**An**

**Industrial Training Report**

**on**

**“Prediction of Heart Attack”**

**For The Award of The Degree of Bachelor of Technology**

**in**

**COMPUTER SCIENCE ENGINEERING**

at



DEPARTMENT OF CSE

**WORLD COLLEGE OF TECHNOLOGY AND MANAGEMENT,**

**FARRUKH NAGAR, GURGAON**

SUBMITTED BY

**Mohan Dev Prasad**

ROLL NO : **40/CSE/16**

Training Co-ordinator Head of Department

**Ms Shweta Mallick Mr. Subhash Chandra**

Assistant Professor Professor & Head

Deptt. Of CSE/IT/MCA Deptt.CSE/IT/MCA

WCTM Gurgaon WCTM Gurgaon

# CERTIFICATE OF EXAMINATION

This is to certify that we have examined the Training report on **“Prediction of Heart attack”** submitted by **MOHAN DEV PRASAD** students of final year B.Tech (Computer Science Engineering/Information Technology)/Masters in computer Application.).We hereby accord our approval of it as an Industrial Training carried out and presented in a manner required for partial fulfillment for the Bachelor of Technology (Computer Science Engineering) degree of Maharshi Dayanand University (Rohtak). This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn as recorded in the Training report; it only signifies the acceptance of the Training report for the purpose for which it is submitted.

Examiner (Internal) HOD (CSE/IT/MCA)

Ms. Shweta Mallick Mr.Subhash Chandra

Date Date

# CERTIFICATE OF COMPLETION

# ACKNOWLEDGEMENT

I take this opportunity to express my sincere thanks and deep gratitude to all those people who extended their whole hearted co-operation and have helped me in completing this project successfully. I am highly indebted to **Mr. Subhash Chandra** for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in completing the project. Her inspiring suggestions and timely guidance enabled me to perceive the various aspects of the project in a new light.

I would like to express my gratitude towards member of **NATIONAL INFOMATIC CENTER** for their kind co-operation and encouragement which help me in completion of this project. I would like to express my special gratitude and thanks to **Ms. Shweta Mallick** for giving me such attention and time. In all I found congenial work environment and this completion of the project will mark a new beginning for me in the coming days.

**MOHAN DEV PRSAD**

# RollNo:40/CSE/16

# DECLARATION

I, **Mohan Dev Prasad ,** hereby declare that the work presented in the project report entitled “**Prediction of Heart attack”** submitted to the Department of Computer Science, World College of Technology and Management, Gurgaon, for the partial fulfillment of the requirement for the award of Degree of “**Bachelor of Technology in Computer Science**” is our true record of work carried during the period from **febuary 2020 to April 2020,** under the guidance of **Ms. Shweta Mallick** The matter embodied in this project has not been submitted by anybody for the award of any other degree.

Mohan Dev Prasad

(40/CSE/16)

# ABSTRACT

MACHINE LEARNING PROJECT 2

Abstract

In this project, we were asked to experiment with a real world dataset, and to explore how

machine learning algorithms can be used to find the patterns in data. We were expected to gain

experience using a common data-mining and machine learning library, Weka, and were expected

to submit a report about the dataset and the algorithms used. After performing the required tasks

on a dataset of my choice, herein lies my final report.

MACHINE LEARNING PROJECT 2

Abstract

In this project, we were asked to experiment with a real world dataset, and to explore how

machine learning algorithms can be used to find the patterns in data. We were expected to gain

experience using a common data-mining and machine learning library, Weka, and were expected

to submit a report about the dataset and the algorithms used. After performing the required tasks

on a dataset of my choice, herein lies my final report.

In this project, we were asked to experiment with a real world dataset, and to explore how machine learning algorithms can be used to find the patterns in data. We were expected to gain experience using a common data-mining and machine learning library and were expected to submit a report about the dataset and the algorithms used. After performing the required tasks on a dataset of my choice, herein lies my final report. This project basically predict a salary on the basis of Interview Test Score. So, that’s why I developed a Web app with the help of flask framework Which predict the salary of an applicant on the basis of Interview Test Score .This are the some steps which is used for completion the Machine Learning Project:-

* Abstract the data from a CSV File
* Data Cleaning
* Data representation and visualization
* Train our Model
* Prediction on given data

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords: Machine Learning, Pattern Recognition, Classification, Supervised learning,

Artificial Intelligence

Keywords:-Machine Learning, Supervised Learning, Linear Regression, Artificial Intelligence

