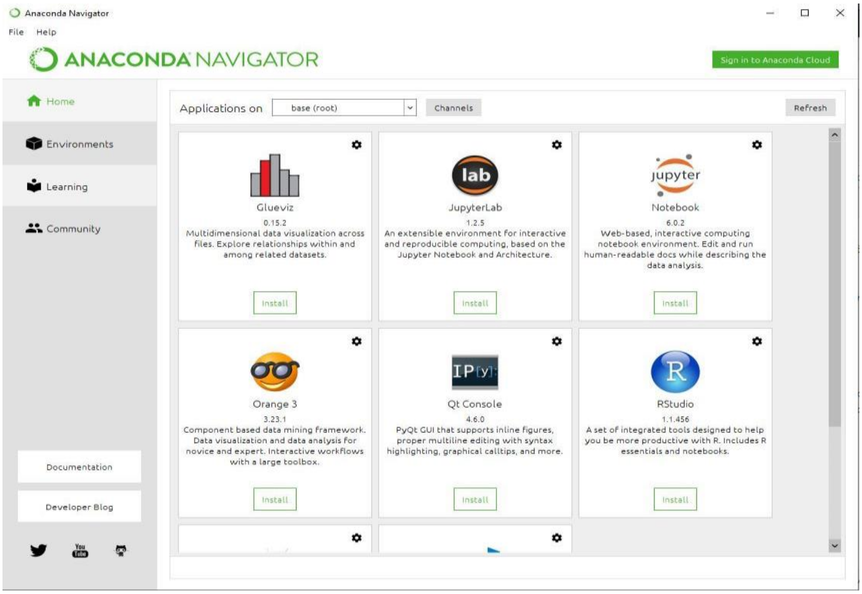
- Using Anaconda: Install Python and Jupyter using the Anaconda Distribution, which includes Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science. To install Anaconda, go through How to install Anaconda on windows? and follow the instructions provided.

- Using PIP: Install Jupyter using the PIP package manager used to install and manage software packages/libraries written in Python. To install pip, go through How to install PIP on Windows? and follow the instructions provided.

Installing Jupyter Notebook Using Anaconda

Anaconda is an open-source software that contains Jupyter, spyder, etc that are used for large data processing, data analytics, heavy scientific computing. Anaconda works for R and python programming language. Spyder (sub-application of Anaconda) is used for python. Opencv for python will work in Spyder. Package versions are managed by the package management system called conda. To install Jupyter using Anaconda, just go through the following instructions:

- Launch Anaconda Navigator:



Glueviz
0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

JupyterLab
1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Notebook
6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

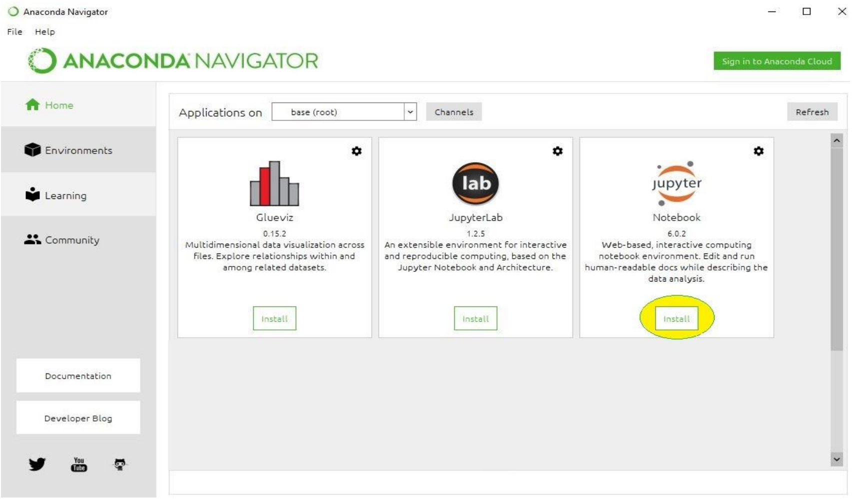
Orange 3
3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Qt Console
4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Figure 1.9. Get start with Jupyter.

- Click on the Install Jupyter Notebook Button:



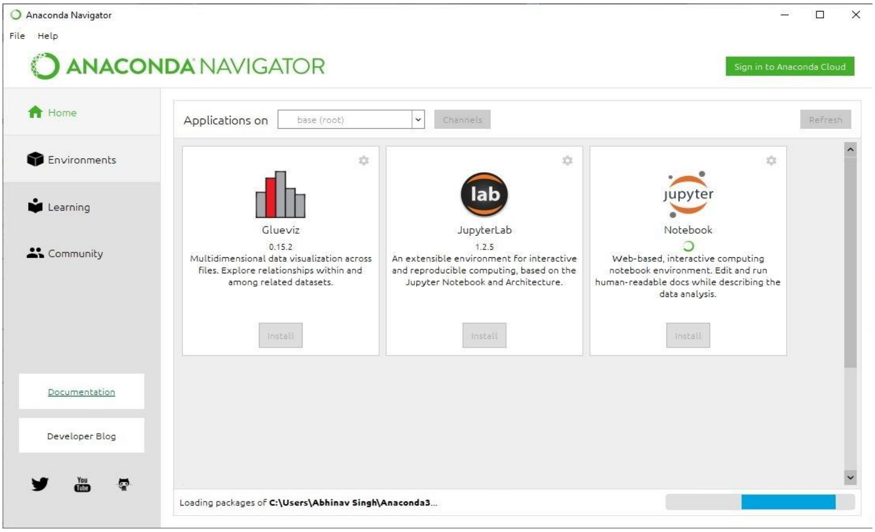
Multidimensional data visualization across files. Explore relationships within and among related datasets.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Figure 1.10. Install Jupyter.

- Beginning the Installation:
- Loading Packages:



Multidimensional data visualization across files. Explore relationships within and among related datasets.

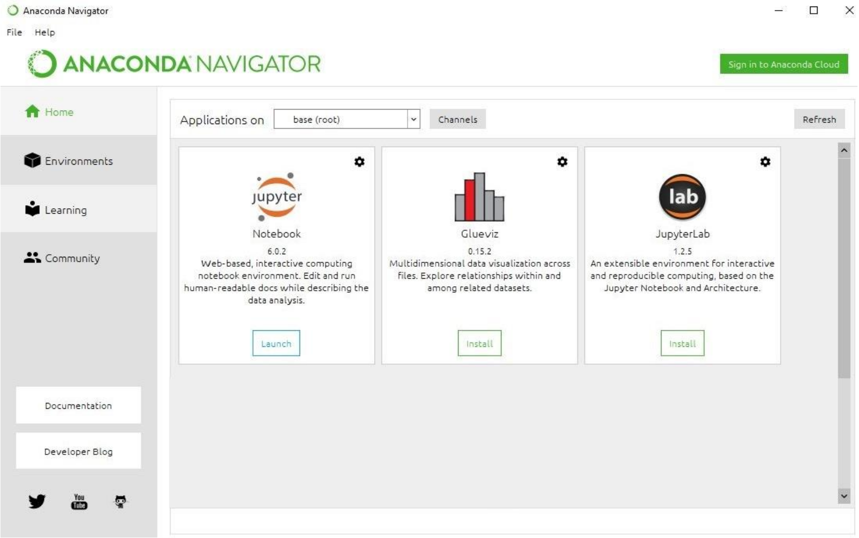
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Loading packages of **C:\Users\Abhinav Singh\Anaconda3**...

Figure 1.11. Installation Progress.

- Finished Installation:



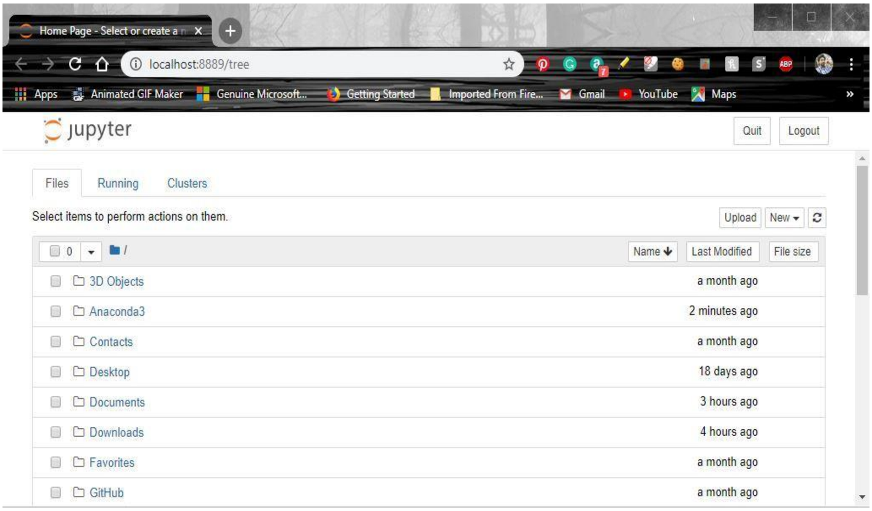
jupyter Notebook 6.0.2
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Glueviz 0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.

lab JupyterLab 1.2.5
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Figure 1.12. Launching Jupyter.

- Launching Jupyter:



Select items to perform actions on them.

Name	Last Modified	File size
□ 3D Objects	a month ago	
□ Anaconda3	2 minutes ago	
□ Contacts	a month ago	
□ Desktop	18 days ago	
□ Documents	3 hours ago	
□ Downloads	4 hours ago	
□ Favorites	a month ago	
□ GitHub	a month ago	

Figure 1.13. Get start with program interface in jupyter.

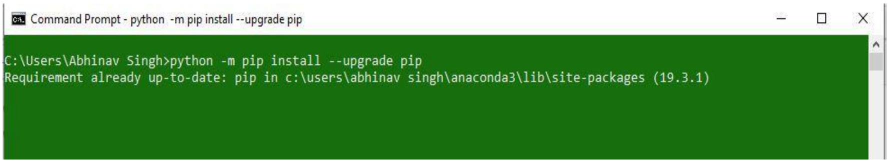
Installing Jupyter Notebook Using Pip

PIP is a package management system used to install and manage software packages/libraries written in Python. These files are stored in a large “on-line repository” termed as Python Package Index (PyPI).

PIP uses PyPI as the default source for packages and their dependencies.

To install Jupyter using pip, we need to first check if pip is updated in our system. Use the following command to update pip:

Python -m pip install –upgrade pip



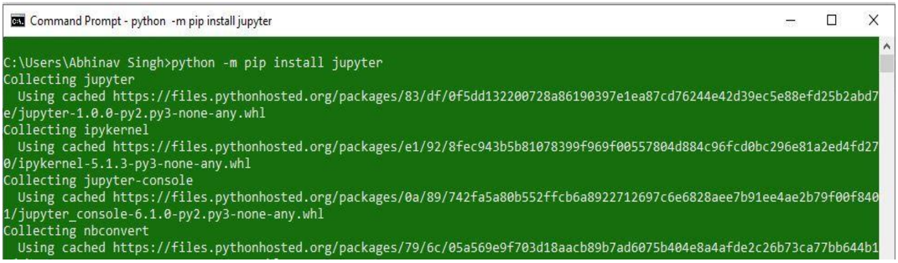
```
C:\Users\Abhinav Singh>python -m pip install --upgrade pip
Requirement already up-to-date: pip in c:\users\abhinav singh\anaconda3\lib\site-packages (19.3.1)
```

Figure 1.14. Command for installing package.

After updating the pip version, follow the instructions provided below to install Jupyter:

Command to install Jupyter:
Python -m pip install jupyter

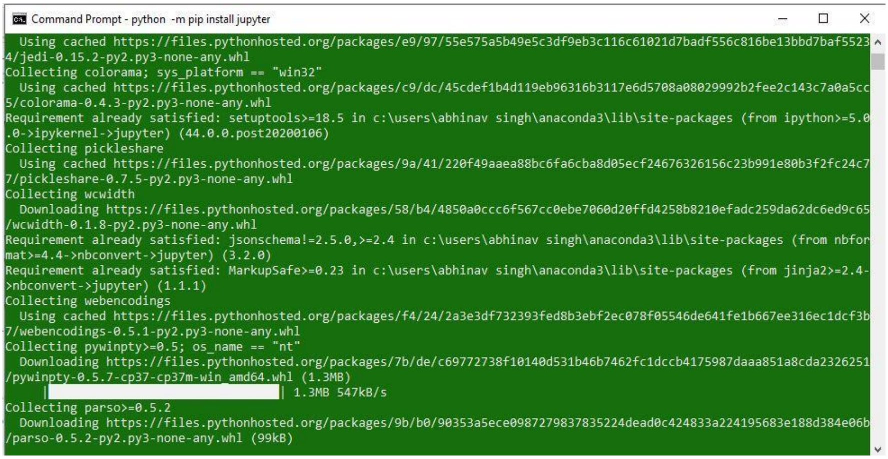
- Beginning Installation:



```
C:\Users\Abhinav Singh>python -m pip install jupyter
Collecting jupyter
  Using cached https://files.pythonhosted.org/packages/83/df/0f5dd132200728a86190397e1ea87cd76244e42d39ec5e88efd25b2abd7
e/jupyter-1.0.0-py2.py3-none-any.whl
Collecting ipykernel
  Using cached https://files.pythonhosted.org/packages/e1/92/8fec943b5b81078399f969f00557804d884c96fcd0bc296e81a2ed4fd27
0/ipykernel-5.1.3-py3-none-any.whl
Collecting jupyter-console
  Using cached https://files.pythonhosted.org/packages/0a/89/742fa5a80b552ffcb6a8922712697c6e6828aee7b91ee4ae2b79f00f840
1/jupyter_console-6.1.0-py2.py3-none-any.whl
Collecting nbconvert
  Using cached https://files.pythonhosted.org/packages/79/6c/05a569e9f703d18aacb89b7ad6075b404e8a4afde2c26b73ca77bb644b1
```

Figure 1.15. Package installation in progress.

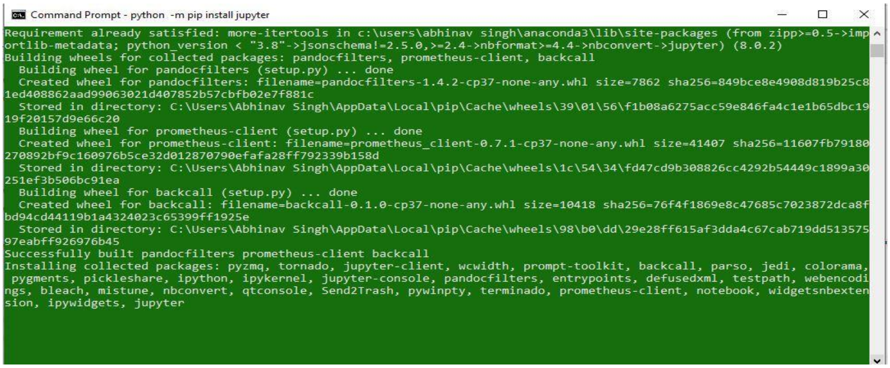
- Downloading Files and Data:



```
  Using cached https://files.pythonhosted.org/packages/e9/97/55e575a5b49e5c3df9eb3c116c61021d7badf556c816be13bbd7baf5523
4/jedi-0.15.2-py2.py3-none-any.whl
Collecting colorama; sys_platform == "win32"
  Using cached https://files.pythonhosted.org/packages/c9/dc/45cdef1b4d119eb96316b3117e6d5708a08029992b2fee2c143c7a0a5cc
5/colorama-0.4.3-py2.py3-none-any.whl
Requirement already satisfied: setuptools>=18.5 in c:\users\abhinav singh\anaconda3\lib\site-packages (from ipython>=5.0
.0->ipykernel->jupyter) (44.0.0.post20200106)
Collecting pickleshare
  Using cached https://files.pythonhosted.org/packages/9a/41/220f49aaea88bc6fa6cba8d05ecf24676326156c23b991e80b3f2fc24c7
7/pickleshare-0.7.5-py2.py3-none-any.whl
Collecting wcwidth
  Downloading https://files.pythonhosted.org/packages/58/b4/4850a0ccc6f567cc0ebe7060d20ffd4258b8210efadc259da62dc6ed9c65
/wcwidth-0.1.8-py2.py3-none-any.whl
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in c:\users\abhinav singh\anaconda3\lib\site-packages (from nbfor
mat>=4.4->nbconvert->jupyter) (3.2.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\abhinav singh\anaconda3\lib\site-packages (from jinja2>=2.4-
>nbconvert->jupyter) (1.1.1)
Collecting webencodings
  Using cached https://files.pythonhosted.org/packages/f4/24/2a3e3df732393fed8b3ebf2ec078f05546de641fe1b667ee316ec1dcf3b
7/webencodings-0.5.1-py2.py3-none-any.whl
Collecting pywinpty>=0.5; os_name == "nt"
  Downloading https://files.pythonhosted.org/packages/7b/de/c69772738f10140d531b46b7462fc1dccb4175987daaa851a8cda2326251
/pywinpty-0.5.7-cp37-cp37m-win_amd64.whl (1.3MB)
     |████████████████████████████████| 1.3MB 547kB/s
Collecting parso>=0.5.2
  Downloading https://files.pythonhosted.org/packages/9b/b0/90353a5ece0987279837835224dead0c424833a224195683e188d384e06b
/parso-0.5.2-py2.py3-none-any.whl (99kB)
```

Figure 1.16. Downloading data & file.

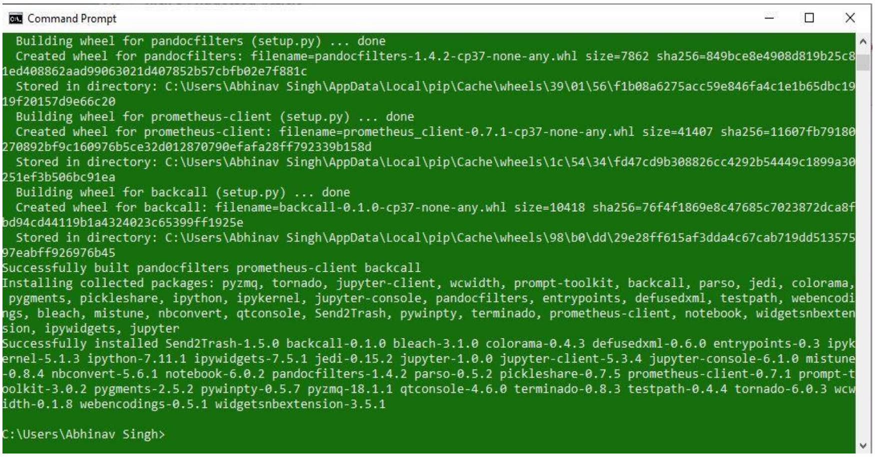
- Installing Packages:



```
Requirement already satisfied: more-itertools in c:\users\abhinav singh\anaconda3\lib\site-packages (from zipp>=0.5->imp
ortlib-metadata; python_version < "3.8"->jsonschema!=2.5.0,>=2.4->nbformat>=4.4->nbconvert->jupyter) (8.0.2)
Building wheels for collected packages: pandocfilters, prometheus-client, backcall
  Building wheel for pandocfilters (setup.py) ... done
  Created wheel for pandocfilters: filename=pandocfilters-1.4.2-cp37-none-any.whl size=7862 sha256=849bce8e4908d819b25c8
1ed408862aad99063021d407852b57cbfb02e7f881c
  Stored in directory: C:\Users\Abhinav Singh\AppData\Local\pip\Cache\wheels\39\01\56\f1b08a6275acc59e846fa4c1e1b65dbc19
19f20157d9e66c20
  Building wheel for prometheus-client (setup.py) ... done
  Created wheel for prometheus-client: filename=prometheus_client-0.7.1-cp37-none-any.whl size=41407 sha256=11607fb79180
270892bf9c160976b5ce32d012870790efafa28ff792339b158d
  Stored in directory: C:\Users\Abhinav Singh\AppData\Local\pip\Cache\wheels\1c\54\34\fd47cd9b308826cc4292b54449c1899a30
251ef3b506bc91ea
  Building wheel for backcall (setup.py) ... done
  Created wheel for backcall: filename=backcall-0.1.0-cp37-none-any.whl size=10418 sha256=76f4f1869e8c47685c7023872dca8f
bd94cd44119b1a4324023c65399ff1925e
  Stored in directory: C:\Users\Abhinav Singh\AppData\Local\pip\Cache\wheels\98\b0\dd\29e28ff615af3dda4c67cab719dd513575
97eabff926976b45
Successfully built pandocfilters prometheus-client backcall
Installing collected packages: pyzmq, tornado, jupyter-client, wcwidth, prompt-toolkit, backcall, parso, jedi, colorama,
 pygments, pickleshare, ipython, ipykernel, jupyter-console, pandocfilters, entrypoints, defusedxml, testpath, webencodi
ngs, bleach, mistune, nbconvert, qtconsole, Send2Trash, pywinpty, terminado, prometheus-client, notebook, widgetsnbexten
sion, ipywidgets, jupyter
```

Figure 1.17. Installation of packages.

- Finished Installation:



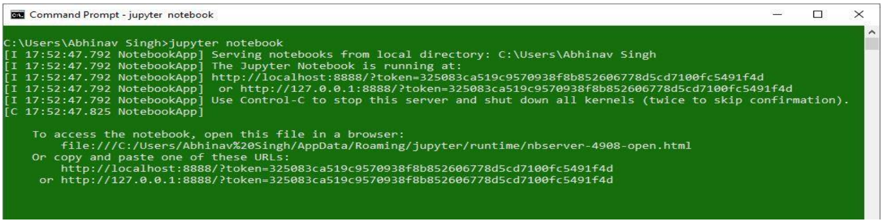
```
  Building wheel for pandocfilters (setup.py) ... done
  Created wheel for pandocfilters: filename=pandocfilters-1.4.2-cp37-none-any.whl size=7862 sha256=849bce8e4908d819b25c8
1ed408862aad99063021d407852b57cbfb02e7f881c
  Stored in directory: C:\Users\Abhinav Singh\AppData\Local\pip\Cache\wheels\39\01\56\f1b08a6275acc59e846fa4c1e1b65dbc19
19f20157d9e66c20
  Building wheel for prometheus-client (setup.py) ... done
  Created wheel for prometheus-client: filename=prometheus_client-0.7.1-cp37-none-any.whl size=41407 sha256=11607fb79180
270892bf9c160976b5ce32d012870790efafa28ff792339b158d
  Stored in directory: C:\Users\Abhinav Singh\AppData\Local\pip\Cache\wheels\1c\54\34\fd47cd9b308826cc4292b54449c1899a30
251ef3b506bc91ea
  Building wheel for backcall (setup.py) ... done
  Created wheel for backcall: filename=backcall-0.1.0-cp37-none-any.whl size=10418 sha256=76f4f1869e8c47685c7023872dca8f
bd94cd44119b1a4324023c65399ff1925e
  Stored in directory: C:\Users\Abhinav Singh\AppData\Local\pip\Cache\wheels\98\b0\dd\29e28ff615af3dda4c67cab719dd513575
97eabff926976b45
Successfully built pandocfilters prometheus-client backcall
Installing collected packages: pyzmq, tornado, jupyter-client, wcwidth, prompt-toolkit, backcall, parso, jedi, colorama,
 pygments, pickleshare, ipython, ipykernel, jupyter-console, pandocfilters, entrypoints, defusedxml, testpath, webencodi
ngs, bleach, mistune, nbconvert, qtconsole, Send2Trash, pywinpty, terminado, prometheus-client, notebook, widgetsnbexten
sion, ipywidgets, jupyter
Successfully installed Send2Trash-1.5.0 backcall-0.1.0 bleach-3.1.0 colorama-0.4.3 defusedxml-0.6.0 entrypoints-0.3 ipyk
ernel-5.1.3 ipython-7.11.1 ipywidgets-7.5.1 jedi-0.15.2 jupyter-1.0.0 jupyter-client-5.3.4 jupyter-console-6.1.0 mistune
-0.8.4 nbconvert-5.6.1 notebook-6.0.2 pandocfilters-1.4.2 parso-0.5.2 pickleshare-0.7.5 prometheus-client-0.7.1 prompt-t
oolkit-3.0.2 pygments-2.5.2 pywinpty-0.5.7 pyzmq-18.1.1 qtconsole-4.6.0 terminado-0.8.3 testpath-0.4.4 tornado-6.0.3 wcw
idth-0.1.8 webencodings-0.5.1 widgetsnbextension-3.5.1

C:\Users\Abhinav Singh>
```

Figure 1.18. Finishing installation.