**TABLE OF CONTENTS**

Certificate of Examination i

Certificate of Completion ii

Acknowledgement iii

Declaration iv

Abstract v

Table of Contents vi

**Chapter 1 About Weblink.in Pvt Ltd 09**

1. History 09
2. Business Ventures 10

**Chapter 2 About Machine Learning 11**

2.1 What is Machine Learning? 11 2.2 Types of Machine Learning 12

2.3 Applications of Machine Learning 14

**Chapter 3 Significance of Training 16**

3.1 Technologies Studied 16

3.1.1 Python 16

3.1.2 Data analyzing 17

3.1.3 Python Libraries 17

3.1.4 Prediction using previous data 19

3.1.5 Spyder I.D.E 20

**Chapter 4 Objective of the Project 21**

4.1 Methodology 21

4.2 Data Collection and Data Required 33

**Chapter 5 Overview of the Project 39**

5.1 Mentioning all the libraries 39

5.2 Data abstraction from the comma separated values 39

5.3 Finding the null values in data 40

5.4 Data visualization 40

5.5 Training the model using Data 45

5.6 Predict the outcomes 46

5.7 Factors that affect the prediction most 46

**Chapter 6 Conclusion 48**

**References 49**

CHAPTER 1

# About National Informatic Center

The **Ministry of Electronics and Information Technology** (**MeitY**) is an executive agency of the [Union Government](https://en.wikipedia.org/wiki/Government_of_India) of the [Republic of India](https://en.wikipedia.org/wiki/India). It was carved out of the [Ministry of Communications and Information Technology](https://en.wikipedia.org/wiki/Ministry_of_Communications_and_Information_Technology_(India)) on 19 July 2016 as a standalone ministerial agency responsible for IT policy, strategy and development of the electronics industry.

The NIC was established in 1976 under the aegis of the [Ministry of Electronics and Information Technology](https://en.wikipedia.org/wiki/Ministry_of_Electronics_and_Information_Technology).

The NIC is credited with helping the Indian government embrace IT in the 1990s and has also helped disseminate e-governance to the masses.

It had an annual budget of [₹](https://en.wikipedia.org/wiki/Indian_rupee)11.5 billion (US$160 million) for the year 2018–19.

In May 2019, the government of India set up the Centre for Smart Governance (CSG), and state governments have since been advised to consult the CSG for IT projects they previously would have consulted the NIC and private firms for. Some claim that government sources have said "NIC is said to be unable to scale up", and Rajeev Chawla, Additional Chief Secretary (e-Governance), was quoted as saying "CSG will be an analogue to NIC".

* 1. Infrastructure

### Network

The National Informatics Centre (NIC) was established in 1976 under the Planning Commission by the India Government. Then Additional Secretary late Dr. N Shesagiri was the first to introduce a network system in India called NICNET. In 1990, the takeoff by the Ministry of Electronics and Information Technology NIC's ICT Network, "NICNET", facilitates the institutional linkages with the Ministries/Departments of the Central Government, state Governments and District administrations of India. NIC is noted for being the primary constructor of e-Government applications.

### Data centres and offices

In 2018, NIC opened its fourth data centre in Bhubaneshwar to complement its existing data centres in New Delhi, Hyderabad and Pune. In addition to the national data centres there are NIC State Centres in 36 [states or Union territories](https://en.wikipedia.org/wiki/Federalism_in_India#Territories). This is supplemented by 708 district offices.

### National Portal of India

NIC maintains the [National Portal of India](https://en.wikipedia.org/wiki/National_Portal_of_India). The portal contains the [Constitution of India](https://en.wikipedia.org/wiki/Constitution_of_India),[[16]](https://en.wikipedia.org/wiki/National_Informatics_Centre#cite_note-TRLAW-portal-16) and claims to have a design objective to a single point to access the information and services of the Government of India.

# Chapter 2

**2.1 About MLP**

**Machine learning** (**ML**) is the study of computer algorithms that improve automatically through experience.[[1]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-1) It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a [mathematical model](https://en.wikipedia.org/wiki/Mathematical_model) based on sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to do so.[[2]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-2)[[3]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-bishop2006-3):2 Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering) and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks.