- Launching Jupyter:

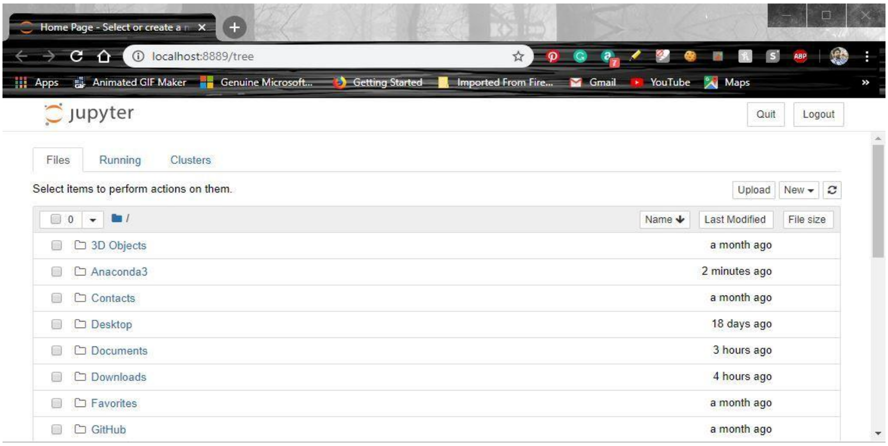
Use the following command to launch Jupyter using command-line:

Jupyter Notebook



```
C:\Users\Abhinav Singh>jupyter notebook
[I 17:52:47.792 NotebookApp] Serving notebooks from local directory: C:\Users\Abhinav Singh
[I 17:52:47.792 NotebookApp] The Jupyter Notebook is running at:
[I 17:52:47.792 NotebookApp] http://localhost:8888/?token=325083ca519c9570938f8b852606778d5cd7100fc5491f4d
[I 17:52:47.792 NotebookApp]  or http://127.0.0.1:8888/?token=325083ca519c9570938f8b852606778d5cd7100fc5491f4d
[I 17:52:47.792 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 17:52:47.825 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/Abhinav%20Singh/AppData/Roaming/jupyter/runtime/nbserver-4908-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=325083ca519c9570938f8b852606778d5cd7100fc5491f4d
     or http://127.0.0.1:8888/?token=325083ca519c9570938f8b852606778d5cd7100fc5491f4d
```

Figure 1.19. Get familiar with Command line.



Select items to perform actions on them.

Name	Last Modified	File size
3D Objects	a month ago	
Anaconda3	2 minutes ago	
Contacts	a month ago	
Desktop	18 days ago	
Documents	3 hours ago	
Downloads	4 hours ago	
Favorites	a month ago	
GitHub	a month ago	

Figure 1.20. Launch Python in Jupyter.

How to Run the Code in Jupyter Notebook?

Type the code in the cell and click on the run button located at top in the menu bar.

As soon as running completed your resultant outcome will get display at buttom of the cell.

You can find more information about Jupyter notebook on the below site:

https://jupyter.org/documentation

1.2.4. Atom

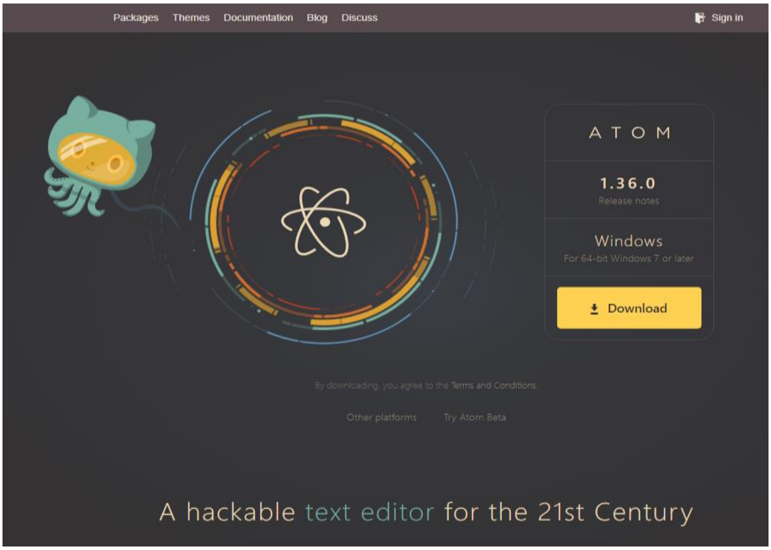
What Is Atom?

Atom is a free and an open source text editor for MacOS, Linux and Windows, developed by GitHub, which provides us with a platform to create responsive and interactive web applications.

How to Install Atom?

To get started with Atom, we'll need to get it on your system. This section will go over installing Atom on your system as well as the basics of how to build it from source.

Installing Atom should be fairly simple. Generally, you can go to https://atom.io and you should see a download button as shown here:

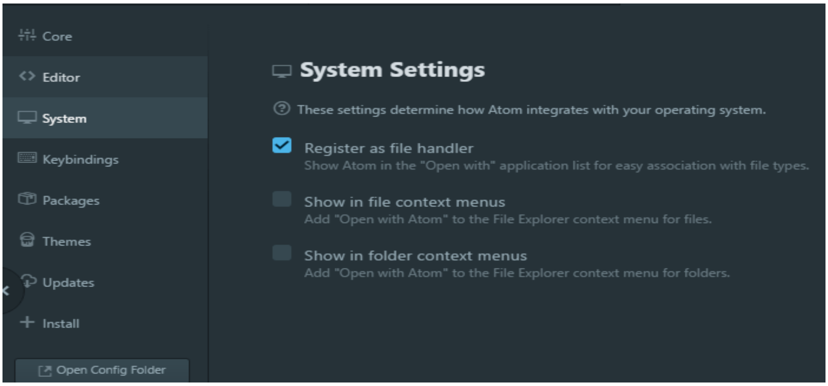


By downloading, you agree to the Terms and Conditions.

Figure 1.21. Download Atom.

The button or buttons should be specific to your platform and the download package should be easily installable. However, let's go over them here in a bit of detail.

Atom is available with Windows installers that can be downloaded from https://atom.io or from the Atom releases page. Use AtomSetup.exe for 32-bit systems and AtomSetup-x64.exe for 64-bit systems. This setup program will install Atom, add the atom and apm commands to your PATH, and create shortcuts on the desktop and in the start menu.



System Settings

These settings determine how Atom integrates with your operating system.

☑ Register as file handler
Show Atom in the "Open with" application list for easy association with file types.

☐ Show in file context menus
Add "Open with Atom" to the File Explorer context menu for files.

☐ Show in folder context menus
Add "Open with Atom" to the File Explorer context menu for folders.

Figure 1.22. Getting set with Atom according to system configuration.

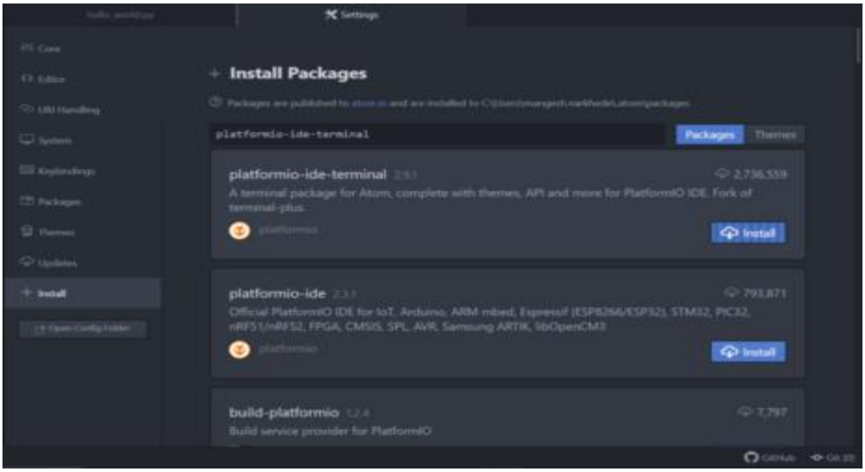
The context menu Open with Atom in File Explorer, and the option to make Atom available for file association using Open with..., is controlled by the System Settings panel as seen above.

With Atom open, click on File > Settings, and then the System tab on the left. Check the boxes next to Show in file context menus, as well as Show in folder context menus. And you're all set.

How to Use Atom?

Executing the Code

Normally, the command prompt is used to run Python programs. However, in Atom, a plugin called *platformio-ide-terminal* is available which can be used to execute the python files, To setup, this plugin, navigate to *File->Settings* click on Install tab and search for the platformio-ide-terminal plug-in and click on install.



Install Packages

Packages are published to atom.io and are installed to C:\Users\mangesh.narkhede\.atom\packages

platformio-ide-terminal

platformio-ide-terminal 2.9.1 — 2,736,559
A terminal package for Atom, complete with themes, API and more for PlatformIO IDE. Fork of terminal-plus.
platformio

platformio-ide 2.3.1 — 793,871
Official PlatformIO IDE for IoT, Arduino, ARM mbed, Espressif (ESP8266/ESP32), STM32, PIC32, nRF51/nRF52, FPGA, CMSIS, SPL, AVR, Samsung ARTIK, libOpenCM3
platformio

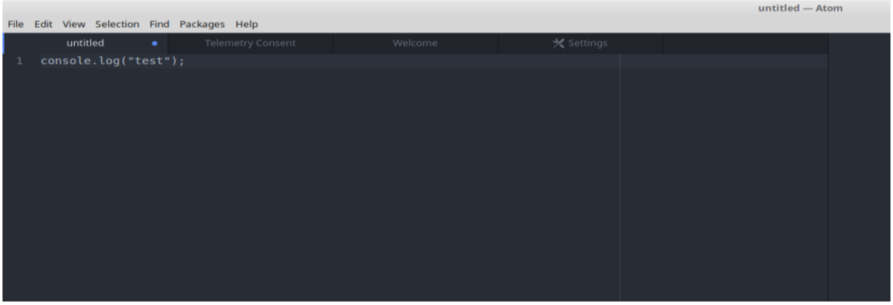
build-platformio 1.2.4 — 7,797
Build service provider for PlatformIO

Figure 1.23. Installation of package in atom.

Once the installation is finished, a terminal will be integrated inside Atom and you will be able to see a + icon in the left corner of the Atom python editor. The terminal will open in the current directory if clicked on it.

You can also view plugin details by clicking on the plugin package tab. It will show all the required details and how to use the plugins.

Start writing the code here...



```
console.log("test");
```

Figure 1.24. Programming interface:Atom.

1.3. LIBRARIES

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs.

1.3.1. Numpy

Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

You may install it through command prompt if your python is installed with following command,

pip install numpy

The source code for NumPy is located at this github repository:

numpy/numpy: The fundamental package for scientific computing with Python.

You can find more information about numpy in link: https://numpy.org/

1.3.2. Pandas

In computer programming, pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license.

When you want to use Pandas for data analysis, you'll usually use it in one of three different ways:

- Convert a Python's list, dictionary or Numpy array to a Pandas data frame
- Open a local file using Pandas, usually a CSV file, but could also be a delimited text file (like TSV), Excel, etc
- Open a remote file or database like a CSV or a JSONon a website through a URL or read from a SQL table/database.

You may install it through command prompt if your python is installed with following command,

pip install pandas

The source code for pandas is located at this github repository:

https://github.com/pandas-dev/pandas

You can find more information about pandas here: https:// pandas.pydata.org/

1.3.3. Scikit-Learn

Scikit-learn is an open source Python library that has powerful tools for data analysis and data mining. Scikit-learn (formerly scikits. learn and also known as sklearn) is a free software machine learning library for the Python programming language.] It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

You may install it through command prompt if your python is installed with following command,

pip install scikit-learn

The source code for scikit learn is located at this github repository:

https://github.com/scikit-learn/scikit-learn

You can find more about scikit-learn from the mentioned link: https://scikit-learn.org/stable/

1.3.4. Matlplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

You may install it through command prompt if your python is installed with following command,

pip install matplotlib

The source code for matplotlib is located at this github repository:

https://github.com/matplotlib/matplotlib

To know more about Matplotlib you may refer the mentioned link:- https://matplotlib.org/

1.3.5. Seaborn

Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.

You may install it through command prompt if your python is installed with following command,

pip install seaborn

The source code for seaborn is located at this github repository:

https://github.com/mwaskom/seaborn

To find more information about seaborn refer the mentioned link for reference: https://seaborn.pydata.org/

LINKS AND REFERENCES USED IN THIS CHAPTER

Links

http://wiki.python.org/moin/BeginnersGuide/Download
http://www.python.org/doc/faq/windows/
http://docs.python.org/dev/3.0/using/windows.html
http://docs.python.org/dev/3.0/using/unix.html
http://www.python.org/download/mac/
http://docs.python.org/dev/3.0/using/mac.html
https://jupyter.org/documentation
numpy/numpy: The fundamental package for scientific computing with Python.
https://numpy.org/
https://github.com/pandas-dev/pandas
https://pandas.pydata.org/
https://github.com/scikit-learn/scikit-learn
https://scikit-learn.org/stable/
https://github.com/matplotlib/matplotlib

https://matplotlib.org/
https://seaborn.pydata.org/

References

Barry, Paul. *Head First Python 2e: A Brain-Friendly Guide*, 16 December, 2016/

Matthes, Eric. *Python Crash Course*, 2nd Edition: A Hands-On, Project-Based Introduction to Programming, 3 May, 2019.

Chapter 2

FOUNDATION OF MACHINE LEARNING

2.1. WHAT IS MACHINE LEARNING?

The scientific field of machine learning (ML) is a branch of artificial intelligence, as defined by Computer Scientist and machine learning pioneer.

"Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience." by Tom M. Mitchell.

Machine learning usually refers to the changes in systems that perform tasks associated with artificial intelligence (AI). By Nils J. Nilsson.

Machine learning is an application of artificial intelligence (AI) which provides system the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

2.1.1. Application of Machine Learning

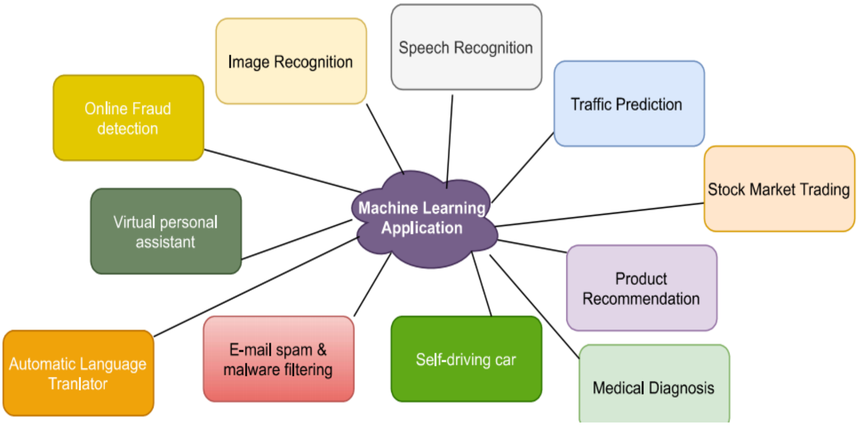


Figure 2.1. Application of Machine Learning.

Image Recognition

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. Image recognition refers to technologies that identify places, logos, people, objects, buildings and several other variables in digital images. It may be very easy for humans like you and me to recognise different images, such as images of animals. We can easily recognise the image of a cat and differentiate it from an image of a horse. But it may not be so simple for a computer.

Figure 2.2. Image Recognition.

A digital image is an image composed of picture elements, also known as pixels, each with finite, discrete quantities of numeric representation for its intensity or grey level. So the computer sees an image as numerical values of these pixels and in order to recognise a certain image, it has to recognise the patterns and regularities in this numerical data.

Speech Recognition

While using Google, we get an option of "Search by voice," it comes under speech recognition and it's a popular application of machine learning.

Speech recognition is a process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition." At present, machine learning algorithms are widely used by various applications of speech recognition. Google assistant, Siri, Cortana, and Alexa are using speech recognition technology to follow the voice instructions.

Figure 2.3. Speech Recognition.

Traffic Prediction

If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions. Similarly, Traffic Prediction is the function of predicting real-time traffic information based on floating vehicle data and road history data, such as traffic flow, average traffic speed and traffic incidents.

It can be predicted by two ways whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:

- Real Time location of the vehicle form Google Map app and sensors
- Average time has taken on past days at the same time.

Everyone who is using Google Map is helping this app to make it better. It takes information from the user and send it back to its database to improve the performance.

Figure 2.4. Traffic Prediction.

Product Recommendations

Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for recommendation of product to the user. Whenever we search for some product on Amazon, then we start getting an advertisement for the same product while internet surfing on the same browser because of machine learning.

Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.

Similarly, when we use Netflix, we find some recommendations for entertainment series, movies, etc., this is also done with the help of machine learning.

Figure 2.5. Product Recommendation.

Self-Driving Cars

One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving cars. It is using unsupervised learning methods to train the car models to detect people and objects while driving.

Figure 2.6. Self driving car.

Email Spam and Malware Filtering

Whenever we receive a new email, it is filtered automatically as important, normal, and spam. We always receive an important mail in our inbox with the important symbol and spam emails in our spam box and the technology behind this is Machine learning. Below are some spam filters used by Gmail:

- Content Filter
- Header Filter
- General blacklists Filter
- Rules-based Filters
- Permission Filters

Some machine learning algorithms such as Multi-Layer Perceptron, Decision Tree and Naïve Bayes Classifier are used for email spam filtering and malware detection.

Figure 2.7. Spam activity.

Virtual Personal Assistant

We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.

These virtual assistants use machine learning algorithms as an important part.

These assistants record our voice instructions, send it over the server on a cloud and decode it using ML algorithms and act accordingly.

Figure 2.8. Virtual assistant.

Online Fraud Detection

Machine learning is making our online transaction safe and secure by detecting fraud transactions. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids and stealing money in the middle of a transaction. So, to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.

For each genuine transaction, the output is converted into some hash values and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets changed for the fraud transaction hence, it detects it and makes our online transactions more secure.

Figure 2.9. Fraud detection.

Stock Market Trading

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning’s long short term memory neural network is used for the prediction of stock market trends.

Figure 2.10. Stock market trading.

Medical Diagnosis

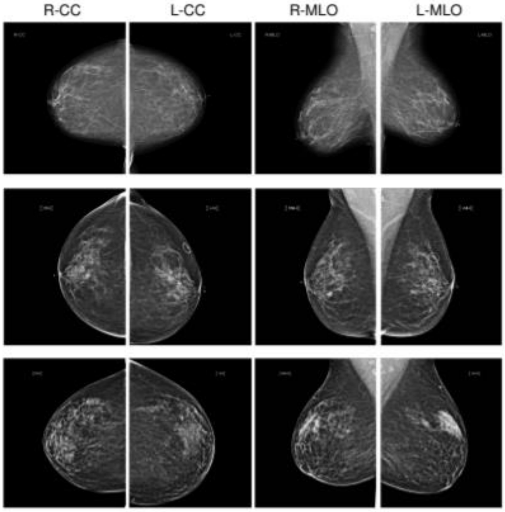


Figure 2.11. Breast cancer detection.

In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.

It helps in finding brain tumors and other brain-related diseases easily.

Automatic Language Translation

Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all. Machine Learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language and it called as automatic translation.

The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

Figure 2.12. Language Translator.

2.1.2. Dataset

What Is Dataset?

A data set (or dataset) is a collection of data. In the case of tabular data, a data set corresponds to one or more database tables, where every column of a table represents a particular variable and each row corresponds to a given record of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. Data sets can also consist of a collection of documents or files.

Types of Data

Lists: In some cases, the vectors we obtain may contain a variable number of features. For instance, a physician might not necessarily decide to perform a full battery of diagnostic tests if the patient appears to be healthy.

Sets may appear in learning problems whenever there is a large number of potential causes of an effect, which are not well determined. For instance, it is relatively easy to obtain data concerning the toxicity of mushrooms. It would be desirable to use such data to infer the toxicity of a new mushroom given information about its chemical compounds. However, mushrooms contain a cocktail of compounds out of which one or more may be toxic. Consequently we need to infer the properties of an object given a set of features, whose composition and number may vary considerably.

Matrices are a convenient means of representing pairwise relationships. For instance, in collaborative filtering applications the rows of the matrix may represent users whereas the columns correspond to products. Only in some cases we will have knowledge about a given (user, product) combination, such as the rating of the product by a user. A related situation occurs whenever we only have similar information between observations, as implemented by a semi-empirical distance measure. Some homology searches in bioinformatics, e.g., variants of BLAST, only return a similarity score which does not necessarily satisfy the requirements of a metric.

Images could be thought of as two-dimensional arrays of numbers, i.e., matrices. This representation is very crude, though, since they exhibit spatial coherence (lines, shapes) and (natural images exhibit) a multiresolution structure. That is, down sampling an image leads to an object which has very similar statistics to the original image. Computer vision and psychotics have created a raft of tools for describing these phenomena.

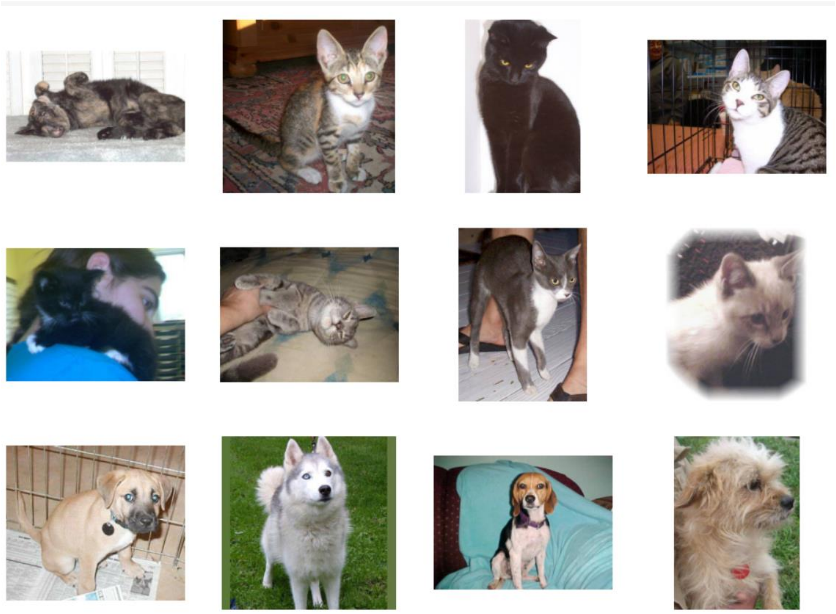


Figure 2.13. Image dataset.

Video adds a temporal dimension to images. Again, we could represent them as a three-dimensional array. Good algorithms, however take the temporal coherence of the image sequence into account.

Trees and Graphs are often used to describe relations between collections of objects. For instance, the ontology of webpages of the DMOZ project (www.dmoz.org) has the form of a tree with topics becoming increasingly refined as we traverse from the root to one of the leaf (Arts → Animation → Anime → General Fan Pages → Official Sites). In the case of gene ontology the relationships form a directed acyclic graph also referred the GO-DAG.

Audio: Audio Set consists of an expanding ontology of 632 audio event classes and a collection of 2,084,320 human-labeled 10-second sound clips drawn from YouTube videos. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres and common everyday environmental sounds.

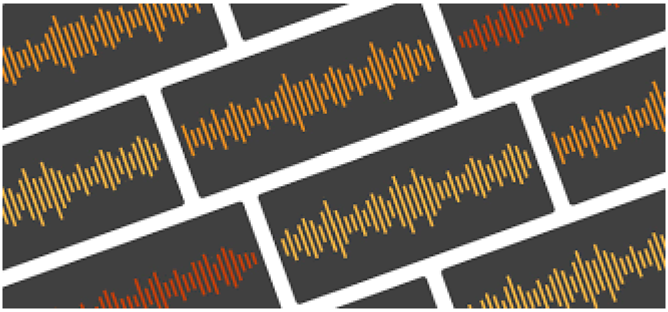


Figure 2.14. Audio dataset.

Why Is Data Important?

Data is important because, without data, there would be no world. The world is built upon data, so it makes sense to learn how to use it so that you can benefit the world in many ways. It's not the quantity of information that counts, but the kind of information you are able to acquire and the reason why you want to get it.

Data is important because, without data, there would be no world. The world is built upon data, so it makes sense to learn how to use it so that you can benefit the world in many ways.

There are many sources from where datasets can be easily found. Some of them are mentioned below:

1. FiveThirtyEight:- https://data.fivethirtyeight.com/
2. Google Public Datasets:- https://cloud.google.com/bigquery /public-data/
3. Kaggle:- https://www.kaggle.com/datasets
4. Socrata:- https://opendata.socrata.com/
5. UCI Machine Learning Repository:- http://archive.ics.uci.edu/ml/ index.php
6. Data.gov:- https://www.data.gov/
7. Academic Torrents:- https://academictorrents.com/browse.php

8. Quandl:- https://www.quandl.com/search
9. Jeremy Singer-Vine:- https://tinyletter.com/data-is-plural

Awesome-Public-Datasets on Github:- https://github.com/awesomedata/awesome-public-datasets

2.1.3. Why Machine Learning in Solving Problems?

We would like machines to adjust their internal structure to produce correct outputs for a large number of sample inputs and thus suitably constrain their input/output function to approximate the relationship implicit in the examples.

It is possible that hidden among large piles of data are important relationships and correlations. Machine learning methods can often be used to extract these relationships (data mining).

Human designers often produce machines that do not work as well as desired in the environments in which they are used. In fact, certain characteristics of the working environment might not be completely known at design time. Machine learning methods can be used for on-the-job improvement of existing machine designs.

The amount of knowledge available about certain tasks might be too large for explicit encoding by humans. Machines that learn this knowledge gradually might be able to capture more of it than humans would want to write down.

Environments change over time. Machines that can adapt to a changing environment would reduce the need for constant redesign.

New knowledge about tasks is constantly being discovered by humans. There is a constant stream of new events in the world. Continuing redesign of AI systems to conform new knowledge is impractical, but machine learning methods might be able to track much of it.

2.2. Technique of Machine Learning

Regression and classification are two techniques used when designing machine learning algorithms. Both regression machine learning algorithms and classification machine learning algorithms are classified under the realm of supervised machine learning.

2.2.1. Regression

Regression is the process of finding a model that predicts a continuous value based on its input variables. In regression problems, the goal is to mathematically estimate a mapping function f(*f*) from the input variables x(*x*) to the output variables y(*y*).

Consider a dataset that contains information about all the students in a university. An example of a regression task would be to predict the height of any student based on their gender, weight, major and diet. We can do this because height is a continuous quantity; i.e., there are an infinite amount of possible values for a person's height.

A regression algorithm is commonly evaluated by calculating the *root mean squared error* of its output.

2.2.2. Classification

Classification is the process of finding a model that separates input data into multiple discrete classes or labels. In other words, a classification problem determines whether or not an input value can be part of a pre-identified group.

Consider the same dataset of all the students at a university. A classification task would be to use parameters, such as a student's weight, major and diet, to determine whether they fall into the "Above Average" or "Below Average" category. Note that there are only two discrete labels in which the data is classified.

A classification algorithm is evaluated by computing the *accuracy* with which it correctly classified its input.

2.3. TYPES OF MACHINE LEARNING

Machine learning is no exception, and a good flow of organized, varied data is required for a robust ML solution. Machine Learning algorithms have the ability to improve themselves through training. Today, ML algorithms are trained using three prominent methods. These are three types of machine learning: supervised learning, unsupervised learning and reinforcement learning.

2.3.1. Supervised Learning

Supervised learning is one of the most basic types of machine learning. In this type, the machine learning algorithm is trained on labeled data. Even though the data needs to be labeled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances.

In supervised learning, the ML algorithm is given a small training dataset to work with. This training dataset is a smaller part of the bigger dataset and serves to give the algorithm a basic idea of the problem, solution and data points to be dealt with. The training dataset is also very similar to the final dataset in its characteristics and provides the algorithm with the labeled parameters required for the problem.

The algorithm then finds relationships between the given parameters, essentially establishing a cause and effect relationship between the variables in the dataset. At the end of the training, the algorithm has an idea of how the data works and the relationship between the input and the output.

This solution is then deployed for use with the final dataset, which learns in the same way as the training dataset. This means that supervised machine learning algorithms will continue to improve even after being deployed, discovering new patterns and relationships as it trains itself on new data.

Figure 2.15. Supervised Learning.

In above image, you can see that we are feeding raw inputs as an image of apple to the algorithm, as a part of the algorithm we have a supervisor who keeps on correcting the machine or who keeps on training the machines or keeps on telling him that yes it is an apple or it is not an apple, things like that.

So this process keeps on repeating until we get a final trained model, once the model is ready it can easily predict the correct output of a never seen input.

Some of the algorithms for supervised learning are as mentioned below:

1. Linear Regression
2. Random Forest
3. Support Vector Machines (SVM)

We will study these Algorithms in subsequent chapters.

Applications of Supervised Learning

- Bioinformatics – This is one of the most well-known applications of Supervised Learning because most of us use it in our day-to-day lives. Bioinformatics is the storage of Biological Information of humans such as fingerprints, iris texture, earlobe and so on. Today’s Cell Phones are capable of learning our biological information and are then able to authenticate us bringing up the security of the system. Smartphones such as iPhones, Google Pixel are capable of facial recognition while OnePlus, Samsung is capable of In-display finger recognition.

Figure 2.16. Bioinformatics.

- *Speech Recognition* – This is the kind of application where you teach the algorithm about your voice and it will be able to recognize you. The most well-known real-world applications are virtual assistants such as Google Assistant and Siri, which will wake up to the keyword with your voice only.

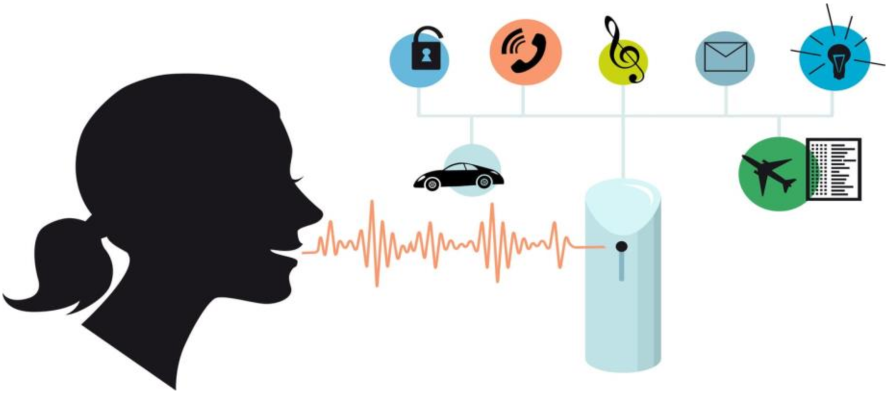


Figure 2.17. Speech Recognition.

- *Spam Detection* – This application is used where the unreal or computer-based messages and E-Mails are to be blocked. G-Mail has an algorithm that learns the different keywords which could be fake such as “You are the winner of something” and so forth blocks those messages directly. OnePlus Messages App gives the user the task of making the application learn which keywords need to be blocked and the app will block those messages with the keyword.

Figure 2.18. Spam detection.

- *Object-Recognition for Vision* – This kind of application is used when you need to identify something. You have a huge dataset which you use to teach your algorithm and this can be used to recognize a new instance. Raspberry Pi algorithms which detect objects are the most well-known example.

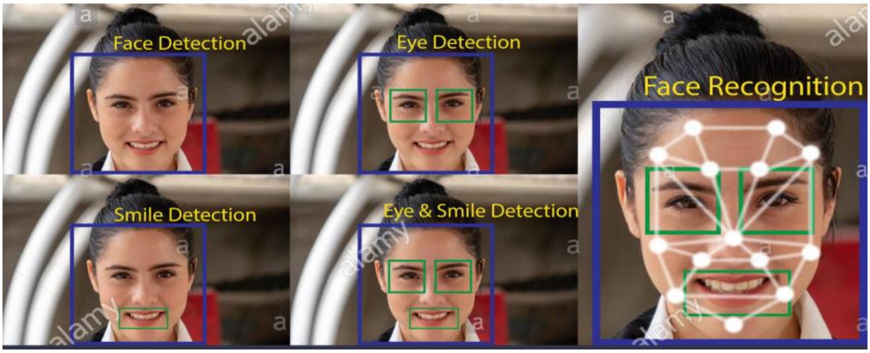


Figure 2.19. Object recognition.

2.3.2. Unsupervised Learning

Unsupervised machine learning holds the advantage of being able to work with unlabeled data. This means that human labor is not required to make the dataset machine-readable, allowing much larger datasets to be worked on by the program.

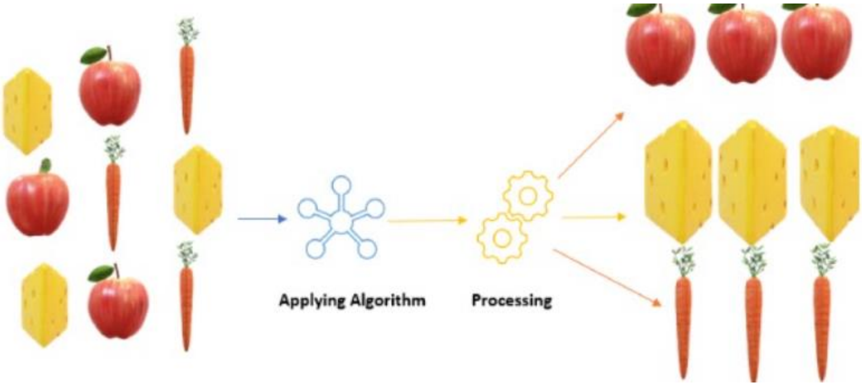


Figure 2.20. Unsupervised Learning.

In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning do not have labels to work off, resulting in the creation of hidden structures. Relationships between data points are perceived by the algorithm in an abstract manner, with no input required from human beings.

The creation of these hidden structures is what makes unsupervised learning algorithms versatile. Instead of a defined and set problem statement, unsupervised learning algorithms can adapt to the data by dynamically changing hidden structures. This offers more post-deployment development than supervised learning algorithms.

As we have already discussed that in unsupervised learning our dataset is not labelled, So if we are feeding apple, carrot and cheese as raw input data then our model will distinguish all three but it cannot tell whether a given cluster is of apple or not as it is unlabelled but any new data will automatically fit into the clusters that are formed.

Some algorithms available for unsupervised learning are as mentioned below:

1. Principal Component Analysis Algorithm
2. K-means Algorithm
3. Singular Value Decomposition Algorithm

We will study this in subsequent chapters.

Applications of Unsupervised Learning in Companies

- *AirBnB* – This is a great application which helps host stays and experiences connecting people all over the world. This application uses Unsupervised Learning where the user queries requirements and Airbnb learns these patterns and recommends stays and experiences which fall under the same group or cluster.

Figure 2.21. AirBnB.

- *Amazon* – Amazon also uses unsupervised learning to learn the customer's purchase and recommend the products which are most frequently bought together which is an example of association rule mining.

Figure 2.22. Amazon.

- *Credit-Card Fraud Detection* – Unsupervised Learning algorithms learn about various patterns of the user and their usage of the credit card. If the card is used in parts that do not match the behavior, an alarm is generated which could possibly be marked as fraud and calls are given to users to confirm whether it was used by them or not.

2.3.3. Reinforcement Learning

Reinforcement learning directly takes inspiration from how human beings learn from data in their lives. It features an algorithm that improves upon itself and learns from new situations using a trial-and-error method. Favorable outputs are encouraged or ‘reinforced’, and non-favorable outputs are discouraged or ‘punished’.

Based on the psychological concept of conditioning, reinforcement learning works by putting the algorithm in a work environment with an interpreter and a reward system. In every iteration of the algorithm, the output result is given to the interpreter, which decides whether the outcome is favorable or not.

In case of the program finding the correct solution, the interpreter reinforces the solution by providing a reward to the algorithm. If the outcome is not favorable, the algorithm is forced to reiterate until it finds a better result. In most cases, the reward system is directly tied to the effectiveness of the result.

In typical reinforcement learning use-cases, such as finding the shortest route between two points on a map, the solution is not an absolute value. Instead, it takes on a score of effectiveness, expressed in a percentage value. The higher this percentage value is, the more reward is given to the algorithm. Thus, the program is trained to give the best possible solution for the best possible reward.

Let’s understand reinforcement learning by this example in this agent gives itself a reward with correct actions or predictions to improve its environment. So basically, this agent is supposed to get more and more rewards to get better results or achieve goals. Lastly, whichever environment of agents creates better results will be our best possible model.

Applications of Reinforcement Learning

- *Robotics.* RL can be used for high-dimensional control problems as well as various industrial applications. Google, for example, has reportedly cut its energy consumption by about 50% after implementing Deep Mind's technologies. There are innovative startups in the space (Bonsai, etc.) that are propagating deep reinforcement learning for efficient machine and equipment tuning.

Figure 2.23. Robotics.

- *Text mining.* The researchers from Salesforce, a renowned cloud computing company, used RL along with an advanced contextual text generation model to develop a system that's able to produce highly readable summaries of long texts. According to them, one can train their algorithm on different types of material (news articles, blogs, etc.).

Figure 2.24. Trade Execution.

- *Trade execution.* Major companies in the financial industry have been using ML algorithms to enhance trading and equity for a while and some of them, such as JPMorgan, have already thrown their hats into the RL ring too. The company announced in 2017 that it would start using a robot for trading execution of large orders. Their model, trained on billions of historic transactions, would allow them to execute trading procedures promptly, at optimal prices and offload huge stakes without creating market swings.
- *Healthcare.* Recent papers suggest multiple applications for RL in the healthcare industry. Among them are medication dosing, optimization of treatment policies for those suffering from chronic, clinical trials, etc.

Figure 2.25. Healthcare.

Links and References Used in this Chapter

Links

1. https://data.fivethirtyeight.com/
2. https://cloud.google.com/bigquery/public-data/
3. https://www.kaggle.com/datasets/

4. https://opendata.socrata.com/
5. http://archive.ics.uci.edu/ml/index.php
6. https://www.data.gov/
7. https://academictorrents.com/browse.php
8. https://www.quandl.com/search/
9. https://tinyletter.com/data-is-plural
10. https://github.com/awesomedata/awesome-public-datasets

References

Alpaydin, Ethem. *Introduction to Machine Learning*, The MIT Press, ISBN: 978-0-262-01243-0.

Kubat, Miroslav. *An Introduction to Machine Learning*, Springer Publishing Company, Incorporated, ISBN:978-3-319-20009-5.

Wub, Nan. Phang, Jason et al. *Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening*. arXiv: 1903.08297v1 [cs.LG] 20 Mar 2019.

Chapter 3

DATA PRE-PROCESSING

A data set (sometimes referred to as data source, or database) in the context of Tableau, contains the data used to build visualizations. Every bar chart, scatter plot, or line chart you see in Tableau has a connected database or spreadsheet that supplies the data.

3.1. WHAT IS DATA PREPROCESSING?

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

To build and develop Machine Learning models, you must first acquire the relevant dataset. This dataset will be composed of data gathered from multiple and disparate sources which are then combined in a proper format to form a dataset. Dataset formats differ according to use cases. For instance, a business dataset will be entirely different from a medical dataset. While a business dataset will contain relevant industry and business data, a medical dataset will include healthcare-related data.

For the further explanation we have used the dataset (income.csv).

CODE: To download, read and display the dataset.
You can find the code here: https://rb.gy/4ve8yw

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
```

```
ls
```

```
import pandas as pd
```

```
data=pd.read_csv(‘income.csv’)
```

```
data.head()
```

3.2. Features in Machine Learning

We will consider the problem and understand the concept accordingly.
Problem:

- Subsidy Inc. delivers subsidies to individuals based on their income.
- Accurate Income data is one of the hardest pieces of data to obtain across the world.
- Subsidy Inc. has obtained a large dataset of authenticated data on individual income, demographic parameters and few financial parameters.
- Subsidy Inc. wishes to:- Develop an income classifier system for individual

Objectives: Simplify the data system by reducing the number of variables to be studied without sacrificing too much of accuracy. Such a system would help Subsidy Inc. in planning subsidy outlay, monitoring and preventing misuse.

You can find it here: https://rb.gy/7ublly

3.2.1. What Is the Feature?

Feature is the process of using the domain knowledge of the data to create features that makes machine learning algorithms work properly.

Below is a short description of each feature in the data set:

1. Age: The age of individual in a year
2. Jobtype: working status of the person which sector does he work in
3. Edtype: Level of Education
4. Marital status: The marital status of the individual
5. Occupation: The type of work individual does
6. Relationship: The relationship of the individual to his/her household
7. Race: the individual race
8. Gender:The individual Gender
9. Capitalgain: The Capital Gain of the individual
10. Capitalloss: The Capital Loss of the individual
11. Hoursperweek: The Number of hours individuals works per week
12. Nativecountry: The native Country of the individual.
13. SalStat: The outcome variable indicating whether a person's salary status.

CODE: print the columns head in the dataset

For more detail, refer to this link:- https://rb.gy/ewnpg5

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
Import pandas as pd
data=pd.read_csv('income.csv')
data.head()
```

```
data.columns
```

3.2.2. Data Type

Having a good understanding of the different data types, also called measurement scales, is a crucial prerequisite for doing Exploratory Data Analysis (EDA), since you can use certain statistical measurements only for specific data types.

You also need to know which data type you are dealing with to choose the right visualization method. Think of data types as a way to categorize different types of variables. We will discuss the main types of variables and look at an example for each. We will sometimes refer to them as measurement scale.

There are two type of dataType mainly:

1. Categorical
2. Numerical

Data type of the feature in the dataset is explained as follows:

Table 3.1. Feature Explanation

S. No	Variable	Data Type
1	Relationship	String
2	Race	String
3	Gender	String
4	Capitalgain	Integer
5	Capitalloss	Integer
6	Hoursperweek	Integer
7	Nativecountry	String
8	Salstat	String
9	Age	Integer
10	Jobtype	String
11	EdType	String
12	Marritalstatus	String
13	Occupaption	String

CODE*:* printing the Data type in the dataset.

You can find the code here:- https://rb.gy/8x3j4r

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv(‘income.csv’)
data.head()
```

```
data.dtype
```

```
data[‘age’].dtype
```

3.2.3. Categorical of Variable

There are variables in the dataset for each feature in the dataset. For this dataset the variable in the feature are as follows:

Table 3.2. Feature Variable

S. No	Variable	Categories of variable
1	Age	– –
2	Job Type	Private,Federal-gov,Self-emp-inc,Self-emp-not-inc,Local-gov & more.
3	EdType	HD-grad,Some-college,9th,Assoc-voc-Bachelors,1st-4th,Masters & more
4	maritalstatus	Divorced,Never-married,Married-civ-spouse,Widowed & more
5	occupation	Adm-cierical,ARmed-Forces.Prof-speciality,craft-repair,Sales,Exec-managerial,machine-op-inspct,Tranport-moving,Farming-fishing, Tech-support & more.
6	relationship	Not-in-family,own-child, Husband, Other-relative,Unmarried,Wife.
7	race	White,Black,Amer-Indian-Eskimo,Asian-Pac-Islander
8	gender	Female,Male
9	capitalgain	– –
10	capitalloss	– –
11	hoursperweek	– –
12	nativecountry	United-States,Mexico,Canada,& 39 more
13	salstat	Less than or equal to 50,000, Greater than 50,000

CODE: Print the number of non null rows in each column and print unique values in them.

You can find the code here: https://rb.gy/9q971w

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv('income.csv')
data.head()
```

```
data.info()
```

```
import numpy as np
print(np.unique(data["JobType"]))
print(np.unique(data[“Occupation”]))
```

```
data[‘JobType’].value_counts()
```

```
data[‘Occupation’].value_counts()
```

3.3. DATA QUALITY ASSESSMENT

The Data Quality Assessment is intended to be a stand-alone report documenting the drivers, process, observations, and recommendations from the data profiling process. The term “data quality” refers to the suitability of data to serve its intended purpose. So, measuring data quality involves performing data quality assessments to determine the degree to which your data adequately supports the business needs of the company.

A data quality assessment is done by measuring particular features of the data to see if they meet defined standards. Each such feature is called a “data quality dimension,” and is rated according to a relevant metric that provides an objective assessment of quality. A Data Quality Assessment is a distinct phase within the data quality life-cycle that is used to verify the source, quantity and impact of any data items that breach pre-defined data quality rules.

3.3.1. Missing Values

Missing values are representative of the messiness of real-world data. There can be a multitude of reasons why they occur - ranging from human errors during data entry, incorrect sensor readings, to software bugs in the data processing pipeline.

The missing value in the dataset according to the row is as follows:

Table 3.3. Missing values

S. No	Variable	Null values
1	age	0
2	JobType	0
3	EdType	0
4	maritalstatus	0
5	occupation	0
6	relationship	0
7	race	0
8	gender	0
9	capitalgain	0
10	capitalloss	0
11	hoursperweek	0
12	nativecountry	0
13	salstat	0

CODE: Printing the missing value in dataset

You can find the code here: https://rb.gy/fkb101

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv('income.csv')
data.head()
```

```
data.isnull().sum()
```

3.3.2. Exploring Dataset

To Arrive at a good conclusion, you need to understand the dataset, and understanding the important features is important. So that the machine learning model can be used to predict good results.

Mean: A mean is the simple mathematical average of a set of two or more numbers.

Mode: The mode is the value that appears most frequently in a data set. A set of data may have one mode, more than one mode, or no mode at all.

Standard Deviation: The standard deviation is a statistic that measures the dispersion of a dataset relative to its mean and is calculated as the square root of the variance. The standard deviation is calculated as the square root of variance by determining each data point's deviation relative to the mean.

Unique: It is essentially material that *you* create rather than data that already exists. This is often a process of 'adding-value' to existing data such as compiling statistics in a spreadsheet or database from a series or variety of pre-existing documents or it could refer to documents that you have transcribed from originals.

Frequency: Frequency refers to the number of times an event or a value occurs. A frequency table is a table that lists items and shows the number of times the items occur.

CODE: Show count, mean, mode, standard deviation, unique, top, frequency in dataset.

You can find the code here: https://rb.gy/1bfie1

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv('income.csv')
data.head()
```

```
data.describe()
```

```
data.describe(include= 'o')
```

The dataset contains the count, mean, standard deviation, minimum, maximum, unique, top, frequency, as follows in the table:

Table 3.4. Exploration of dataset

	age	capitalgain	capitalloss	hoursperweek
count	31978.000000	31978.000000	31978.000000	31978.000000
mean	38.579023	1064.360623	86.739352	40.417850
std	13.662085	7298.596271	401.594301	12.345285
min	17.000000	0.000000	0.000000	1.000000
25%	28.000000	0.000000	0.000000	40.000000
50%	37.000000	0.000000	0.000000	40.000000
75%	48.000000	0.000000	0.000000	45.000000
max	90.000000	99999.000000	4356.000000	99.000000

Table-3.5. Exploration of dataset

	Count	Unique	Top	freq
JobType	31978	9	Private	22286
EdType	31978	16	HS-grad	10368
maritalstatus	31978	7	Married-civ-spouse	14692
occupation	31978	15	prof-speciality	4038
relationship	31978	6	Husband	12947
race	31978	5	White	27430
gender	31978	2	Male	21370
nativecountry	31978	41	United-States	29170
SalStat	31978	2	50,000>=	24283

The unique values in the dataset for some columns is as follows:

1) Job Type:

   ‘Fedral-gov’, ‘Local-gov’, ‘Never-worked’, ‘Private’, ‘Self-emp-inc’, ‘self-emp-not-inc’, ‘State-gov’, ‘Without-pay’.

2) Occupation:

   ‘Adm-clerical’, ‘Armed-Forces’, ‘Craft-repair’, ‘Exec-managerial’, ‘Farming-fishing’, ‘Handlers-cleaners’, ‘Machine-op-inspct’, ‘Other-services’, ‘Priv-house-serv’, ‘Prof-speciality’, ‘Protective-serv’, ‘sales’, ‘Tech-support’, ‘Transport-moving’.

As we see there are impurities in the dataset, such as ‘?’ which does not imply any value. Thus there are some impurities in the dataset which need to be converted into NaN value, since it does not contain any value.

CODE: read? as NaN

You can find the code here: https://rb.gy/yrwijc

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv(‘income.csv’,na_values=[“?”])
data.head()
```

The Null value in the dataset after converting “?” into NaN is as follows:

Table-3.6. Null value of variables

S. No	**Variable**	**Null values**
1	age	0
2	JobType	1809
3	EdType	0
4	maritalstatus	0
5	occupation	1816
6	relationship	0
7	race	0
8	gender	0
9	capitalgain	0
10	capitalloss	0
11	hoursperweek	0
12	nativecountry	0
13	salstat	0

CODE: Checking the missing value, nan value

You can find the code here: https://rb.gy/il7u2x

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv('income.csv',na_values=["?"])
data.head()
```

```
data.insull().sum()
```

Here is the Image how the code look after it is given in Figure 3.1.

In order to fill the value in the column we can:

1) Delete the whole column if the column is Mostly Empty
2) Considering Mean, Mode, Median if the data type of column is int or float to fill the missing value.
3) Considering Top unique value in the column if the data type of column is Objective to fill the missing values.

CODE: Filling int with mean, mode, median, Fill object with max Freq.

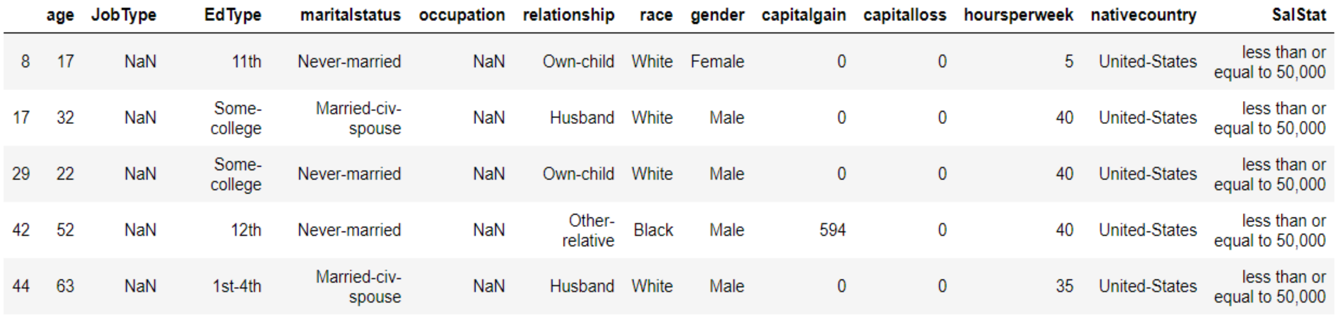
You can find the code here: https://rb.gy/jfnmsh

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv('income.csv',na_values=["?"])
data.head()
```

```
data.insull().sum()
```

```
data.describe(include- 'o')
```



	age	JobType	EdType	maritalstatus	occupation	relationship	race	gender	capitalgain	capitalloss	hoursperweek	nativecountry	SalStat
8	17	NaN	11th	Never-married	NaN	Own-child	White	Female	0	0	5	United-States	less than or equal to 50,000
17	32	NaN	Some-college	Married-civ-spouse	NaN	Husband	White	Male	0	0	40	United-States	less than or equal to 50,000
29	22	NaN	Some-college	Never-married	NaN	Own-child	White	Male	0	0	40	United-States	less than or equal to 50,000
42	52	NaN	12th	Never-married	NaN	Other-relative	Black	Male	594	0	40	United-States	less than or equal to 50,000
44	63	NaN	1st-4th	Married-civ-spouse	NaN	Husband	White	Male	0	0	35	United-States	less than or equal to 50,000

Figure 3.1. Output of the above code.

```
data['JobType']=data['JobType'].fillna(data['JobType'].describe().top)
```

```
data['Occupation']=data['Occupation'].fillna(data['Occupation'].describe().top)
```

```
data.insull().sum()
```

Finally after this the Null value per column in the dataset is as follows:

Table-3.7. Null value per column

S.No	Variable	Null values
1	age	0
2	JobType	0
3	EdType	0
4	maritalstatus	0
5	occupation	0
6	relationship	0
7	race	0
8	gender	0
9	capitalgain	0
10	capitalloss	0
11	hoursperweek	0
12	nativecountry	0
13	salstat	0

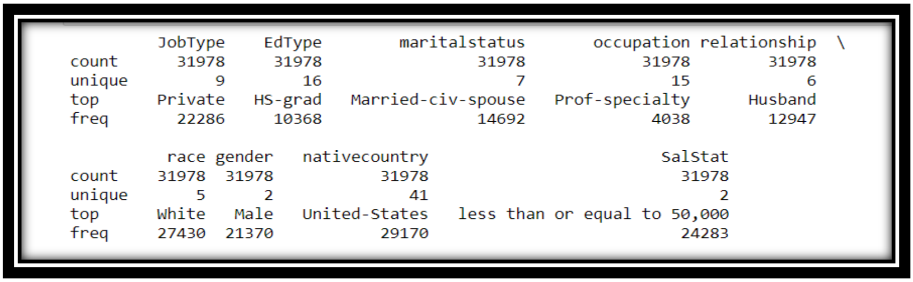
3.4. FEATURE ENCODING

Encoding refers to converting the labels into numeric form so as to convert it into the machine-readable form.

Pandas have a function which can turn a categorical variable into a series of zeros and ones.

The pandas function pd.get_dummies for my such as {JobType, EdType, maritalstatus, occupation, relationship, race, gender, native country, salstat} categorical variable. Let's consider a Column gender, since this variable has only two answer choices: male and female. pd.

get_dummies create a new dataframe that consists of zeros and ones similarly for all the choices in the remaining Categorical variable.



```
        JobType   EdType        maritalstatus       occupation relationship  \
count     31978    31978                31978            31978        31978
unique        9       16                    7               15            6
top     Private  HS-grad   Married-civ-spouse   Prof-specialty      Husband
freq      22286    10368                14692             4038        12947

         race gender   nativecountry                          SalStat
count   31978  31978           31978                            31978
unique      5      2              41                                2
top     White   Male   United-States   less than or equal to 50,000
freq    27430  21370           29170                            24283
```

Figure 3.2. Feature encoding.

The Feature are as follows after removing the 'SalStat' column:

1. 'nativecountry_ Ecuador', 'EdType_ Assoc-voc',
2. 'nativecountry_ Poland', 'EdType_ 5th-6th', 'race_ Other', 'occupation_ Machine-op-inspct', 'nativecountry_ India', 'occupation_ Prof-specialty',
3. 'nativecountry_ Philippines', 'occupation_ Sales', 'race_ Asian-Pac-Islander',
4. 'nativecountry_ Peru', 'capitalgain', 'occupation_ Tech-support',
5. 'nativecountry_ Vietnam', 'EdType_ Masters', 'EdType_ 1st-4th', 'EdType_ Preschool', 'maritalstatus_ Widowed',
6. 'nativecountry_ Trinadad&Tobago',
7. 'nativecountry_ Iran', 'EdType_ HS-grad', 'occupation_ Protective-serv', 'relationship_ Own-child',
8. 'nativecountry_ Italy', 'occupation_ Armed-Forces',
9. 'nativecountry_ Honduras',
10. 'nativecountry_ Haiti', 'nativecountry_ Puerto-Rico', 'JobType_ State-gov',
11. 'nativecountry_ Thailand', 'EdType_ Bachelors', 'occupation_ Handlers-cleaners', 'nativecountry_ United-States',
12. 'nativecountry_ Yugoslavia', 'EdType_ Some-college', 'race_ Black', 'nativecountry_ Holand-Netherlands',

13. ‘nativecountry_ El-Salvador’, ‘hoursperweek’, ‘nativecountry_ Outlying-US(Guam-USVI-etc)’, ‘nativecountry_ Hungary’, ‘nativecountry_ France’, ‘EdType_ Assoc-acdm’,
14. ‘nativecountry_ Taiwan’, ‘race_ White’,
15. ‘nativecountry_ Dominican-Republic’, ‘nativecountry_ Mexico’,
16. ‘nativecountry_ Greece’, ‘occupation_ Other-service’, ‘maritalstatus_ Married-civ-spouse’, ‘JobType_ Local-gov’, ‘occupation_ Craft-repair’, ‘EdType_ 9th’,
17. ‘nativecountry_ Ireland’, ‘EdType_ Prof-school’, ‘gender_ Male’,
18. ‘nativecountry_ Columbia’,
19. ‘nativecountry_ Hong’, ‘occupation_ Priv-house-serv’, ‘maritalstatus_ Separated’,
20. ‘nativecountry_ Japan’, ‘age’, ‘occupation_ Transport-moving’, ‘occupation_ Farming-fishing’, ‘JobType_ Without-pay’, ‘relationship_ Not-in-family’,
21. ‘nativecountry_ Canada’,
22. ‘nativecountry_ England’, ‘relationship_ Wife’,
23. ‘nativecountry_ South’, ‘EdType_ 12th’, ‘JobType_ Self-emp-not-inc’,
24. ‘nativecountry_ Laos’, ‘capitalloss’, ‘maritalstatus_ Married-spouse-absent’, ‘relationship_ Unmarried’, ‘nativecountry_ Jamaica’,
25. ‘nativecountry_ Portugal’, ‘EdType_ 7th-8th’, ‘maritalstatus_ Never-married’, ‘maritalstatus_ Married-AF-spouse’, ‘occupation_ Exec-managerial’,
26. ‘nativecountry_ Guatemala’,
27. ‘nativecountry_ Cuba’, ‘EdType_ 11th’, ‘JobType_ Private’,
28. ‘nativecountry_ China’, ‘JobType_ Self-emp-inc’,
29. ‘nativecountry_ Nicaragua’, ‘EdType_ Doctorate’, ‘relationship_ Other-relative’,
30. ‘nativecountry_ Scotland’,
31. ‘nativecountry_ Germany’]

The value of all the features in the dataset will act as array X. The values in the ‘SalStat’ will act as array Y. This X and Y will further be used for Training Supervised Machine Learning algorithm and predicting Y value for an unknown X value. We will see this in further chapters.

CODE: Create feature array and target array.

You can find the code here: https://rb.gy/nyi1hv

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv(‘income.csv’,na_values=[“?”])
data.head()
```

```
data[‘JobType’]=data[‘JobType’].fillna(data[‘JobType’].describe().top)
data[‘occupation’]=data[‘occupation’].fillna(data[‘occupation’].describe().top)
```

```
data[‘SalStat’]=data[‘SalStat’].map({“ less than or equal to 50,000”:0,” greater than 50,000”:1})
```

```
new_data=pd.get_dummies(data,drop_first=True)
new_data.head()
```

```
columns_list=list(new_data.columns)
print(columns_list)
```

```
features=list(set(columns_list)-set([“SalStat”]))
print(features)
```

```
x=new_data[features].values
print(x)
```

```
x=new_data[features].values
print(x)
```

```
y=new_data[“SalStat”].values
print(y)
```

3.5. Splitting the Dataset

Using Sklearn train_test_split() from model_selection we can split the dataset as we want.

The parameters include x values and y values which need to be splitted in specific size mentioned into test_size parameter. The code returns train_x, test_x, train_y, test_y which will further be used to train the model.

Example;

train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.3,random_state=0)

This code will return 70 % train dataset and rest 30% will be used for testing.

CODE: for splitting the dataset.

You can find the code here:- https://rb.gy/ls6w9a

```
!git clone -l -s git://github.com/neural2020ml/machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
import pandas as pd
data=pd.read_csv(‘income.csv’,na_values=[“?”])
data.head()
```

```
data[‘JobType’]=data[‘JobType’].fillna(data[‘JobType’].describe().top)
data[‘occupation’]=data[‘occupation’].fillna(data[‘occupation’].describe().top)
```

```
#convert text into 1 and 0 whereever possible
data[‘SalStat’]=data[‘SalStat’].map({“ less than or equal to 50,000”:0,” greater than 50,000”:1})
#converting the dataset into computer understable 1 and 0 form
new_data=pd.get_dummies(data,drop_first=True)
#column in dataset are:
columns_list=list(new_data.columns)
#the feature are
features=list(set(columns_list)-set([“SalStat”]))
#the x array is
x=new_data[features].values
#the y array is
```

```
y=new_data[“SalStat”].values
```

```
from sklearn.model_selection import train_test_split
```

```
train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.2,random_state=0)
```

```
train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.4,random_state=0)
```

LINKS AND REFERENCES USED IN THIS CHAPTER

Links

1. https://rb.gy/4ve8yw
2. https://rb.gy/7ublly
3. https://rb.gy/ewnpg5
4. https://rb.gy/8x3j4r
5. https://rb.gy/9q971w
6. https://rb.gy/fkb101
7. https://rb.gy/1bfie1
8. https://rb.gy/yrwijc
9. https://rb.gy/il7u2x
10. https://rb.gy/jfnmsh
11. https://rb.gy/nyi1hv
12. https://rb.gy/ls6w9a

References

Krzanowski, W. Multiple discriminant analysis in the presence of mixed continuous and categorical data. *Comput. Math. Appl.* 12(2, Part A), 179–185 (1986).

Little, R. J. A., Schluchter, M.D. Maximum likelihood estimation for mixed continuous and categorical data with missing values. *Biometrika* 72, 497–512 (1985).

Pyle, D. *Data Preparation for Data Mining.* Morgan Kaufmann, San Francisco (1999).

Chapter 4

SUPERVISED LEARNING

4.1. INTRODUCTION

Supervised learning is one of the most basic types of machine learning. In this type, the machine learning algorithm is trained on labeled data. Even though the data needs to be labeled accurately for this method to work, supervised learning is extremely powerful when used in the right circumstances.

Learning under supervision directly translates to being under guidance and learning from an entity that is in charge of providing feedback through this process. When training a machine, supervised learning refers to a category del learns to fit mapping between examples of input features with their associated labels. Where we teach or train a machine learning algorithm using data while guiding the algorithm model with labels associated with the data.

In supervised learning, our goal is to learn the mapping function (f), which refers to being able to understand how the input (X) should be matched with output (Y) using available data.

Here, the machine learning models are trained with these examples, we can use them to make new predictions on unseen data.

The predicted labels can be both numbers or categories. For instance, if we are predicting house prices, then the output is a number. In this case, the model is a regression model. If we are predicting if an email is spam or not, the output is a category and the model is a classification model.

4.2. Linear Regression

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithms show a linear relationship between a dependent (y) and one or more independent (y) variables, hence called linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

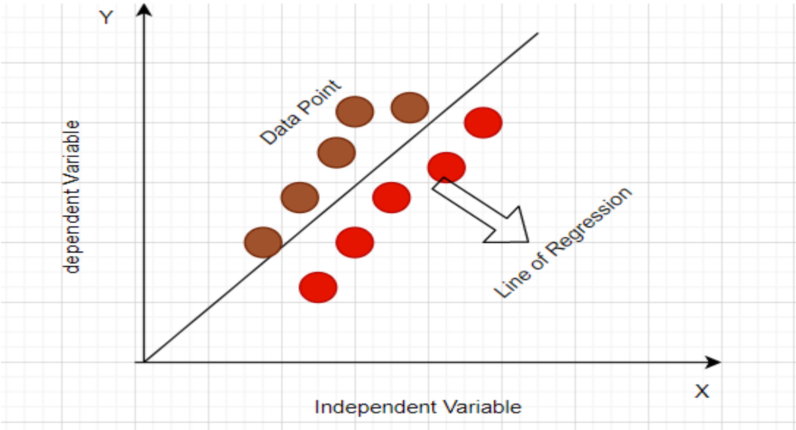


Figure 4.1. Linear Regression; Dependent v/s independent variable.

4.2.1. Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

- Simple Linear Regression:If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.
- Multiple Linear regression:If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

CODE: The code for linear regression is as mentioned below.
You can find the code here:- https://rb.gy/6uxiuk

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

```
# Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)
print(‘The first row in Dataset is:’,diabetes_X[0],’\n’,’The first label in the dataset
is:’,diabetes_y[0],’\n’)
print(‘Number of rows in Dataset and Label:\n’,len(diabetes_X),
len(diabetes_y))
```

```
# Use only one feature
diabetes_X = diabetes_X[:, np.newaxis, 2]
# Split the data into training/testing sets
diabetes_X_train = diabetes_X[:-20]
diabetes_X_test = diabetes_X[-20:]
print(‘Number of rows in Training:’,len(diabetes_X_train))
print(‘Number of rows in Testing:’,len(diabetes_X_test))
```

```
# Split the targets into training/testing sets
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]
print(‘Number of Label in Training:’,len(diabetes_y_train))
print(‘Number of Label in Testing:’,len(diabetes_y_test))
```

```
# Create linear regression object
l_regression =LinearRegression()
# Train the model using the training sets
l_regression.fit(diabetes_X_train, diabetes_y_train)
# Make predictions using the testing set
diabetes_y_pred = l_regression.predict(diabetes_X_test)
print(diabetes_y_pred)
```

```
# The coefficients
print(‘Coefficients: \n’, l_regression.coef_)
# The mean squared error
print(‘Mean squared error:’ ,mean_squared_error(diabetes_y_test, diabetes_y_pred))
# The coefficient of determination: 1 is perfect prediction
print(‘Coefficient of determination:’,r2_score(diabetes_y_test, diabetes_y_pred))
```

```
# Plot outputs
plt.scatter(diabetes_X_test, diabetes_y_test, color=‘black’)
plt.plot(diabetes_X_test, diabetes_y_pred, color=‘blue’, linewidth=3)
plt.xticks(())
```

```
plt.yticks(())
plt.show()
```

4.3. Logistic Regression

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes. Some of the examples of classification problems are Email spam or not spam, online transactions Fraud or not Fraud, Tumor Malignant or Benign. Logistic regression transforms its output using the logistic sigmoid function to return a probability value.

Logistic Regression is a Machine Learning algorithm which is used for classification problems, it is a predictive analysis algorithm and based on the concept of probability. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

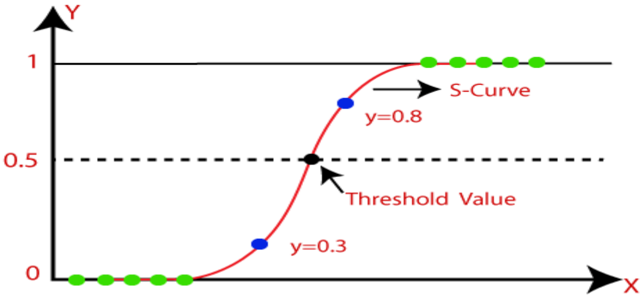


Figure 4.2. Logistic variable.

4.3.1. Types of Logistic Regression

Logistic Regression can be classified into three types:

- Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as “cat”, “dogs”, or “sheep”
- Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as “low”, “Medium”, or “High”.

CODE: The code for logistic regression is as mentioned below.
You can find the code here:- https://rb.gy/palefj

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn import datasets
```

```
# import some data to play with
iris = datasets.load_iris()
X = iris.data[:, :2] # we only take the first two features.
Y = iris.target
```

```
# Create an instance of Logistic Regression Classifier and fit the data.
log_reg = LogisticRegression()
log_reg.fit(X, Y)
```

```
# Plot the decision boundary. For that, we will assign a color to each
# point in the mesh [x_min, x_max]x[y_min, y_max].
x_min, x_max = X[:, 0].min() - .5, X[:, 0].max() + .5
y_min, y_max = X[:, 1].min() - .5, X[:, 1].max() + .5
h = .02 # step size in the mesh
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = log_reg.predict(np.c_[xx.ravel(), yy.ravel()])
```

```
# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1, figsize=(4, 3))
plt.pcolormesh(xx, yy, Z, cmap=plt.cm.Paired)
```

```
# Plot also the training points
plt.scatter(X[:, 0], X[:, 1], c=Y, edgecolors=‘k’, cmap=plt.cm.Paired)
plt.xlabel(‘Sepal length’)
plt.ylabel(‘Sepal width’)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xticks(())
plt.yticks(())
plt.show()
```

```
# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure(1, figsize=(4, 3))
plt.pcolormesh(xx, yy, Z, cmap=plt.cm.Paired)
# Plot also the training points
plt.scatter(X[:, 0], X[:, 1], c=Y, edgecolors=‘k’, cmap=plt.cm.Paired)
plt.xlabel(‘Sepal length’)
plt.ylabel(‘Sepal width’)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xticks(())
plt.yticks(())
plt.show()
```

4.4. NAÏVE BAYES

The Naïve Bayes algorithm consists of two words Naïve and Bayes, It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of colour, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identifying that it is an apple without depending on each other. Bayes, it is called Bayes because it depends on the principle of Bayes’ Theorem.

Naive Bayes is a machine learning model that is used for large volumes of data, even if you are working with data that has millions of data records the recommended approach is Naive Bayes. It gives very good results when it comes to NLP tasks such as sentimental analysis. It is a fast and uncomplicated classification algorithm.

4.5. Bayes’ Theorem

- Bayes’ theorem is also known as Bayes’ Rule or Bayes’ law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes’ theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

4.5.1. Types of Naive Bayes Algorithms

1. Gaussian Naïve Bayes: When characteristic values are continuous in nature then an assumption is made that the values linked with each class are dispersed according to Gaussian that is Normal Distribution.
2. Multinomial Naïve Bayes: Multinomial Naive Bayes is favored to use on data that is multinomial distributed. It is widely used in text classification in NLP. Each event in text classification constitutes the presence of a word in a document.
3. Bernoulli Naïve Bayes: When data is dispensed according to the multivariate Bernoulli distributions then Bernoulli Naive Bayes is used. That means there exist multiple features but each one is assumed to contain a binary value. So, it requires features to be binary-valued.

CODE: The code for Naïve Bayes is as mentioned below.
You can find the code here: https://rb.gy/rnazpm

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
```

```
#dataset
X, y = load_iris(return_X_y=True)
```

```
#first feature
X[0]
```

```
#first feature label
y[0]
```

```
#split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
```

```
gnb = GaussianNB()
```

```
y_pred = gnb.fit(X_train, y_train)
```

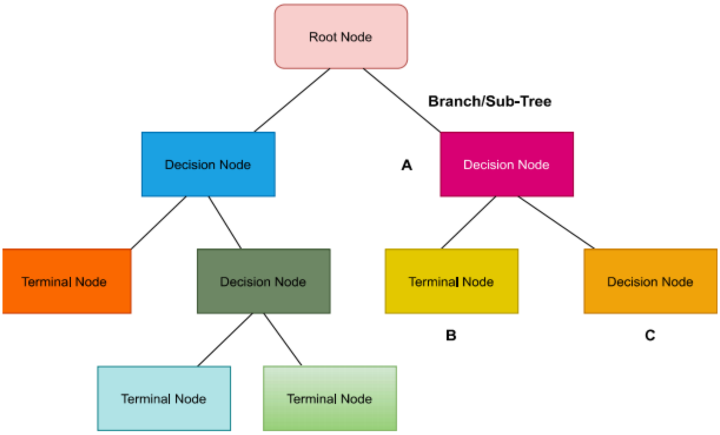
```
answer=gnb.predict(X_test)
```

```
print(“Number of mislabeled points out of a total %d points : %d” % (X_test.shape[0], (y_test !=
y_pred).sum()))
```

4.6. Decision Tree

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record’s attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

- Root Node: It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
- Splitting: It is a process of dividing a node into two or more sub-nodes.
- Decision Node: When a sub-node splits into further sub-nodes, then it is called the decision node.
- Leaf/Terminal Node: Nodes that do not split are called Leaf or Terminal nodes.
- Pruning: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
- Branch/Sub-Tree: A subsection of the entire tree is called a branch or sub-tree.
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.



Note: a is parent node of B and C.

Figure 4.3. Decision Tree.

CODE: The Code of Decision Tree is as mentioned below.

You can find the code here: https://rb.gy/fi6scz

```
# Clone the entire repo.
!gitclone-l-shttps://github.com/neural2020ml/Machine-learning-Algorithm-with-python-Book.git
cloned-repo
%cd cloned-repo
!ls
```

```
#load dataset
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
#show dataset
dataset = pd.read_csv(‘consumption_of_petrol.csv’)
dataset.head()
```

```
#create an X and Y array
X = dataset.drop(‘Petrol_Consumption’, axis=1)
y = dataset[‘Petrol_Consumption’]
```

```
#split the dataset
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
from sklearn.tree import DecisionTreeRegressor
regres = DecisionTreeRegressor()
regres.fit(X_train, y_train)
```

```
y_pred = regres.predict(X_test)
```

```
df=pd.DataFrame({‘Actual’:y_test, ‘Predicted’:y_pred})
df.head()
```

```
#evalution
from sklearn import metrics
print(‘Mean Absolute Error:’, metrics.mean_absolute_error(y_test, y_pred))
print(‘Mean Squared Error:’, metrics.mean_squared_error(y_test, y_pred))
print(‘RootMeanSquared Error:’, np.sqrt(metrics.mean_squared_
error(y_test, y_pred)))
```

4.7. K-Nearest Neighbours

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

The KNN algorithm hinges on this assumption being true enough for the algorithm to be useful. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics we might have learned in our childhood— calculating the distance between points on a graph.

KNN algorithm is one of the simplest classification algorithms. Even with such simplicity, it can give highly competitive results. KNN algorithms can also be used for regression problems. The only difference from the discussed methodology will be using averages of nearest neighbors rather than voting from nearest neighbors.

CODE: The code for KNN is as mentioned below.

You can find the code here: https://rb.gy/w5eb6y

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
url = “https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data”
# Assign colum names to the dataset
names = [‘sepal-length’, ‘sepal-width’, ‘petal-length’, ‘petal-width’, ‘Class’]
# Read dataset to pandas dataframe
dataset = pd.read_csv(url, names=names)
```

```
dataset.head()
```

```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 4].values
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
```

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5)
classifier.fit(X_train, y_train)
```

```
y_pred = classifier.predict(X_test)
```

```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
```

```
print(classification_report(y_test, y_pred))
```

4.8. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. Linear Discriminant Analysis is the most commonly used dimensionality reduction technique in supervised learning. Basically, it is a preprocessing step for pattern classification and machine learning applications.

Under Linear Discriminant Analysis, we are basically looking for:

1. Which set of parameters can best describe the association of the group for an object?
2. What is the best classification preceptor model that separates those groups?

It is widely used for modeling varieties in groups, i.e., distributing variables into two or more classes, suppose we have two classes and we need to classify them efficiently.

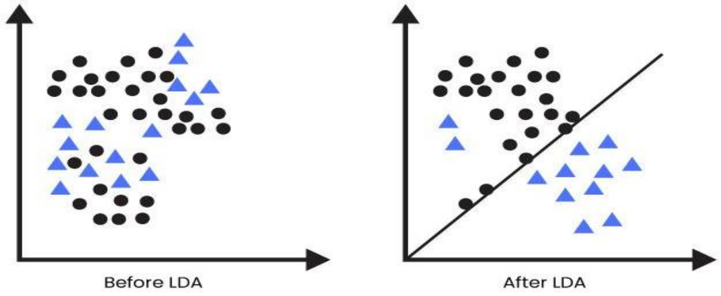


Figure 4.4. Linear discriminant analysis.

Classes can have multiple features, using one single feature to classify may yield in some kind of overlapping of variables, so there is a need of increasing the number of features to avoid overlapping that would result in proper classification in return.

CODE: The code for Linear Discriminant analysis is as mentioned below.

You can find the code here: https://rb.gy/wmafxn

```
import numpy as np
import pandas as pd
```

```
url = “https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data”
names = [‘sepal-length’, ‘sepal-width’, ‘petal-length’, ‘petal-width’, ‘Class’]
dataset = pd.read_csv(url, names=names)
```

```
X = dataset.iloc[:, 0:4].values
y = dataset.iloc[:, 4].values
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis as LDA
lda = LDA(n_components=1)
X_train = lda.fit_transform(X_train, y_train)
X_test = lda.transform(X_test)
```

```
from sklearn.ensemble import RandomForestClassifier
classifier = RandomForestClassifier(max_depth=2, random_state=0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
```

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
cm = confusion_matrix(y_test, y_pred)
print(cm)
```

```
print(‘Accuracy=‘ + str(accuracy_score(y_test, y_pred)))
```

4.9. Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

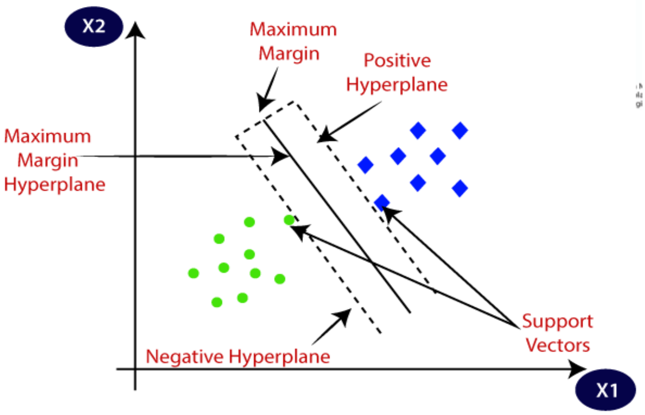


Figure 4.5. Support vector machine.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM algorithm can be used for Face detection, image classification, text categorization, etc.

Types of SVM

SVM can be of two types:

- Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

CODE: The code for Support Vector Machine (SVM) is mentioned below.

You can find the code here: https://rb.gy/s4anxk

```
#Import scikit-learn dataset library
from sklearn import datasets
#Load dataset
cancer = datasets.load_breast_cancer()
```

```
# print the names of the 13 features
print(“Features: “, cancer.feature_names)
# print the label type of cancer(‘malignant’ ‘benign’)
print(“Labels: “, cancer.target_names)
```

```
cancer.data.shape
```

```
# Import train_test_split function
from sklearn.model_selection import train_test_split# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target,
test_size=0.3,random_state=109) # 70% training and 30% test
```

```
#Import svm model
from sklearn import svm
#Create a svm Classifier
clf = svm.SVC(kernel=‘linear’) # Linear Kernel
#Train the model using the training sets
clf.fit(X_train, y_train)
#Predict the response for test dataset
y_pred = clf.predict(X_test)
```

```
#Import scikit-learn metrics module for accuracy calculation
from sklearn import metrics
# Model Accuracy: how often is the classifier correct?
print(“Accuracy:”,metrics.accuracy_score(y_test, y_pred))
```

```
# Model Precision: what percentage of positive tuples are labeled as such?
print(“Precision:”,metrics.precision_score(y_test, y_pred))
# Model Recall: what percentage of positive tuples are labelled as such?
print(“Recall:”,metrics.recall_score(y_test, y_pred))
```

4.10. APPLICATION OF SUPERVISED LEARNING

Supervised Learning Algorithms are used in a variety of applications.

- Bioinformatics: This is one of the most well-known applications of Supervised Learning because most of us use it in our day-to-day lives. Bioinformatics is the storage of Biological Information of us humans such as fingerprints, iris texture, earlobe and so on. Cellphones of today are capable of learning our biological information and are then able to authenticate us bringing up the security of the system. Smartphones such as iPhones, Google Pixel are capable of facial recognition while OnePlus, Samsung is capable of In-display finger recognition.

Figure 4.6. Bioinformatics.

- Speech Recognition: This is the kind of application where you teach the algorithm about your voice and it will be able to recognize you. The most well-known real-world applications are virtual assistants such as Google Assistant and Siri, which will wake up to the keyword with your voice only.

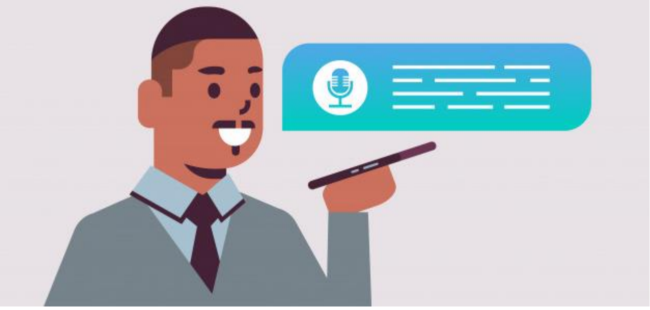


Figure 4.7. Speech recognition.

- Spam Detection: This application is used where the unreal or computer-based messages and E-Mails are to be blocked. G-Mail has an algorithm that learns the different keywords which could be fake such as “You are the winner of something” and so forth and blocks those messages directly. OnePlus Messages App gives the user the task of making the application learn which keywords need to be blocked and the app will block those messages with the keyword.

Figure 4.8. spam detection.

- Object-Recognition for Vision: This kind of application is used when you need to identify something. You have a huge dataset which you use to teach your algorithm and this can be used to recognize a new instance. Raspberry Pi algorithms which detect objects are the most well-known example.

Figure 4.9. Object recognition.

Figure 4.10. House price prediction.

- House prices Prediction: One practical example of supervised learning problems is predicting house prices. First, we need data about the houses: square footage, number of rooms, features, whether a house has a garden or not, and so on. We then need to know the prices of these houses, i.e., the corresponding labels. By leveraging data coming from thousands of houses, their features and prices, we can now train a supervised machine learning model to predict a new house's price based on the examples observed by the model.

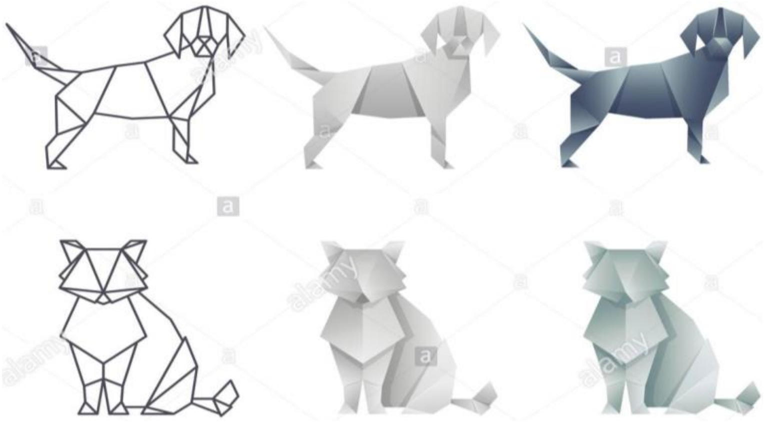


Figure 4.11. Image classification.

- Image Classification: Is it a cat or a dog?: Image classification is a popular problem in the computer vision field. Here, the goal is to predict what class an image belongs to. In this set of problems, we are interested in finding the class label of an image. More precisely: is the image of a car or a plane? A cat or a dog.
- Weather Forecast: How's the weather today?: One particularly interesting problem which requires considering a lot of different parameters is predicting weather conditions in a particular location. To make correct predictions for the weather, we need to take into account various parameters, including historical temperature data, precipitation, wind, humidity, and so on. This particularly interesting and challenging problem may require developing complex supervised models that include multiple tasks. Predicting today's temperature is a regression problem, where the output labels are continuous variables. By contrast, predicting whether it is going to snow or not tomorrow is a binary classification problem.

Figure 4.12. Weather forecast.

- Customer Segmentation: Who are the unhappy customers?: Another great example of supervised learning is text classification problems. In this set of problems, the goal is to predict the class label of a given piece of text. One particularly popular topic in text classification is to predict the sentiment of a piece of text, like a tweet or a product review. This is widely used in the e-commerce industry to help companies to determine negative comments made by customers.

LINKS AND REFERENCES USED IN THIS CHAPTER

Links

1. https://rb.gy/6uxiuk
2. https://rb.gy/palefj
3. Faizan Ahmad, Aaima Najam and Zeeshan Ahmed. Image-based Face Detection and Recognition: "State of the Art".
4. https://www.shutterstock.com/license
5. https://rb.gy/s4anxk
6. https://rb.gy/wmafxn
7. https://rb.gy/w5eb6y
8. https://rb.gy/fi6scz
9. https://rb.gy/rnazpm
10. https://www.freepikcompany.com/legal#nav-freepik-agreement
11. https://www.dreamstime.com/about-stock-image-licenses

References

Ahmad, Faizan, Najam, Aaima and Ahmed, Zeeshan. *Image-based Face Detection and Recognition: "State of the Art"*.

Taylor, C., Michie, D., and Spiegalhalter, D. *Machine Learning, Neural and Statistical Classification,* Paramount Publishing International.

Utgoff, P., "Incremental Induction of Decision Trees," Machine Learning, 4:161–186, Nov., 1989.

Valiant, L., "*A Theory of the Learnable*," Communications of the ACM, Vol. 27, pp. 1134-1142, 1984.

Chapter 5

UNSUPERVISED LEARNING

5.1. INTRODUCTION

Unsupervised Learning is a machine learning technique in which the users do not need to supervise the model. Instead, it allows the model to work on its own to discover patterns and information that was previously undetected. It mainly deals with the unlabelled data.

Unsupervised Learning Algorithms allow users to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning methods. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of a dataset, group that data according to similarities, and represent that dataset in a compressed format.

Unsupervised learning algorithms include clustering, anomaly detection, neural networks, etc. Clustering automatically splits the dataset into groups based on their similarities. Anomaly detection can discover unusual data points in your dataset. It is useful for finding fraudulent transactions. Association mining identifies sets of items which often occur together in your dataset. Latent variable models are widely used for data preprocessing. Like reducing the number of features in a dataset or decomposing the dataset into multiple components

5.2. K-MEANS FOR CLUSTERING PROBLEMS

K means it is an iterative clustering algorithm which helps you to find the highest value for every iteration. Initially, the desired number of clusters are selected. In this clustering method, you need to cluster the data points into k groups. A larger k means smaller groups with more granularity in the same way. A lower k means larger groups with less granularity.

The output of the algorithm is a group of "labels." It assigns data points to one of the k groups. In k-means clustering, each group is defined by creating a centroid for each group. The centroids are like the heart of the cluster, which captures the points closest to them and adds them to the cluster.

K-mean clustering further defines two subgroups:

- Agglomerative clustering
- Dendrogram

Agglomerative clustering: This type of K-means clustering starts with a fixed number of clusters. It allocates all data into the exact number of clusters. This clustering method does not require the number of clusters K as an input. Agglomeration process starts by forming each data as a single cluster. This method uses some distance measure, reducing the number of clusters (one in each iteration) by merging processes. Lastly, we have one big cluster that contains all the objects.

Dendrogram: In the Dendrogram clustering method, each level will represent a possible cluster. The height of the dendrogram shows the level of similarity between two join clusters. The closer to the bottom of the process they are a more similar cluster which is finding the group from a dendrogram which is not natural and mostly subjective.

CODE: The code for Kmean is as mentioned below.

You can find the code here: https://rb.gy/qf31os

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans
```

```
# Configuration options
num_samples_total = 1000
cluster_centers = [(20,20), (4,4)]
num_classes = len(cluster_centers)
```

```
# Generate data
X, targets = make_blobs(n_samples = num_samples_total, centers = cluster_centers, n_features =
num_classes, center_box=(0, 1), cluster_std = 2)
```

```
# Fit K-means with Scikit
kmeans = KMeans(init=‘k-means++’, n_clusters=num_classes, n_init=10)
kmeans.fit(X)
```

```
# Predict the cluster for all the samples
P = kmeans.predict(X)
```

```
# Generate scatter plot for training data
colors = list(map(lambda x: ‘#3b4cc0’ if x == 1 else ‘#b40426’, P))
plt.scatter(X[:,0], X[:,1], c=colors, marker=“o”, picker=True)
plt.title(‘Two clusters of data’)
plt.xlabel(‘Temperature yesterday’)
plt.ylabel(‘Temperature today’)
plt.show()
```

5.3. CLUSTERING

Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms will process your data and find natural clusters(groups) if they exist in the data. You can also modify how many clusters your algorithms should identify. It allows you to adjust the granularity of these groups.

Clustering methods are used to identify groups of similar objects in a multivariate data set collected from fields such as marketing, bio-medical and geo-spatial. They are different types of clustering methods, including:

- Partitioning methods
- Hierarchical clustering
- Fuzzy clustering
- Density-based clustering
- Model-based clustering

There are different types of clustering you can utilize:

5.3.1. Exclusive (Partitioning)

In this clustering method, Data is grouped in such a way that one data can belong to one cluster only.

Example: K-means

5.3.2. Agglomerative

In this clustering technique, every data is a cluster. The iterative unions between the two nearest clusters reduce the number of clusters.

Example: Hierarchical clustering

5.3.3. Overlapping

In this technique, fuzzy sets are used to cluster data. Each point may belong to two or more clusters with separate degrees of membership.

5.4. Principal Component Analysis

Principal Component Analysis (PCA) is an unsupervised statistical technique algorithm. PCA is a “dimensionality reduction” method. It reduces the number of variables that are correlated to each other into fewer independent variables without losing the essence of these variables. It provides an overview of linear relationships between inputs and variables.

PCA helps in Dimensionality reduction. Converts set of correlated variables to non-correlated variables. It finds a sequence of linear combinations of variables.PCA also serves as a tool for better data visualization of high dimensional data. We can create a heat map to show the correlation between each component. It is often used to help in dealing with multi- collinearity before a model is developed. It describes that data is a good story teller of its own.These models are useful in data interpretation and variable selection.

Principal Component Analysis can be used in Image compression. Image can be resized as per the requirement and patterns can be determined. Principal Component Analysis helps in Customer profiling based on demographics as well as their intellect in the purchase. PCA is a technique that is widely used by researchers in the food science field. It can also be used in the Banking field in many areas like applicants applied for loans, credit cards, etc.It can also be used in the Finance field to analyze stocks quantitatively, forecasting portfolio returns, also in the interest rate implantation. PCA is also applied in Healthcare industries in multiple areas like patient insurance data where there are multiple sources of data and with a huge number of variables that are correlated to each other. Sources are like hospitals, pharmacies, etc.

CODE: The code for Principal Component Analysis (PCA) is as mentioned below.

You can find the code here: https://rb.gy/tul2ug

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn import decomposition
from sklearn import datasets
```

```
np.random.seed(5)
```

```
centers = [[1, 1], [-1, -1], [1, -1]]
iris = datasets.load_iris()
X = iris.data
y = iris.target
```

```
fig = plt.figure(1, figsize=(4, 3))
plt.clf()
ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)
plt.cla()
pca = decomposition.PCA(n_components=3)
pca.fit(X)
X = pca.transform(X)
for name, label in [(‘Setosa’, 0), (‘Versicolour’, 1), (‘Virginica’, 2)]:
ax.text3D(X[y == label, 0].mean(),
X[y == label, 1].mean() + 1.5,
X[y == label, 2].mean(), name,
horizontalalignment=‘center’,
bbox=dict(alpha=.5, edgecolor=‘w’, facecolor=‘w’))
# Reorder the labels to have colors matching the cluster results
y = np.choose(y, [1, 2, 0]).astype(float)
ax.scatter(X[:, 0], X[:, 1], X[:, 2], c=y, cmap=plt.cm.nipy_spectral,
edgecolor=‘k’)
ax.w_xaxis.set_ticklabels([])
ax.w_yaxis.set_ticklabels([])
ax.w_zaxis.set_ticklabels([])
plt.show()
```

5.5. Singular Value Decomposition

The singular value decomposition of a matrix A is the factorization of A into the product of three matrices A = (UDV)^T where the columns of U and V are orthonormal and the matrix D is diagonal with positive real entries.

SVD is known under many different names. In the early days, as the above passage implies, it was called, "factor analysis." Other terms include principal component (PC) decomposition and empirical orthogonal function (EOF) analysis. All these are mathematically equivalent, although the way they are treated in the literature is often quite different.

5.6. Independent Component Analysis

Independent Component Analysis (ICA) is a machine learning technique to separate independent sources from a mixed signal. Unlike principal component analysis which focuses on maximizing the variance of the data points, the independent component analysis focuses on independence, i.e., independent components.

ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. In the model, the data variables are assumed to be linear mixtures of some unknown latent variables, and the mixing system is also unknown. The latent variables are assumed non gaussian and mutually independent, and they are called the independent components of the observed data. These independent components, also called sources or factors, can be found by ICA.

ICA is superficially related to principal component analysis and factor analysis. ICA is a much more powerful technique, however, capable of finding the underlying factors or sources when these classic methods fail completely.

The data analyzed by ICA could originate from many different kinds of application fields, including digital images, document databases, economic indicators and psychometric measurements. In many cases, the measurements are given as a set of parallel signals or time series; the term blind source separation is used to characterize this problem. Typical examples are mixtures of simultaneous speech signals that have been picked up by several microphones, brain waves recorded by multiple sensors, interfering radio signals arriving at a mobile phone, or parallel time series obtained from some industrial process.

Independent component analysis (ICA) is used to estimate sources given noisy measurements. Imagine 3 instruments playing simultaneously and 3 microphones recording the mixed signals. ICA is used to recover the sources i.e. what is played by each instrument. Importantly, PCA fails at recovering our instruments since the related signals reflect non-Gaussian processes.

CODE: The code for Independent Component Analysis (ICA) is as mentioned below.

You can find the code here: https://rb.gy/i5va0w

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal
from sklearn.decomposition import FastICA, PCA
```

```
# Generate sample data
np.random.seed(0)
n_samples = 2000
time = np.linspace(0, 8, n_samples)
```

```
s1 = np.sin(2 * time) # Signal 1: sinusoidal signal
s2 = np.sign(np.sin(3 * time)) # Signal 2: square signal
s3 = signal.sawtooth(2 * np.pi * time) # Signal 3: saw tooth signal
```

```
S = np.c_[s1, s2, s3]
S += 0.2 * np.random.normal(size=S.shape) # Add noise
S/= S.std(axis=0) # Standardize data
# Mix data
A = np.array([[1, 1, 1], [0.5, 2, 1.0], [1.5, 1.0, 2.0]]) # Mixing matrix
X = np.dot(S, A.T) # Generate observations
# Compute ICA
ica = FastICA(n_components=3)
```

```
S_ = ica.fit_transform(X) # Reconstruct signals
A_ = ica.mixing_ # Get estimated mixing matrix
# We can ‘prove’ that the ICA model applies by reverting the unmixing.
assert np.allclose(X, np.dot(S_, A_.T) + ica.mean_)
# For comparison, compute PCA
pca = PCA(n_components=3)
H = pca.fit_transform(X) # Reconstruct signals based on orthogonal components
```

```
# Plot results
plt.figure()
models = [X, S, S_, H]
names = [‘Observations (mixed signal)’,
‘True Sources’,
‘ICA recovered signals’,
‘PCA recovered signals’]
colors = [‘red’, ‘steelblue’, ‘orange’]
for ii, (model, name) in enumerate(zip(models, names), 1):
plt.subplot(4, 1, ii)
plt.title(name)
for sig, color in zip(model.T, colors):
plt.plot(sig, color=color)
plt.tight_layout()
plt.show()
```

5.7. APPLICATION OF UNSUPERVISED MACHINE LEARNING

Unsupervised Learning helps in a variety of ways which can be used to solve various real-world problems.

- They help us in understanding patterns which can be used to cluster the data points based on various features.
- Understanding various defects in the dataset which we would not be able to detect initially.

- They help in mapping the various items based on the dependencies of each other.
- Cleansing the datasets by removing features which are not really required for the machine to learn from.

This ultimately leads to applications which are helpful to us. Certain examples of where Unsupervised Learning algorithms are used are discussed below:

- *AirBnB* – This is a great application which helps host stays and experiences connecting people all over the world. This application uses Unsupervised Learning where the user queries his or her requirements and Airbnb learns these patterns and recommends stays and experiences which fall under the same group or cluster.

Figure 5.1. Airbnb.

- *Amazon* – Amazon also uses unsupervised learning to learn the customer's purchase and recommend the products which are most frequently bought together which is an example of association rule mining.

Figure 5.2. Amazon.

- *Credit-Card Fraud Detection* – Unsupervised Learning algorithms learn about various patterns of the user and their usage of the credit card. If the card is used in parts that do not match the behaviour, an alarm is generated which could possibly be marked fraud and calls are given to you to confirm whether it was you using the card or not.

Figure 5.3. Credit card fraud detection.

Links and References Used in This Chapter

Links

1. https://rb.gy/qf31os
2. https://www.gettyimages.in/eula?utm_medium=organic&utm_source=google&utm_campaign=iptcurl
3. https://www.gettyimages.in/eula?utm_medium=organic&utm_source=google&utm_campaign=iptcurl
4. https://rb.gy/i5va0w
5. https://rb.gy/tul2ug

References

Shavlik, J. and Dietterich, T. *Readings in Machine Learning,* San Francisco: Morgan Kaufmann, 1990.

Sutton, R. S. "Learning to Predict by the Methods of Temporal Differences," *Machine Learning* 3: 9-44, 1988

Tesauro, G. *Practical Issues in Temporal Difference Learning*, Machine Learning, 8, nos. 3/4, pp. 257-277, 1992.

Chapter 6

REINFORCEMENT LEARNING

6.1. INTRODUCTION

Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty. In Reinforcement Learning, the agent learns automatically using feedback without any labeled data, unlike supervised learning. Since there is no labeled data, the agent is bound to learn by its experience only. RL solves a specific type of problem where decision making is sequential, and the goal is long-term, such as game-playing, robotics, etc. The agent interacts with the environment and explores it by itself. The primary goal of an agent in reinforcement learning is to improve the performance by getting the maximum positive rewards.

The agent learns with the process of hit and trial, and based on the experience, it learns to perform the task in a better way. Hence, we can say that "Reinforcement learning is a type of machine learning method where an intelligent agent (computer program) interacts with the environment and learns to act within that." How a Robotic dog learns the movement of his arms is an example of Reinforcement learning. It is a core part of Artificial intelligence, and all AI agents work on the concept of reinforcement learning. Here we do not need to pre-program the agent, as it learns from its own experience without any human intervention.

Example: Suppose there is an AI agent present within a maze environment, and his goal is to find the diamond. The agent interacts with the environment by performing some actions, and based on those actions, the state of the agent gets changed, and it also receives a reward or penalty as feedback. The agent continues doing these three things (take action, change state/remain in the same state, and get feedback), and by doing these actions, he learns and explores the environment. The agent learns what actions lead to positive feedback or rewards and what actions lead to negative feedback penalty. As a positive reward, the agent gets a positive point, and as a penalty, it gets a negative point.

6.2. Terms Used in Reinforcement Learning

- Value(): It is expected long-term retuned with the discount factor and opposite to the short-term reward.
- Reward(): A feedback returned to the agent from the environment to evaluate the action of the agent.
- Policy(): Policy is a strategy applied by the agent for the next action based on the current state.
- Environment(): A situation in which an agent is present or surrounded by. In RL, we assume the stochastic environment, which means it is random in nature.
- Action(): Actions are the moves taken by an agent within the environment.
- State(): State is a situation returned by the environment after each action taken by the agent.
- Agent(): An entity that can perceive/explore the environment and act upon it.

- Q-value(): It is mostly similar to the value, but it takes one additional parameter as a current action

6.3. Key Feature of Reinforcement Learning

- In RL, the agent is not instructed about the environment and what actions need to be taken.
- It is based on the hit and trial process.
- The agent takes the next action and changes states according to the feedback of the previous action.
- The agent may get a delayed reward.
- The environment is stochastic, and the agent needs to explore it to get the maximum positive rewards.

6.4. Elements of Reinforcement Learning

There are four main elements of Reinforcement Learning, which are given below:

1. Policy
2. Reward Signal
3. Value Function
4. Model of the environment

1) Policy: A policy can be defined as a way how an agent behaves at a given time. It maps the perceived states of the environment to the actions taken on those states. A policy is the core element of the RL as it alone can define the behavior of the agent. In some cases, it may be a simple function or a lookup table, whereas, for other cases, it may involve general computation as a search process.

2) Reward Signal: The goal of reinforcement learning is defined by the reward signal. At each state, the environment sends an immediate signal to the learning agent, and this signal is known as a reward signal. These rewards are given according to the good and bad actions taken by the agent. The agent's main objective is to maximize the total number of rewards for good actions. The reward signal can change the policy, such as if an action selected by the agent leads to low reward, then the policy may change to select other actions in the future.
3) Value Function: The value function gives information about how good the situation and action are and how much reward an agent can expect. A reward indicates the immediate signal for each good and bad action, whereas a value function specifies the good state and action for the future. The value function depends on the reward as, without reward, there could be no value. The goal of estimating values is to achieve more rewards.
4) Model: The last element of reinforcement learning is the model, which mimics the behavior of the environment. With the help of the model, one can make inferences about how the environment will behave. Such as, if a state and an action are given, then a model can predict the next state and reward.

The model is used for planning, which means it provides a way to take a course of action by considering all future situations before actually experiencing those situations. The approaches for solving the RL problems with the help of the model are termed as the model-based approach. Comparatively, an approach without using a model is called a model-free approach.

6.5. HOW DOES REINFORCEMENT LEARNING WORKS?

To understand the working process of the RL, we need to consider two main things:

- Environment: It can be anything such as a room, maze, football ground, etc.
- Agent: An intelligent agent such as AI robot.

Let's take an example of a maze environment that the agent needs to explore. Consider the below image

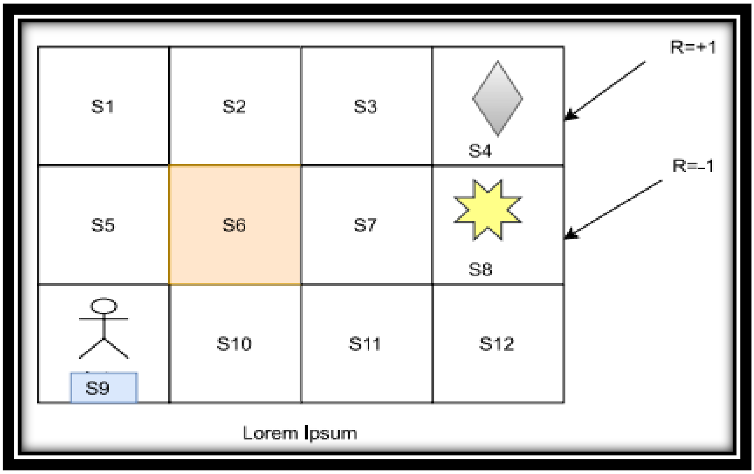


Figure 6.1. Reinforcement Learning working.

In the above image, the agent is at the very first block of the maze. The maze is consisting of an S_6 block, which is a wall, S_8 a fire pit, and S_4 a diamond block.

The agent cannot cross the S_6 block, as it is a solid wall. If the agent reaches the S_4 block, then get the +1 reward; if it reaches the fire pit, then gets -1 reward point. It can take four actions: move up, move down, move left, and move right.

The agent can take any path to reach the final point, but he needs to make it in possibly fewer steps. Suppose the agent considers the path S9-S5-S1-S2-S3, so he will get the +1-reward point.

The agent will try to remember the preceding steps that it has taken to reach the final step. To memorize the steps, it assigns 1 value to each previous step. Consider the below step:

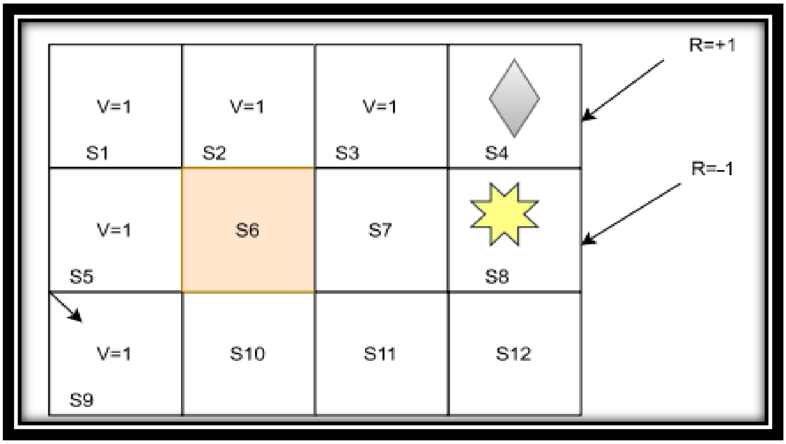


Figure 6.2. Reinforcement Learning working.

Now, the agent has successfully stored the previous steps assigning the 1 value to each previous block. But what will the agent do if he starts moving from the block, which has 1 value block on both sides? Consider the below diagram:

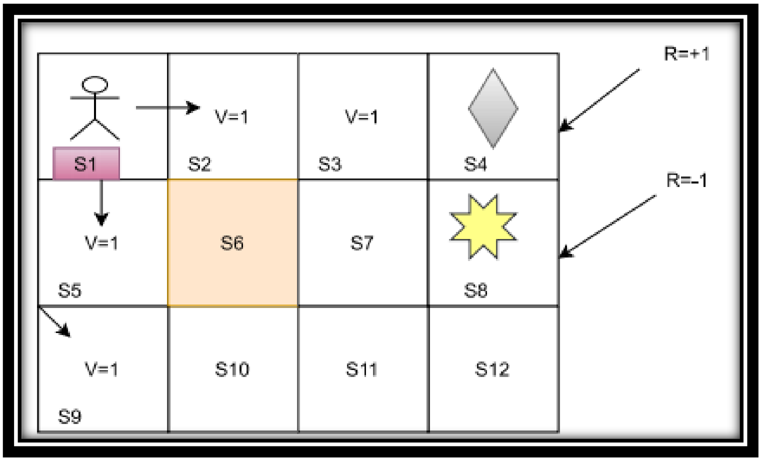


Figure 6.3. Reinforcement Learning working.

It will be a difficult condition for the agent whether he should go up or down as each block has the same value. So, the above approach is not suitable for the agent to reach the destination. Hence to solve the problem, we will use the Bellman equation, which is the main concept behind reinforcement learning.

6.6. TYPES OF REINFORCEMENT LEARNING

There are mainly two types of reinforcement learning, which are:

- Positive Reinforcement
- Negative Reinforcement

6.6.1. Positive Reinforcement

The positive reinforcement learning means adding something to increase the tendency that expected behavior would occur again. It impacts positively on the behavior of the agent and increases the strength of the behavior.

This type of reinforcement can sustain the changes for a long time, but too much positive reinforcement may lead to an overload of states that can reduce the consequences.

6.6.2. Negative Reinforcement

The negative reinforcement learning is opposite to the positive reinforcement as it increases the tendency that the specific behavior will occur again by avoiding the negative condition.

It can be more effective than positive reinforcement depending on situation and behavior, but it provides reinforcement only to meet minimum behavior.

6.7. Markov Decision Process

Markov Decision Process or MDP, is used to formalize the reinforcement learning problems. If the environment is completely observable, then its dynamic can be modeled as a Markov Process. In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state.

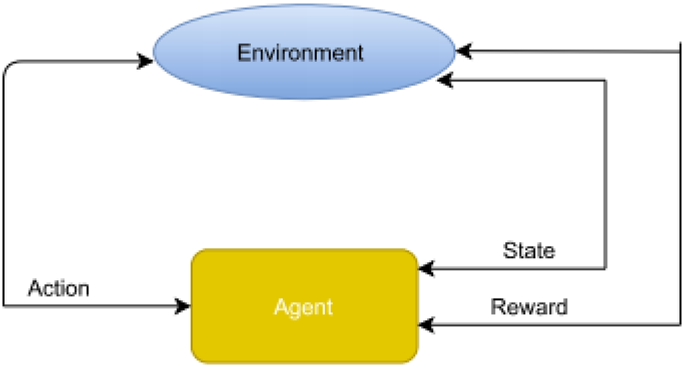


Figure 6.4. Markov decision process.

MDP is used to describe the environment for the RL, and almost all the RL problem can be formalized using MDP.

MDP contains a tuple of four elements (S, A, P_a, R_a):

- A set of finite States S
- A set of finite Actions A
- Rewards received after transitioning from state S to state S’, due to action a.
- Probability P_a.

MDP uses Markov property, and to better understand the MDP, we need to learn about it.

6.7.1. Markov Property

It says that “If the agent is present in the current state S1, performs an action a1 and moves to the state s2, then the state transition from s1 to s2 only depends on the current state and future action and states do not depend on past actions, rewards, or states.”

Or, In other words, as per Markov Property, the current state transition does not depend on any past action or state. Hence, MDP is an RL problem that satisfies the Markov property. Such as in a Chess game, the players only focus on the current state and do not need to remember past actions or states.

6.8. Reinforcement Learning Algorithm

Reinforcement learning algorithms are mainly used in AI applications and gaming applications. The main used algorithms are:

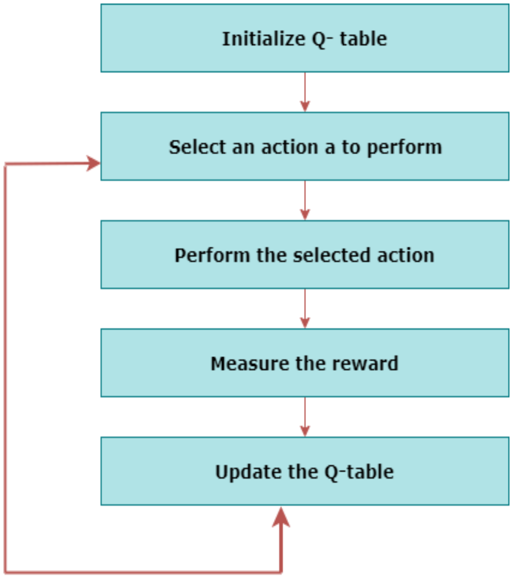


Figure 6.5. Q-learning.

- Q-Learning:
    - Q-learning is an Off policy RL algorithm, which is used for the temporal difference Learning. The temporal difference learning methods are the way of comparing temporally successive predictions.
    - It learns the value function Q (S, a), which means how good to take action “a” at a particular state “s.”
    - Figure 6.5 chart explains the working of Q- learning.
- State Action Reward State action (SARSA):
    - SARSA stands for State Action Reward State action, which is an on-policy temporal difference learning method. The on-policy control method selects the action for each state while learning using a specific policy.
    - The goal of SARSA is to calculate the Q π (s, a) for the selected current policy π and all pairs of (s-a).
    - The main difference between Q-learning and SARSA algorithms is that unlike Q-learning, the maximum reward for the next state is not required for updating the Q-value in the table.
    - In SARSA, new actions and rewards are selected using the same policy, which has determined the original action.

      The SARSA is named because it uses the quintuple Q (s, a, r, s’, a’). Where,
      s: original state
      a: Original action
      r: reward observed while following the states
      s’ and a’: New state, action pair.
- Deep Q Neural Network (DQN):
    - As the name suggests, DQN is a Q-learning using Neural networks.
    - For a big state space environment, it will be a challenging and complex task to define and update a Q-table.

- To solve such an issue, we can use a DQN algorithm. Where, instead of defining a Q-table, the neural network approximates the Q-values for each action and state.

6.9. Q-LEARNING

- Q-learning is a popular model-free reinforcement learning algorithm based on the Bellman equation.
- The main objective of Q-learning is to learn the policy which can inform the agent that what actions should be taken for maximizing the reward under what circumstances.
- It is an off-policy RL that attempts to find the best action to take at a current state.
- The goal of the agent in Q-learning is to maximize the value of Q.
- The value of Q-learning can be derived from the Bellman equation. Consider the Bellman equation given below:

$$V(s) = \max [R(s,a) + \gamma \textstyle\sum_{s'} P(s, a, s')V(s`)]$$

In the equation, we have various components, including reward, discount factor (γ), probability, and end states s’. But there is no any Q-value is given so first consider the below image:

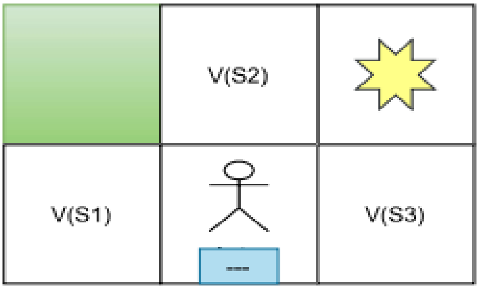


Figure 6.6. Q-learning.

In the above image, we can see there is an agent who has three values options, $V(s_1)$, $V(s_2)$, $V(s_3)$. As this is MDP, the agent only cares for the current state and the future state. The agent can go in any direction (Up, Left, or Right), so he needs to decide where to go for the optimal path. Here the agent will take a move as per probability bases and change the state. But if we want some exact moves, so for this, we need to make some changes in terms of Q-value. Consider the below image:

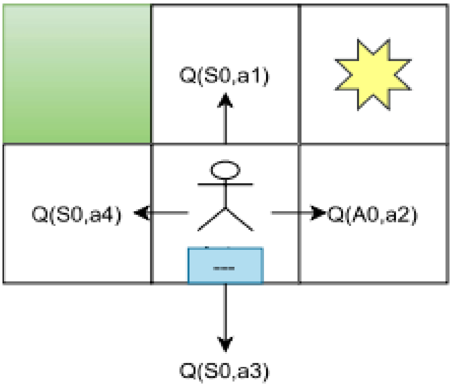


Figure 6.7. Q-learning.

Q- represents the quality of the actions at each state. So instead of using a value at each state, we will use a pair of states and action, i.e., Q(s, a). Q-value specifies that which action is more lubricated than others, and according to the best Q-value, the agent takes his next move. The Bellman equation can be used for deriving the Q-value.

To perform any action, the agent will get a reward R(s, a), and also he will end up on a certain state, so the Q -value equation will be:

$$Q(S, a)= R(s, a)+ \gamma\sum_{s'} P(s, a, s')V(s`)$$

Hence, we can say that, $V(s) = max [Q(s, a)]$

$Q(S,a)= R(s,a)=y\sum s'(P(s,a,s')maxQ(s',a'))$

The above formula is used to estimate the Q-values in Q-Learning.

6.9.1. What is ‘Q’ in Q-Learning?

The Q stands for quality in Q-learning, which means it specifies the quality of an action taken by the agent.

6.9.2. Q-Table

A Q-table or matrix is created while performing the Q-learning. The table follows the state and action pair, i.e., [s, a], and initializes the values to zero. After each action, the table is updated, and the q-values are stored within the table.

The RL agent uses this Q-table as a reference table to select the best action based on the q-values.

6.10. DIFFERENCE BETWEEN REINFORCEMENT LEARNING AND SUPERVISED LEARNING

Table 6.1. Difference between Reinforcement and supervised learning

Reinforcement Learning	**Supervised Learning**
RL works by interacting with the environment.	Supervised learning works on the existing dataset.
The RL algorithm works like the human brain works when making some decisions.	Supervised Learning works as when a human learns things in the supervision of a guide.
There is no labeled dataset is present	The labeled dataset is present.
No previous training is provided to the learning agent.	Training is provided to the algorithm so that it can predict the output.
RL helps to take decisions sequentially.	In Supervised learning, decisions are made when input is given.

6.11. Reinforcement Learning Application

The application of Reinforcement Learning are:

1. Application in Self driving car: In self-driving cars, there are various aspects to consider, such as speed limits at various places, drivable zones, avoiding collisions. Some of the autonomous driving tasks where reinforcement learning could be applied include trajectory optimization, motion planning, dynamic pathing, controller optimization, and scenario-based learning policies for highways. For example, parking can be achieved by learning automatic parking policies. Lane changing can be achieved using Q-Learning while overtaking can be implemented by learning an overtaking policy while avoiding collision and maintaining a steady speed thereafter. AWS Deep Racer is an autonomous racing car that has been designed to test out RL in a physical track. It uses cameras to visualize the runway and a reinforcement learning model to control the throttle and direction.

Figure 6.8. Self driving vehicle.

2. Industry automation with Reinforcement Learning: In industry reinforcement, learning-based robots are used to perform various tasks. Apart from the fact that these robots are more efficient than human beings, they can also perform tasks that would be dangerous for people. A great example is the use of AI agents by

Deep mind to cool Google Data Centers. This led to a 40% reduction in energy spending. The centers are now fully controlled with the AI system without the need for human intervention. There is obviously still supervision from data center experts.

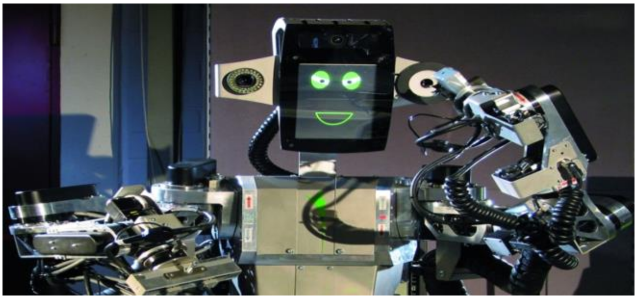


Figure 6.9. Industry Automation.

3. Reinforcement Learning applications in trading and finance: Supervised time series models can be used for predicting future sales as well as predicting stock prices. However, these models don’t determine the action to take at a particular stock price. Enter Reinforcement Learning (RL). An RL agent can decide on such a task; whether to hold, buy, or sell. The RL model is evaluated using market benchmark standards in order to ensure that it’s performing optimally. This automation brings consistency into the process, unlike previous methods where analysts would have to make every single decision. IBM for example has a sophisticated reinforcement learning based platform that has the ability to make financial trades. It computes the reward function based on the loss or profit of every financial transaction.
4. Reinforcement Learning applications in healthcare: In healthcare, patients can receive treatment from policies learned from RL systems. RL is able to find optimal policies using previous experiences without the need for previous information on the mathematical model of biological systems. It makes this approach more applicable than other control-based systems in healthcare. RL in healthcare is categorized as dynamic treatment regimens (DTRs)

in chronic disease or critical care, automated medical diagnosis, and other general domains.

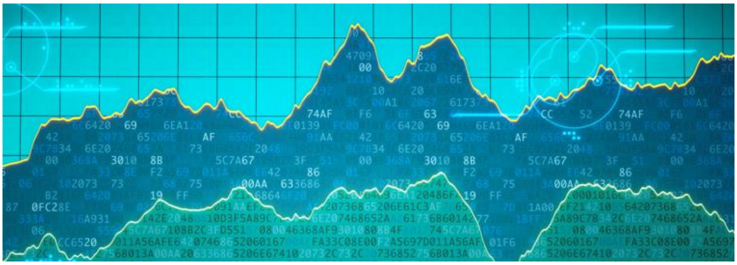


Figure 6.10. Trading & finance.

5. Reinforcement Learning in news recommendation:-User preferences can change frequently, therefore recommending news to users based on reviews and likes could become obsolete quickly. With reinforcement learning, the RL system can track the reader’s return behaviors. Construction of such a system would involve obtaining news features, reader features, context features, and reader news features. News features include but are not limited to the content, headline, and publisher. Reader features refer to how the reader interacts with the content e.g. clicks and shares. Context features include news aspects such as timing and freshness of the news. A reward is then defined based on these user behaviors.

Figure 6.11. Healthcare.

Figure 6.12. News recommendation.

6. Reinforcement Learning in robotics manipulation: The use of deep learning and reinforcement learning can train robots that have the ability to grasp various objects—even those unseen during training. This can, for example, be used in building products in an assembly line. This is achieved by combining large-scale distributed optimization and a variant of deep Q-Learning called QT-Opt. QT-Opt support for continuous action spaces makes it suitable for robotics problems. A model is first trained offline and then deployed and fine-tuned on the real robot. Google AI applied this approach to robotics grasping where 7 real-world robots ran for 800 robot hours in a 4-month period.

Figure 6.13. Robotics.

Links and References Used in This Chapter

Links

1. https://www.istockphoto.com/legal/license-agreement?utm_medium=organic&utm_source=google&utm_campaign=iptcurl
2. https://www.istockphoto.com/legal/license-agreement?utm_medium=organic&utm_source=google&utm_campaign=iptcurl

References

Leslie Pack Kaelbling, Andrew W. Moore. Reinforcement Learning: A Survey. *Journal of Articial Intelligence Research* 4 (1996) 237-285.

Russell, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, Englewood Cliffs, NJ: Prentice Hall, 1995.

Schwartz, A., "A Reinforcement Learning Method for Maximizing Undiscounted Rewards," *Proc. Tenth Intl. Conf. on Machine Learning*, pp. 298-305, San Francisco: Morgan Kaufmann, 1993.

Watkins, C. J. C. H., and Dayan, P., "Technical Note: Q-Learning," *Machine Learning*, 8, 279-292, 1992.

Chapter 7

KERNEL MACHINES

7.1. INTRODUCTION

A kernel is a similarity function. It is a function that you, as the domain expert, provide to a machine learning algorithm. It takes two inputs and spits out how similar they are.

Suppose your task is to learn to classify images. You have (image, label) pairs as training data. Consider the typical machine learning pipeline: you take your images, you compute features, you string the features for each image into a vector, and you feed these “feature vectors” and labels into a learning algorithm.

Data --> Features --> Learning algorithm

Kernels offer an alternative. Instead of defining a slew of features, you define a single kernel function to compute similarity between images. You provide this kernel, together with the images and labels to the learning algorithm, and out comes a classifier.

Kernel machines are a class of algorithms for pattern analysis, whose best-known member is the support vector machine (SVM). The general task of pattern analysis is to find and study general types of relations (for example clusters, rankings, principal components, correlations, classify-cations) in datasets. For many algorithms that solve these tasks, the data in raw representation have to be explicitly transformed into feature vector representations via a user-specified feature map: in contrast, kernel methods require only a user-specified kernel, i.e., a similarity function over pairs of data points in raw representation.

7.2. Kernel Methods

Kernel methods are a class of algorithms for patter analysis or recognition whose best-known element is the support vector machine SVM). The general task of pattern analysis is to find and study general types of relations (such as clusters, ranking, principal components, correlations, classifications) in general types of data (such as sequences, text documents, sets of points, vectors, images, graphs, etc).

The main characteristic of Kernel Methods, however, is their distinct approach to this problem. Kernel methods map the data into higher dimensional spaces in the hope that in this higher-dimensional space the data could become more easily separated or better structured. There are also no constraints on the form of this mapping, which could even lead to infinite-dimensional spaces. This mapping function, however, hardly needs to be computed because of a tool called the Kernel Trick.

7.3. Optimal Separating Hyperplane (OSH)

The optimal separating hyperplane is the one that correctly classifies all the data while being farthest away from the data points. It is said to be the hyperplane that maximizes the margin, defined as the distance from the hyperplane to the closest data point.

The idea behind the optimality of this classifier can be illustrated as follows. New test points are drawn according to the same distribution as the training data. Thus, if the separating hyperplane is far away from the data points, previously unseen test points will most likely fall far away from the hyperplane or in the margin. As a consequence, the larger the margin is, the less likely the points are to fall on the wrong side of the hyperplane.

The optimal separating hyperplane is one of the core ideas behind the support vector machines. In particular, it gives rise to the so-called support vectors which are the data points lying on the margin boundary of the hyperplane. These points support the hyperplane in the sense that they contain all the required information to compute the hyperplane: removing other points does not change the optimal separating hyperplane.

7.4. Kernel Trick

A kernel is a method of placing a two-dimensional plane into a higher dimensional space, so that it is curved in the higher dimensional space. (In simple terms, a kernel is a function from the low dimensional space into a higher dimensional space.)

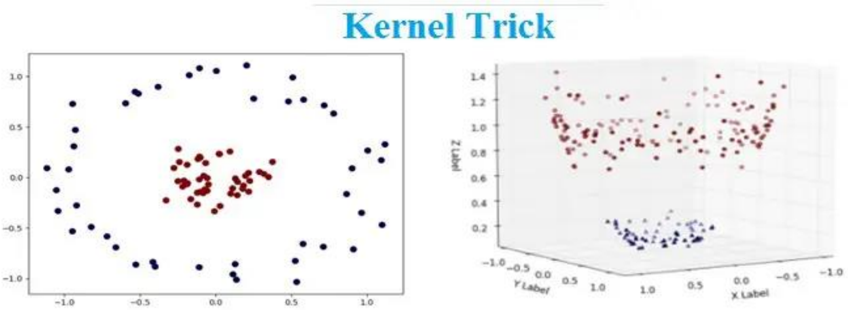


Figure 7.1. Kernel Tricks.

After understanding the above it is now easier to relate and understand the Kernel Trick. We now know that SVM works better with two-dimensional space which are linearly separable.

However, for non-linear data SVM finds it difficult to classify the data. The easy solution here is to use the Kernel Trick. A Kernel Trick is a simple method where a Non-Linear data is projected onto a higher dimension space so as to make it easier to classify the data where it could be linearly divided by a plane.

7.5. Kernel Regression

Kernel values are used to derive weights to predict outputs from given inputs. Steps involved to calculate weights and finally to use them in predicting output variable, *y* from predictor variable, *x* is explained in detail in the following sections. Let’s start with an example to clearly understand how kernel regression works.

Example:

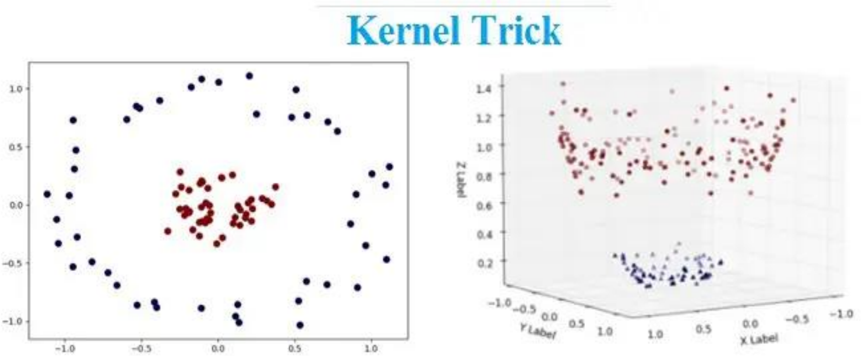


Figure 7.2. Kernel Regression.

A kernel regression model is developed to predict river flow from the catchment area. As shown in the data below, there exists a non-linear relationship between catchment area (in square mile) and river flow (in cubic feet per sec). The output, *y* is the river flow and input, *x* is the catchment area in this example.

7.6. Kernel Dimensionality Reduction

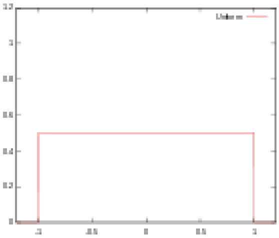
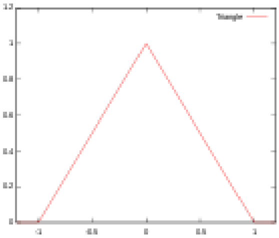
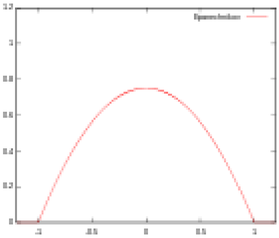
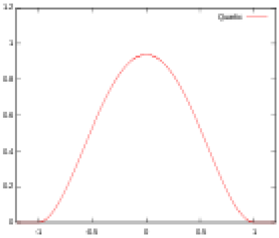
Many statistical learning problems involve some form of dimensionality reduction. The goal may be one of feature selection, in which the aim is to find linear or nonlinear combinations of the original set of variables, or one of variable selection, in which we wish to select a subset of variables from the original set. Motivations for such dimensionality reduction include providing a simplified explanation and visualization for a human, suppressing noise so as to make a better prediction or decision, or reducing the computational burden.

Dimensionality Reduction for supervised learning studied, in which the data consists of (X, Y) pairs, where X is an m-dimensional explanatory variable and Y is an -dimensional response. The variable Y may be either continuous or discrete. We refer to these problems generically as "regression," which indicates our focus on the conditional probability density pY |X(y|x). Thus, the framework includes classification problems, where Y is discrete.

7.7. Kernel Function

The kernel function is what is applied on each data instance to map the original non-linear observations into a higher-dimensional space in which they become separable.

Table 7.1. Kernel Efficiency

Kernel Functions, *K*(*u*)			**Efficiency relative to the Epanechnikov kernel**
Uniform (“rectangular window”)	Support:	 “Boxcar function“	92.9%
Triangular	Support:		98.6%
Epanechnikov (parabolic)	Support:		100%
Quartic (biweight)	Support:		99.4%

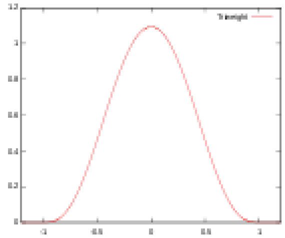
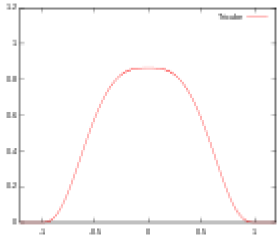
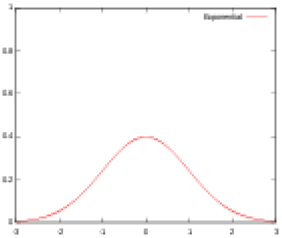
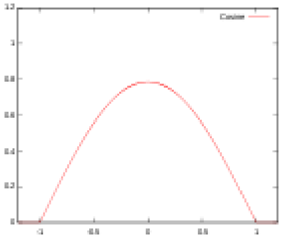
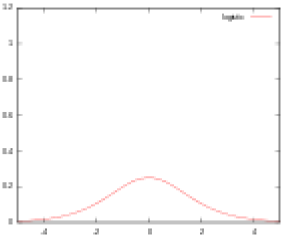
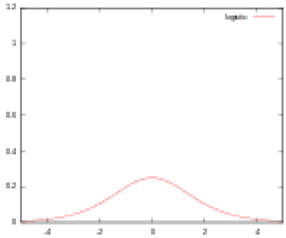
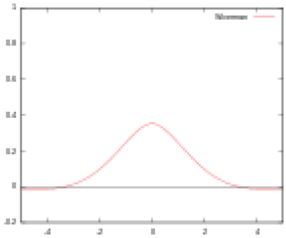
Kernel Functions, *K*(*u*)			**Efficiency relative to the Epanechnikov kernel**
Triweight	Support:		98.7%
Tricube	Support:		99.8%
Gaussian			95.1%
Cosine	Support:		99.9%
Logistic			88.7%

Table 7.1. (Continued)

Kernel Functions, *K*(*u*)			**Efficiency[4] relative to the Epanechnikov kernel**
Sigmoid function			84.3%
Silverman kernel[5]			not applicable

7.8. Kernel Properties

Kernel functions must be continuous, symmetric, and most preferably should have a positive (semi-) definite Gram matrix. Kernels which are said to satisfy the Mercer's Theorem are positive semi-definite, meaning their kernel matrices have only non-negative Eigen values. The use of a positive definite kernel ensures that the optimization problem will be convex and solution will be unique.

However, many kernel functions which aren't strictly positive definite also have been shown to perform very well in practice. An example is the Sigmoid kernel, which, despite its wide use, it is not positive semi-definite for certain values of its parameters.

7.9. Choosing the Right Kernel

Choosing the most appropriate kernel highly depends on the problem at hand and fine tuning its parameters can easily become a tedious and cumbersome task.

The choice of a Kernel depends on the problem at hand because it depends on what we are trying to model. A polynomial kernel, for example, allows us to model feature conjunctions up to the order of the polynomial. Radial basis functions allow to pick out circles (or hyperspheres) in contrast with the Linear kernel, which allows only to pick out lines (or hyperplanes).

The motivation behind the choice of a particular kernel can be very intuitive and straightforward depending on what kind of information we are expecting to extract about the data.

References

Comaniciu, D., Meer, P. (2002). “Mean shift: A robust approach toward feature space analysis”. IEEE Transactions on Pattern Analysis and Machine Intelligence. 24 (5): 603–619. C iteSeerX 1 0.1.1.76.8968. d oi:1 0.1109/34.1000236

Li, Qi, Racine, Jeffrey S. (2007). *Nonparametric Econometrics: Theory and Practice.* Princeton University Press. ISBN: 978-0-691-12161-1.

Ross, S. *Introduction to Stochastic Dynamic Programming,* New York: Academic Press, 1983.

Valiant, L. “*A Theory of the Learnable*,” Communications of the ACM, Vol. 27, pp. 1134-1142, 1984.

Zucchini, Walter. “*Applied Smoothing Techniques Part 1: Kernel Density Estimation*” (PDF). Retrieved 6 September 2018.

Chapter 8

DATA VISUALIZATION

8.1. WHAT IS DATA VISUALIZATION

Figure 8.1. Data visualization.

Data visualization is the creation of visual representations of data. These representations clearly communicate insights from data through charts and graphs. In terms of business intelligence (BI), these visualizations help users make better data-based decisions.

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns.

According to Simon Samuel, "Data visualization is going to change the way our analysts work with data. They're going to be expected to respond to issues more rapidly. And they'll need to be able to dig for more insights – look at data differently, more imaginatively. Data visualization will promote that creative data exploration."

8.2. WHY TO USE DATA VISUALIZATION?

The best reason to use a visualization to understand your data is that most data sets are far too large to consume in their raw format. Humans are limited in what information we can process and compare in our heads, especially if that information resides in a million-row data set, but we are good at quickly processing visual information.

Figure 8.2. Types of data visualization.

Even if you are new to reading data in a chart, you already have the built-in capabilities to spot light and dark colors, large and small shapes, groups and orientations of objects. These are referred to as pre-attentive attributes.

Some pre-attentive attributes are better at showing quantitative (measured) data, and some are better at showing qualitative (categorical) data. For example, using length bars to represent the amount of sales is an effective choice to indicate differences in sales across categories.

8.3. Types of Data Visualization

8.3.1. Temporal

Data visualizations belong in the temporal category if they satisfy two conditions: that they are linear, and that they are one-dimensional. Temporal visualizations normally feature lines that either stand alone or overlap with each other, with a start and finish time.

Examples of temporal data visualization include:

- Scatter plots
- Polar area diagrams
- Time series sequences
- Timelines
- Line graphs

8.3.2.Hierarchical

Data visualizations that belong in the hierarchical category are those that order groups within larger groups. Hierarchical visualizations are best suited if you're looking to display clusters of information, especially if they flow from a single origin point.

THE downside to these graphs is that they tend to be more complex and difficult to read, which is why the tree diagram is used most often. It is the simplest to follow due to its linear path.

Examples of hierarchical data visualizations include:

- Tree diagrams
- Ring charts
- Sunburst diagrams

8.3.3. Network

Datasets connect deeply with other datasets. Network data visualizations show how they relate to one another within a network. In other words, demonstrating relationships between datasets without wordy explanations.

Examples of network data visualizations include:

- Matrix charts
- Node-link diagrams
- Word clouds
- Alluvial diagrams

8.3.4. Multidimensional

Just like the name, multidimensional data visualizations have multiple dimensions. This means that there are always 2 or more variables in the mix to create a 3D data visualization. Because of the many concurrent layers and datasets, these types of visualizations tend to be the most vibrant or eye-catching visuals. These visuals can break down a ton of data down to key takeaways.

Examples of multidimensional data visualizations include:

- Scatter plots
- Pie charts
- Venn diagrams
- Stacked bar graphs
- Histograms

8.3.5. Geospatial

Geospatial or spatial data visualizations relate to real life physical locations, overlaying familiar maps with different data points. These types of data visualizations are commonly used to display sales or acquisitions over time, and can be most recognizable for their use in political campaigns or to display market penetration in multinational corporations.

Examples of geospatial data visualizations include:

- Flow map
- Density map
- Cartogram
- Heat map

8.4. COMMON GRAPH TYPES

8.4.1. Bar Chart

The below bar chart is drawn between Cost of campaigns v/s Time period. At some point or another, you’ve either seen, interacted with, or built a bar chart before. Bar charts are such a popular graph visualization because of how easy you can scan them for quick information. Bar charts organize data into rectangular bars that make it a breeze to compare related data sets.

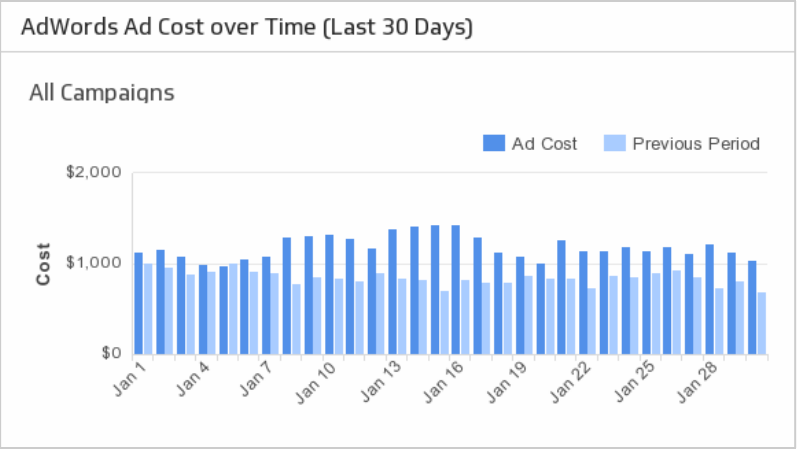


Figure 8.3. Bar chart.

When Do I Use a Bar Chart Visualization?

Use a bar chart for the following reasons:

- You want to compare two or more values in the same category
- You want to compare parts of a whole
- You don't have too many groups (less than 10 works best)
- You want to understand how multiple similar data sets relate to each other

Don't use a bar chart for the following reasons:

- The category you're visualizing only has one value associated with it
- You want to visualize continuous data

Best Practices for a Bar Chart Visualization

If you use a bar chart, here are the key design best practices:

- Use consistent colors and labeling throughout so that you can identify relationships more easily

- Simplify the length of the y-axis labels and don't forget to start from 0 so you can keep your data in order

8.4.2. Line Chart

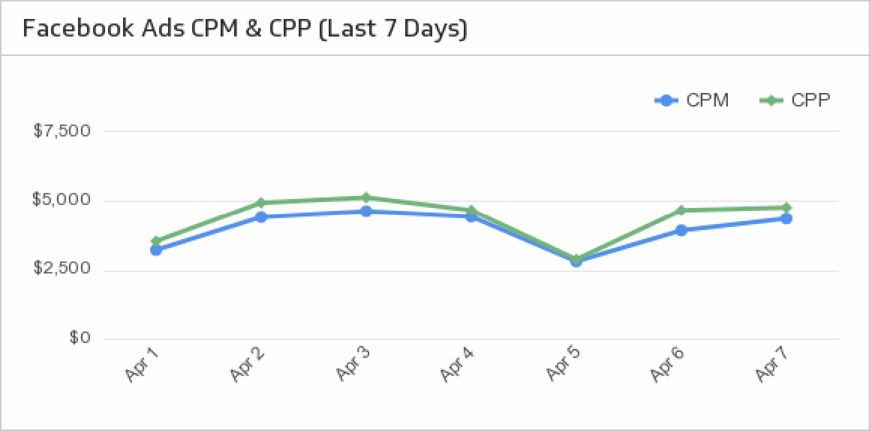


Figure 8.4. Line chart.

The above Line Chart is also drawn between Cost of campaigns v/s Time period. Like bar charts, line charts help to visualize data in a compact and precise format which makes it easy to rapidly scan information in order to understand trends. Line charts are used to show resulting data relative to a continuous variable - most commonly time or money. The proper use of color in this visualization is necessary because different colored lines can make it even easier for users to analyze information.

When Do I Use a Line Chart Visualization?

Use a line chart for the following reasons:

- You want to understand trends, patterns, and fluctuations in your data
- You want to compare different yet related data sets with multiple series

- You want to make projections beyond your data

Don’t use a line chart for the following reason:

- You want to demonstrate an in-depth view of your data

Best Practices for a Line Chart Visualization

If you use a line chart, here are the key design best practices:

- Along with using a different color for each category you’re comparing, make sure you also use solid lines to keep the line chart clear and concise
- To avoid confusion, try not to compare more than 4 categories in one-line chart

8.4.3. Scatterplot

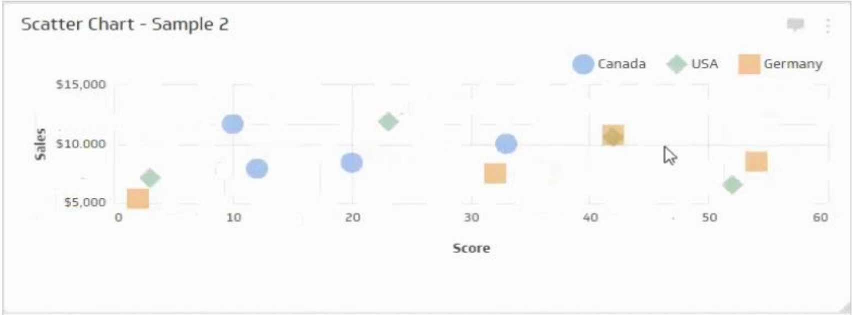


Figure 8.5. Scatterplot.

The above scatterplot is drawn between sales v/s score. Scatterplots are the right data visualizations to use when there are many different data points, and you want to highlight similarities in the data set. This is useful when looking for outliers or for understanding the distribution of your data.

If the data forms a band extending from lower left to upper right, there most likely a positive correlation between the two variables. If the band runs from upper left to lower right, a negative correlation is probable. If it is hard to see a pattern, there is probably no correlation.

When Do I Use a Scatter Plot Visualization?

Use a scatterplot for the following reasons:

- You want to show the relationship between two variables
- You want a compact data visualization

Don't use a scatterplot for the following reasons:

- You want to rapidly scan information
- You want clear and precise data points

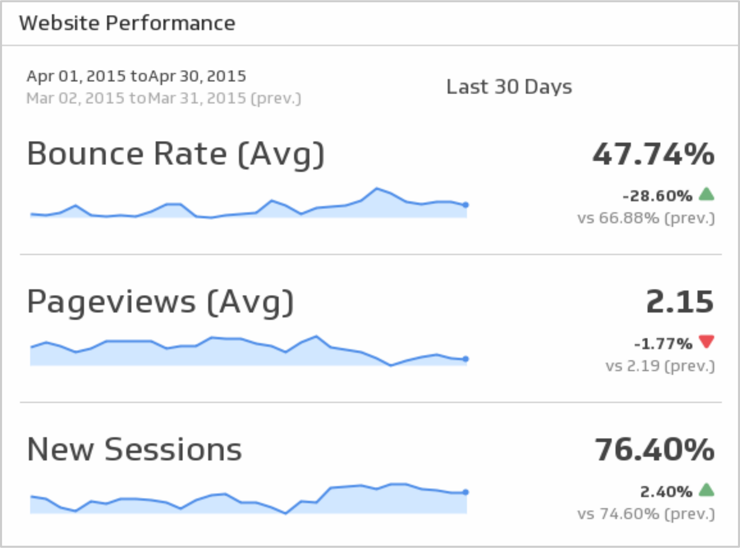
Best Practices for a Scatter Plot Visualization

If you use a scatter plot, here are the key design best practices:

- Although trend lines are a great way to analyze the data on a scatter plot, ensure you stick to 1 or 2 trend lines to avoid confusion
- Don't forget to start at 0 for the y-axis

8.4.4. Sparkline

Sparklines are arguably the best data visualization for showing trends because of how compact they are. They get the job done when it comes to painting a picture for your audience fast. Though, it is important to make sure your audience understands how to read sparklines correctly to optimize their use.



Website Performance

Apr 01, 2015 toApr 30, 2015
Mar 02, 2015 toMar 31, 2015 (prev.)

Last 30 Days

Bounce Rate (Avg) 47.74%
-28.60% ▲
vs 66.88% (prev.)

Pageviews (Avg) 2.15
-1.77% ▼
vs 2.19 (prev.)

New Sessions 76.40%
2.40% ▲
vs 74.60% (prev.)

Figure 8.6. Sparkline.

When Do I Use a Sparkline Visualization?

Use a sparkline for the following reasons:

- You can pair it with a metric that has a current status value tracked over a specific time period
- You want to show a specific trend behind a metric

Don't use a sparkline for the following reasons:

- You want to plot multiple series
- You want to illustrate precise data points (i.e., individual values)

Best Practices for a Sparkline Visualization

If you use a sparkline, here are the key design best practices:

- To assist with readability, consider adding indicators on the side that give a better glimpse into the data, like in the example above

- Stick to one colour for your sparklines to keep them consistent on your dashboard

8.4.5. Pie Chart

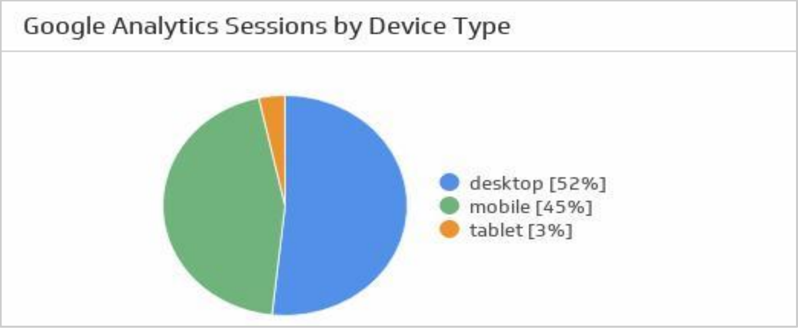


Figure 8.7. Pie chart.

Here is the pie chart showing the sales ratio of electronic gadgets like Mobile, desktop, tablet. Pie charts are an interesting graph visualization. At a high-level, they're easy to read and understand because the parts-of-a-whole relationship is made very obvious. But top data visual experts agree that one of their disadvantages is that the percentage of each section isn't obvious without adding numerical values to each slice of the pie.

When Do I Use a Pie Chart Visualization?

Use a pie chart for the following reasons:

- You want to compare relative values
- You want to compare parts of a whole
- You want to rapidly scan metrics

Don't use a pie chart for the following reason:

- You want to precisely compare data

Best Practices for a Pie Chart Visualization

If you use a pie chart, here are the key design best practices:

- Make sure that the pie slices add up to 100%. To make this easier, add the numerical values and percentages to your pie chart
- Order the pieces of your pie according to size
- Use a pie chart if you have only up to 5 categories to compare. If you have too many categories, you won't be able to differentiate between the slices

8.4.6. Gauge

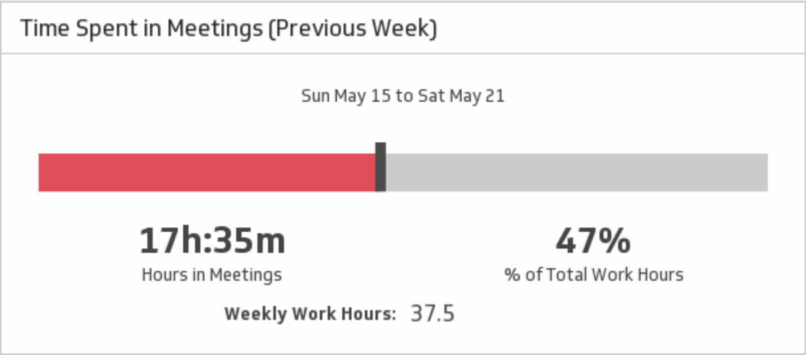


Figure 8.8. Gauge.

Gauges typically only compare two values on a scale: they compare a current value and a target value, which often indicates whether your progress is either good or bad, in the green or in the red.

When Do I Use a Gauge Visualization?

Use a gauge for the following reason:

- You want to track single metrics that have a clear, in the moment objective

Don’t use a gauge for the following reasons:

- You want to track multiple metrics
- You’re looking to visualize precise data points

Best Practices for a Gauge Visualization

If you use a gauge, here are the key design best practices:

- Feel free to play around with the size and shape of the gauge. Whether it’s an arc, a circle or a line, it’ll get the same job done
- Keep the colours consistent with what means “good” or “bad” for you and your numbers
- Use consistent colours and labeling throughout so that you can identify relationships more easily
- Simplify the length of the y-axis labels and don’t forget to start from 0 so you can keep your data in order

8.4.7. Waterfall Chart

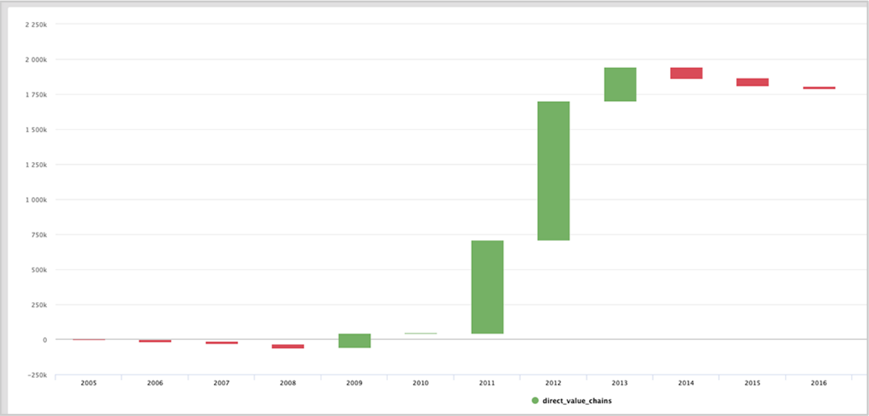


Figure 8.9. Waterfall chart.

Here is the waterfall chart plotting between rate v/s timeline in year. A waterfall chart is an information visualization that should be used to show how an initial value is affected by intermediate values and results in a final value. The values can be either negative or positive.

When Do I Use a Waterfall Chart Visualization?

Use a waterfall chart for the following reason:

- To reveal the composition or makeup of a number

Don't use a waterfall chart for the following reason:

- You want to focus on more than one number or metric

Best Practices for a Waterfall Chart Visualization

If you use a waterfall chart, here are the key design best practices:

- Use contrasting colors to highlight differences in data sets
- Choose warm colors to indicate increases and cool colors to indicate decreases

8.4.8. Funnel Chart

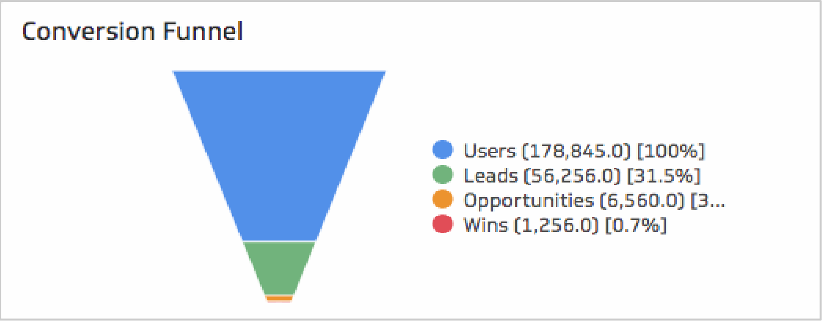


Figure 8.10. Funnel chart.

A funnel chart is your data visualization of choice if you want to display a series of steps and the completion rate for each step. This can be used to track the sales process, a marketing funnel or the conversion rate across a series of pages or steps. Funnel charts are most often used to represent how something moves through different stages in a process. A funnel chart displays values as progressively decreasing proportions amounting to 100 percent in total.

When Do I Use a Funnel Chart Visualization?

Use a funnel chart for the following reason:

- To display a series of steps and each step's completion rate

Don't use a funnel chart for the following reason:

- To visualize individual, unconnected metrics

Best Practices for a Funnel Chart Visualization

If you use a funnel chart, here are the key design best practices:

- Scale the size of each section to accurately reflect the size of its data set
- Use contrasting colors or one color in gradating hues, from darkest to lightest as the size of the funnel decreases

8.4.9. Heat Map

A heat map or choropleth map is a data visualization that shows the relationship between two measures and provides rating information. The rating information is displayed using varying colors or saturation and can exhibit ratings such as high to low or bad to awesome, and needs improvement to work well.

It can also be a thematic map in which the area inside recognized boundaries is shaded in proportion to the data being represented.

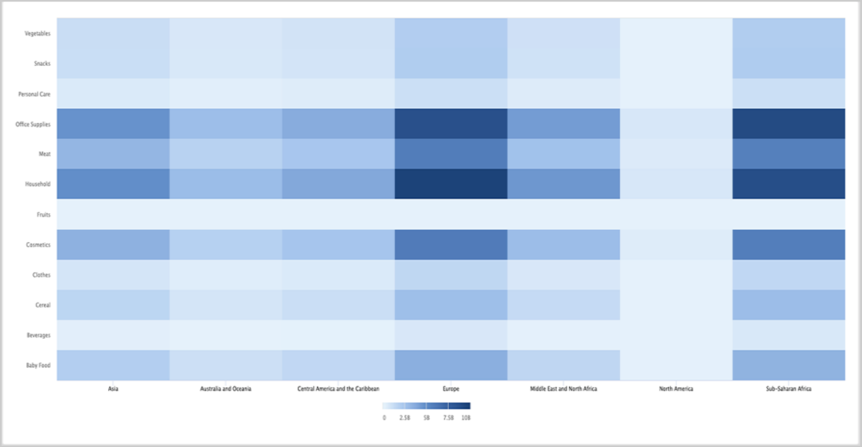


Figure 8.11. Heatmap.

When Do I Use a Heat Map Visualization?

Use a heat map for the following reasons:

- To show a relationship between two measures
- To illustrate an important detail
- To use a rating system

Don't use a heat map for the following reason:

- To visualize individual, unconnected metrics

Best Practices for a Heat Map Visualization

If you use a heat map, here are the key design best practices:

- Use a simple map outline to avoid distracting from the data
- Use a single color in varying shades to show changes in data
- Avoid using multiple patterns

8.4.10. Histogram

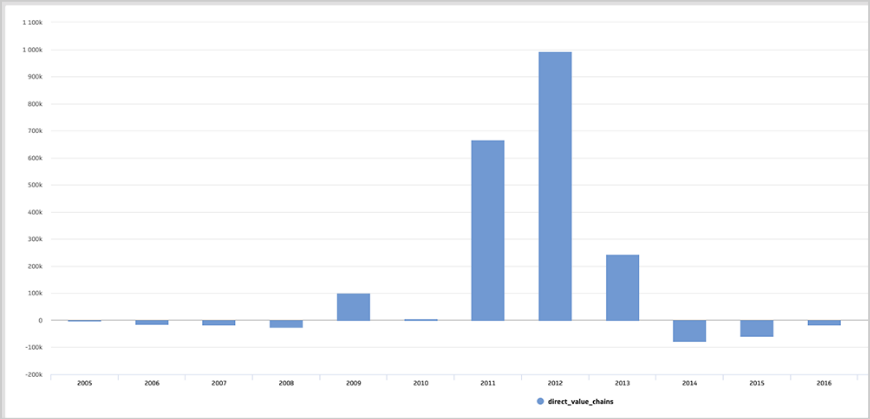


Figure 8.12. Histogram.

A histogram is a data visualization that shows the distribution of data over a continuous interval or certain time period. It's basically a combination of a vertical bar chart and a line chart. The continuous variable shown on the X-axis is broken into discrete intervals and the number of data you have in that discrete interval determines the height of the bar.

Histograms give an estimate as to where values are concentrated, what the extremes are and whether there are any gaps or unusual values throughout your data set.

When Do I Use a Histogram Visualization?

Use a histogram for the following reason:

- To make comparisons in data sets over an interval or time
- To show a distribution of data

Don't use a histogram for the following reason:

- To compare 3+ variables in data sets

Best Practices for a Histogram Visualization

If you use a histogram, here are the key design best practices:

- Avoid bars that are too wide that can hide important details or too narrow that can cause a lot of noise
- Use equal round numbers to create bar sizes
- Use consistent colours and labeling throughout so that you can identify relationships more easily

8.4.11. Box Plot

A box plot, or box and whisker diagram, is a visual representation of displaying a distribution of data, usually across groups, based on a five-number summary: the minimum, first quartile, the median (second quartile), third quartile, and the maximum.

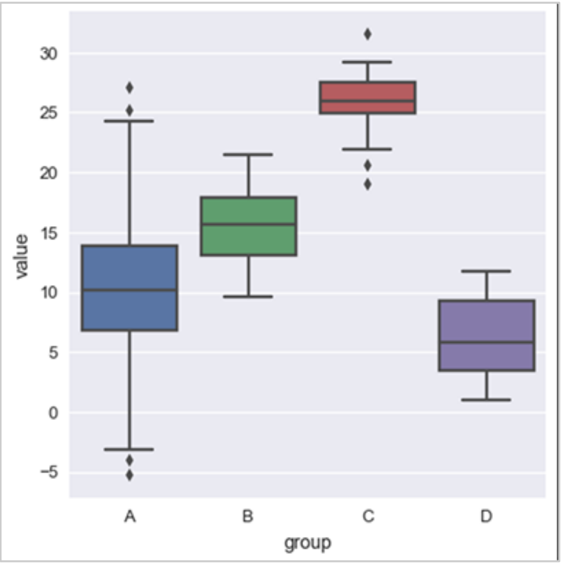


Figure 8.13. Box plot.

The simplest of box plots display the full range of variation from minimum to maximum, the likely range of variation, and a typical value. A box plot will also show the outliers.

When Do I Use a Box Plot Visualization?

Use a box plot for the following reasons:

- To display or compare a distribution of data
- To identify the minimum, maximum and median of data

Don't use a box plot for the following reason:

- To visualize individual, unconnected data sets

Best Practices for a Box Plot Visualization

If you use a box plot, here are the key design best practices:

- Ensure font sizes for labels and legends are big enough and line widths are thick enough to understand the findings easily
- If plotting multiple datasets, use different symbols, line styles or colour to differentiate each.
- Always remove unnecessary clutter from the plots

8.4.12. Maps

Maps are an amazing visualization to add to your dashboard if organizing data geographically tells an important story for your business. For example, if your dashboard is looking at monthly sales, it could be extremely useful to see the geographic locations of your customers.

Above, you'll find a map visualization that integrates with Salesforce to measure accounts by country. Keep in mind that if your dashboard is looking at daily sales, this visualization may provide less value to your day-to-day discussions.

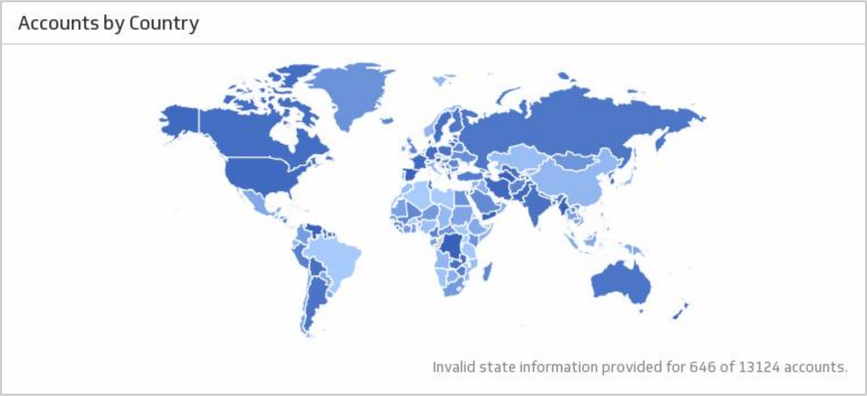

Accounts by Country
Invalid state information provided for 646 of 13124 accounts.

Figure 8.14. Maps.

When Do I Use a Map Visualization?

Use a map for the following reason:

- Geography is an important part of your data story

Don't use a map for the following reasons:

- You want to show precise data points
- Geography is not an important element of the dashboard's overarching story

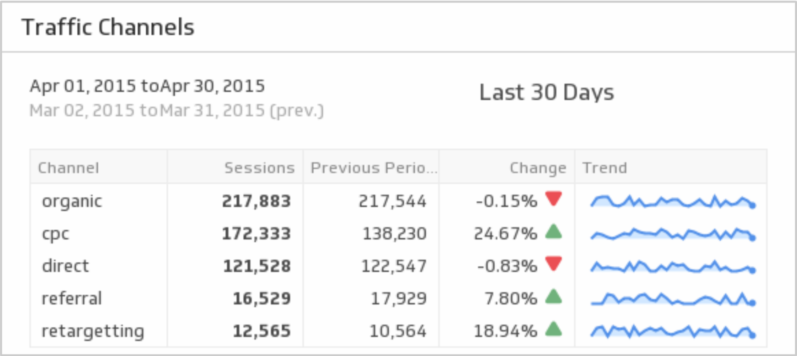
Best Practices for a Map Visualization

If you use a map visualization, here are the key design best practices:

- Avoid using multiple colours and patterns on your map. Use varying shades of the same colour instead
- Make sure to include a legend with your map, so that everyone understands what the data means

8.4.13. Tables

If you’re someone who wants a little bit of everything in front of you in order to make thorough decisions, then tables are the visualization to go with. Tables are great because you can display both data points and graphics, such as bullet charts, icons, and sparklines. This visualization type also organizes your data into columns and rows, which is great for reporting.



Traffic Channels

Apr 01, 2015 toApr 30, 2015
Mar 02, 2015 toMar 31, 2015 (prev.)

Last 30 Days

Channel	Sessions	Previous Perio...	Change	Trend
organic	**217,883**	217,544	-0.15% ▼	
cpc	**172,333**	138,230	24.67% ▲	
direct	**121,528**	122,547	-0.83% ▼	
referral	**16,529**	17,929	7.80% ▲	
retargetting	**12,565**	10,564	18.94% ▲	

Figure 8.15. Tables.

Above is an example of how to bring in your Google Analytics data into a table, so that you can see all the information you need in one place.

One thing to keep in mind is that tables can sometimes be overwhelming if you have a dashboard with many metrics that you want to display. It’s important to find a happy medium between large amounts of data (confusing) and too little data (waste of dashboard space).

When Do I Use a Table Visualization?

Use a table for the following reasons:

- You want to display two-dimensional data sets that can be organized categorically
- You can drill-down to break up large data sets with a natural drill-down path

Don't use a table for the following reason:

- You want to display large amounts of data

Best Practices for a Table Visualization

If you use a table, here are the key design best practices:

- Be mindful of the order of the data. Make sure that labels, categories and numbers come first then move on to the graphics
- Try not to have more than 10 different rows in your table to avoid clutter

8.4.14. Indicators

Indicators are useful for an at a glance view of a metric you need to keep track of. An indicator is simply a number showing the current value of whichever performance metric you're tracking. To make it more useful, add a comparison to the previous time period to show whether your metric is tracking up or down.

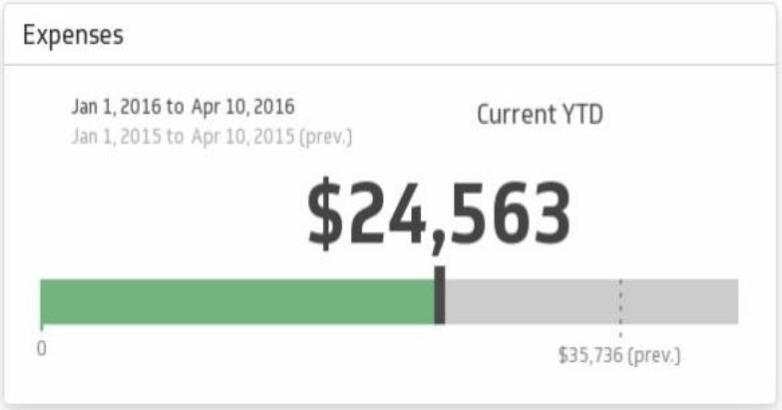
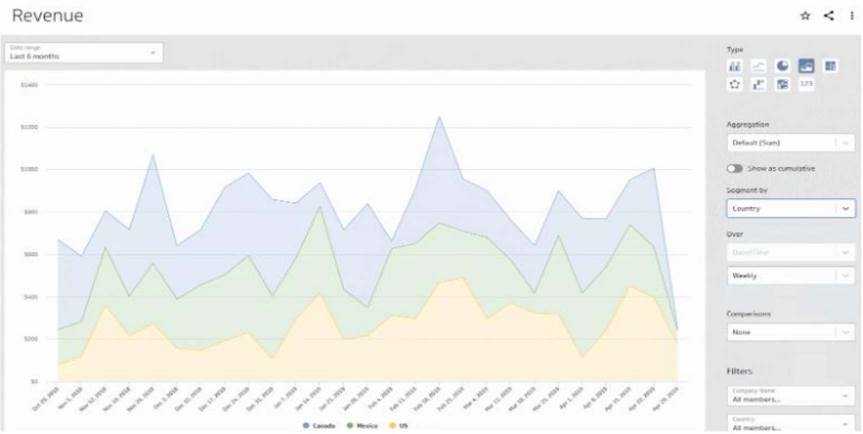

Expenses
Jan 1, 2016 to Apr 10, 2016
Jan 1, 2015 to Apr 10, 2015 (prev.)
Current YTD
$24,563
0
$35,736 (prev.)

Figure 8.16. Indicators.

Some people like to get fancy with indicators and use gauges or tickers. They present the same type of information, just in a different visual way.

8.4.15. Area Chart

An area chart is very similar to a line graph but may do a better job at highlighting the relative differences between items. Use an area chart when you want to see how different items stack up or contribute to the whole.



Revenue

Date range: Last 6 months

Aggregation: Default (Sum)

Show as cumulative

Segment by: Country

Over: Date/Time

Weekly

Comparisons: None

Filters

Company name: All members...

Country: All members...

Figure 8.17. Area chart.

8.5. Tools

8.5.1. Tableau

What Is Tableau?

“Tabulae is a white-label data visualization and analytics platform, especially designed to smoothly embed dashboards and reports into third-party software.” Tabulae is a data visualization and analysis platform.

Tableau is the fastest growing data visualization and data analytics tool that aims to help people see and understand data. In order to transform the way people, use data to solve problems, tableau software ensures to meet strict requirements. In other words, it simply converts raw data into a very easily understandable format.

Data analysis is great, as it is a powerful visualization tool in the business intelligence industry. Data that is created using this software becomes so easy that it allows even a non-technical user to create a customized dashboard. It provides top class interactive data visualization with the purpose to help organizations solve their data problems.

For more detailed information visit to official site: https://www.tableau.com/

Features of Tableau

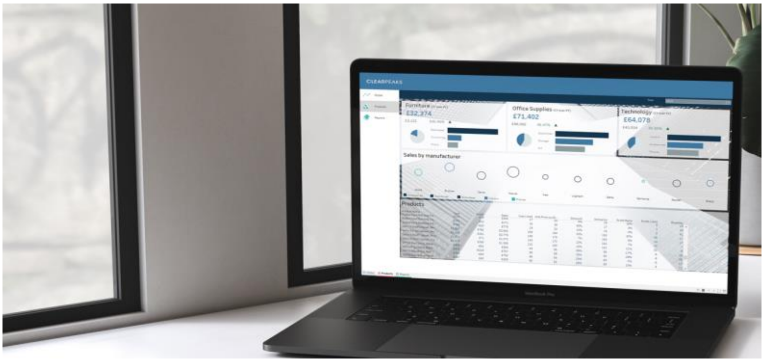


Figure 8.18. Tableau.

- Translate queries to visualizations
- Share dashboards
- Highlight and filter data
- Toggle view and drag-and-drop
- Data Notifications
- Create interactive dashboards
- Tableau Public for data sharing
- Dashboard commenting

- Tableau Reader for data viewing
- Import all ranges and sizes of data
- Security permissions at any level
- Automatic updates

Company Uses Tableau

The companies mentioned here are just like a few waters drops of the ocean of Tableau users. There are a plethora of companies including Amazon, Adobe, Ferrari, LinkedIn, Nike, Coca-Cola, Skype, The World Bank, Wells Fargo, Citigroup, Amica, The New York Times, etc. that use Tableau heavily and effectively.

Advantages of Tableau

- Fantastic Visualizations: You can now work with a lot of data that doesn’t have any order to it and create a range of visualizations. Well, thanks to the in-built features of Tableau which help you create visualizations that surely stand out of the crowd. You also have the option of switching between different visualizations to bring about a greater context, ways of drilling down data, and exploring the data at a minute level.
- In-depth Insights: Tableau can help enterprises futuristically to analyze data without any specific goals in mind. You can explore visualizations and have a look at the same data from different angles. You can frame ‘what if’ queries and work with data by hypothetically visualizing it in a different manner and dynamically adding components for comparison and analysis. When you are working with real-time data, then these capabilities are highlighted in a huge manner.
- User-friendly Approach: This is the greatest strength of Tableau. It is built from the ground level for people who don’t have any technical skills or coding experience. So, everything can be done with this tool by anybody without any prior set of skills. Since

most of the features are in a drag-and-drop format, each visualization is so intuitive and self-depicting.

8.5.2. Google Spreadsheet

What Is Google Spreadsheet?

A spreadsheet is a computer application for organization, analysis and storage of data in tabular form. Spreadsheets were developed as computerized analogs of paper accounting worksheets. The program operates on data entered in cells of a table.

Google Sheets is a spreadsheet program included as part of the free, web-based Google Docs office suite offered by Google within its Google Drive service. The service also includes Google Docs and Google Slides, a word processor and presentation program respectively. Google Sheets is available as a web application, mobile app for Android, iOS, Windows, BlackBerry, and as a desktop application on Google's ChromeOS. The app is compatible with Microsoft Excel file formats. The app allows users to create and edit files online while collaborating with other users in real-time. Edits are tracked by users with a revision history presenting changes. An editor's position is highlighted with an editor-specific color and cursor and a permissions system regulates what users can do. Updates have introduced features using machine learning, including "Explore", offering answers based on natural language questions in a spreadsheet.

For more detailed information visit to official site:-https://www.google.com/sheets/about/

Features of Google Spreadsheet

- Exploration
- Supported file format
- Offline editing
- Editing
- Integration with other google product

Figure 8.19. Spreadsheet.

Advantages of Google spreadsheet

- Collaboration: The most immediate benefit from using Sheets is in the ability to collaborate in completely new ways. The “old style” of working would be using a master file that someone has to “own,” which is then (in the best case) kept on a shared network folder, or painstakingly emailed around.

- Working at scale: One of the misconceptions my colleagues and I had was the notion that Sheets is fine for small calculations—more or less like an advanced calculator—but not useful for larger models or datasets. Turns out we were wrong. I've used it for a number of larger operating and valuation models over the past years and am very impressed with the performance.
- Creating charts and linking to google slides: Raw spreadsheet work ("wrangling") is an important part of the finance professional's daily life, but even the best analysis is of limited value if you are unable to communicate your findings in a cogent and compelling way. This brings us to two other staples of the finance toolbox: charts and presentations.
- Version control: If you have ever had the painful experience of a spreadsheet crash beyond recovery, resulting in hours of lost work, you might have a developed habit of saving new files frequently. This can end up in numerous iterations of files with tedious version updates (v3.4.0, v3.4.1, etc.)

Company Uses Google Spreadsheets

There are many companies who use google spreadsheets, some of them are, Doordash, Whirlpool Corporation, Ascension health, Carvana, Stitch fix and many more.

8.5.3. Excel

What Is Excel?

A software program created by Microsoft that uses spreadsheets to organize numbers and data with formulas and functions. Excel analysis is ubiquitous around the world and used by businesses of all sizes to perform financial analysis.

For more detailed information visit to official site:- www.microsoft.com/en/microsoft-365/excel

Feature of Excel

- Conditional Formality
- Add multiple rows
- Paste special
- Pivot Table
- Absolute references
- Print Optimization
- Filter
- Index match
- Flash-fill

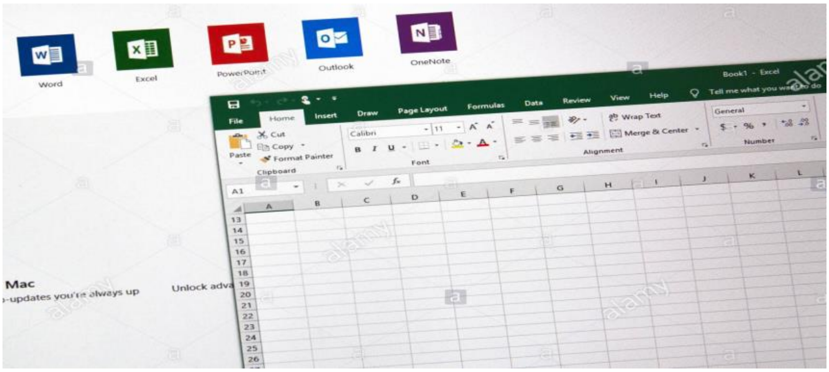


Figure 8.20. Excel.

Company Uses Excel

There are many companies who use Excel, some of them are: .io DevOps, Akelius Infrastructure, Outsystems, Wanderlust AI, Direct Market, Trounceflow.

Advantages of Excel

- *Easy data entry and operations:* One of the main advantages of MS excel is that it facilitates smooth and easy data entry. Compared to any other data entry and analyzing tools, MS Excel offers features like Ribbon interface, a set of commands used to perform certain operations. Ribbon consists of many tabs, which again consist of many command groups and their buttons. You can select the commands by clicking the related button and perform operations very easily.

- *Accurate comparisons and analysis options:* MS Excel provides many analytical tools for the accurate analysis and comparison of large amounts of data. The advanced sorting and filtering techniques allows you to sort out large amounts of data so that it will be easier for you to find out the required information. Also, filtering removes unwanted or repeated data and helps to save time and effort.
- *Allows graphical representation of data:* MS Excel allows you to create the visual representation of data and information. The data can be visually displayed in the form of bar charts, column charts and graphs. It automatically revises the charts and graphs, once the data gets modified. Tables help to classify different entities according to their characteristics and features.
- *Compatible with other business applications:* Since the recent versions of MS Excel is compatible with many other business applications like MS office, other web applications etc., it allows you to import excel data to other applications. Also, the cloud computing facility helps to update and upload your excel document from all locations, which can be accessed later through various devices like smartphones, tablets, laptops etc.
- *Ready to use formulas:* MS Excel performs all mathematical and logical functions like addition, subtraction, multiplication, division, average, sum, mod, product etc. Excel provides many formulas that help you to solve both simple and complex calculations.

LINKS AND REFERENCES USE IN THIS CHAPTER

Links

1. https://www.tableau.com/
2. https://www.google.com/sheets/about/
3. www.microsoft.com/en/microsoft-365/excel

References

Bateman, S., Mandryk, R., Gutwin, C., Genest, A., McDine, D. and Brooks, C. 2010. “Useful Junk? The Effects of Visual Embellishment on Comprehension and Memorability of Charts.” *ACM Conference on Human Factors in Computing Systems,* 2573–82. doi:10.1145/1753326.1753716.

Becker, R. A., Cleveland, W. S., and Shyu, M.-J. 1996. “The Visual Design and Control of Trellis Display.” *Journal of Computational and Graphical Statistics* 5: 123–55.

Bergstrom, C. T., and West, J. 2016. “*The Principle of Proportional Ink.*” http://callingbullshit.org/tools/tools_proportional_ink.html.

Borgo, R., Abdul-Rahman, A., Mohamed, F., Grant, P. W., Reppa, I. and Floridi, L. 2012. “*An Empirical Study on Using Visual Embellishments in Visualization.*” IEEE Transactions on Visualization and Computer Graphics 18: 2759–68. doi:10.1109/TVCG.2012.197.

Brewer, Cynthia A. 2017. *ColorBrewer 2.0. Color Advice for Cartography*. http://www.ColorBrewer.org.

Carr, D. B., Littlefield, R. J., Nicholson, W. L. and Littlefield, J. S. 1987. “Scatterplot Matrix Techniques for Large N.” *Journal of the American Statistical Association* 82: 424–36.

Clauset, A., Shalizi, C. R., and Newman, M. E. J. 2009. “Power-Law Distributions in Empirical Data.” *SIAM Review* 51: 661–703. Cleveland, R. B., W. S.

Cleveland, J., McRae, E. and Terpenning, I. 1990. “STL: A Seasonal-Trend Decomposition Procedure Based on Loess.” *Journal of Official Statistics* 6: 3–73.

Cleveland, W. S. 1979. “Robust Locally Weighted Regression and Smoothing Scatter- plots.” *Journal of the American Statistical Association* 74: 829–36.

ABOUT THE AUTHORS

Dr. Gopal Sakarkar is currently working at Department of Artificial Intelligence and Machine Learning at G H Raisoni College of Engineering, Nagpur (Autonomous Institute). He has completed his Ph.D. in the faculty of Science and Technology (Comp. Science & Engineering) in 2017 from S.G. B. Amravati University, Amravati, Maharashtra. He had filed 01 Patent and also his 03 Copyrights are granted by Gov. of India. He is ISTE life Member and an IEEE - CIS Member. His research area includes AI, Machine Learning, Data Pre-processing techniques, intelligent e-learning, Image Processing etc.

He was organized 3 National conference under TEQIP I/II, two International AI Summits successfully under his leadership. He has good research collaboration with international researchers including Australia, Germany, Japan, United Kingdom, Sweden, Venezuela and Czech Republic. Currently he is working with two industry-based projects in his

Centre of Excellence in AI and Machine Learning at G H Raisoni College of Engineering.

As research is his passion, till date he has 30 plus publications in his bag that includes 13 international journal publications, including 5 Scopus and 5 Web of Science journals, 12 international conferences paper's publication in reputed publication like Springer, IEEE, Elsevier etc. and 5 National level papers publication. He is the convener and organizer for various International and National level Conferences, STTP, Summits and Workshops. He is a author of a book on Machine Learning and his one book chapter was published by the Springer Nature Journal publication. Till date he has 175+ citations, 6 h-index and 6 i10-index for his research publications.

He has received a 'Best Teacher Award -2019' for contributing to develop a Centre of Excellence in AI and Machine Learning and taking tired less effort to start the Department of AI at GHRCE, Nagpur.

E-mail: g.sakarkar@gmail.com.

Mr. Gaurav Patil is a Tech-aspirant candidate who is passionate to work for the development of technology in modern society. Gaurav Patil is an Artificial Intelligence and Data Science Enthusiast. He loves to work in the field of Research & Development. He has experience with working and researching on project based on Artificial Intelligence, Machine Learning, Deep Learning, Computer Vision, Natural Language Processing. He has developed AI based solution in field of Cybersecurity, Medical Science, etc and have presented research paper in International Conference. He is an IEEE member. He has complete various certification course on ML and

DL and has a good hold on Python language. His Passion and Hobbies include Playing Cricket Game, Football, Running, Coding and Reading Books.

He used to get familiar with every corresponding working environment and corporate with his colleague in a deterministic manner. I grateful to work with Dr. Gopal Sakarkar & my colleague Mr. Prateek Dutta & thankful them for supporting me and worked together to complete the book. We have achieved a measurable success in past few years by working together in several projects.

E-mail: gauravpatil22301@gmail.com.

Mr. Prateek Dutta has a technical expertized in the field of Artificial Intelligence, Machine Learning, Deep Learning, Data science. He is having a background in research and have presented research paper in International conference. Currently he is working in several AI based project and having free hand in python. He is technology aspirant candidate who is deterministic & passionate about working with technologies and explore more about it. He has worked with several domain like Healthcare & cybersecurity in order to provide AI based solution. He has completed various AI based certifications. He is an IEEE & IFERP member & provide part time support to Coding Blocks society. His passion & hobbies include Playing Cricket, listening songs, reading research articles & many more. I have the fertile fortunate of acquaintances with a great scholar who is none other than Dr. Gopal Sakarkar and my other Co-author Mr. Gaurav Patil, to whom I deeply beholden for their inspiration and hard work they put to complete this book.

E-mail: prateekdutta2001@gmail.com.

INDEX

Gopal Sakarkar • Gaurav Patil • Prateek Dutta

Machine Learning Algorithms Using Python Programming

nova
science publishers

www.novapublishers.com

ISBN 978-1-53619-512-5

9 781536 195125