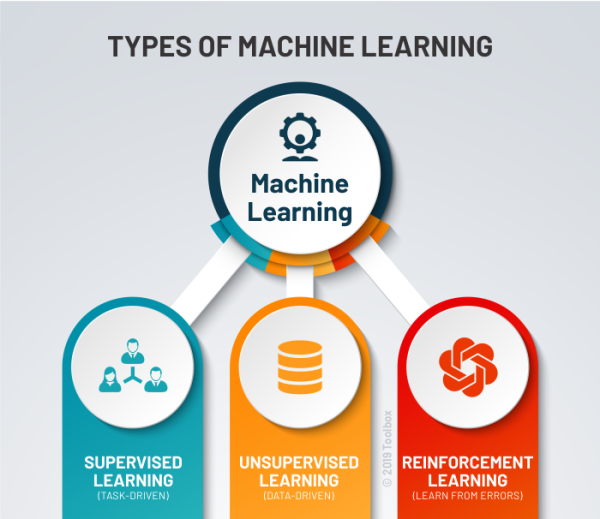
Machine learning is closely related to [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning. [Data mining](https://en.wikipedia.org/wiki/Data_mining) is a related field of study, focusing on [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) through [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning).[[4]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-4)[[5]](https://en.wikipedia.org/wiki/Machine_learning#cite_note-5) In its application across business problems, machine learning is also referred to as [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics)

## What *type* of thing is machine learning?

* An [academic discipline](https://en.wikipedia.org/wiki/Academic_discipline)
* A branch of [science](https://en.wikipedia.org/wiki/Science)
  + An [applied science](https://en.wikipedia.org/wiki/Applied_science)
    - A subfield of [computer science](https://en.wikipedia.org/wiki/Computer_science)
      * A branch of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence)
      * A subfield of [soft computing](https://en.wikipedia.org/w/index.php?title=Soft_computing&action=edit&redlink=1)



## 2.2 Types of Machine Learning

As with any method, there are different ways to train machine learning algorithms, each with their own advantages and disadvantages. To understand the pros and cons of each type of machine learning, we must first look at what kind of data they ingest. In ML, there are two kinds of data — labeled data and unlabeled data.

Labeled data has both the input and output parameters in a completely machine-readable pattern, but requires a lot of human labor to label the data, to begin with. Unlabeled data only has one or none of the parameters in a machine-readable form. This negates the need for human labor but requires more complex solutions.

There are also some types of machine learning algorithms that are used in very specific use-cases, but three main methods are used today.

### 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 parameters given, essentially establishing a cause and effect relationship between the variables in the dataset. At the end of the training, the [algorithm](https://it.toolbox.com/blogs/shrutiumathe/mphasis-unveils-its-deep-learning-algorithms-on-amazon-web-services-marketplace-for-machine-learning-073119) 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 it learns from 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.

### 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.

In supervised learning, the labels allow the algorithm to find the exact nature of the relationship between any two data points. However, unsupervised learning does not have labels to work off of, 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.

## 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.

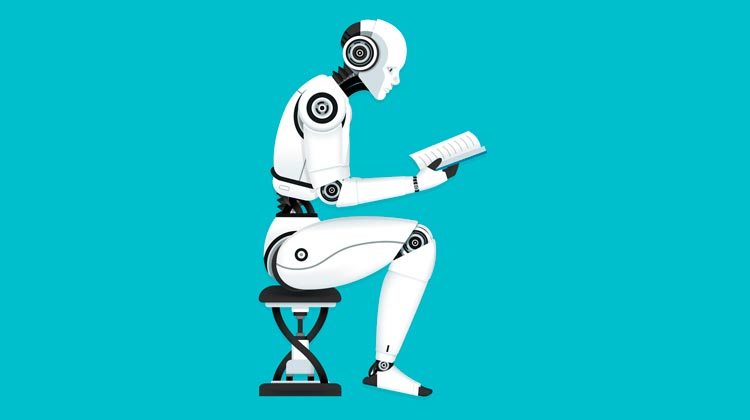
## Applications of Machine Learning

Machine learning algorithms are used in circumstances where the solution is required to continue improving post-deployment. The dynamic nature of adaptable machine learning solutions is one of the main selling points for its adoption by companies and organizations across verticals.

Machine learning algorithms and solutions are versatile and can be used as a substitute for medium-skilled human labor given the right circumstances. For example, customer service executives in large B2C companies have now been replaced by natural language processing machine learning algorithms known as chatbots. These chatbots can analyze customer queries and provide support for human customer support executives or deal with the customers directly.

Machine learning algorithms also help to improve user experience and customization for online platforms. Facebook, Netflix, Google, and Amazon all use recommendation systems to prevent content glut and provide unique content to individual users based on their likes and dislikes.

Facebook utilizes recommendation engines for its news feed on both Facebook and Instagram, as well as for its advertising services to find relevant leads. Netflix collects user data and recommends various movies and series based on the preferences of the user. Google utilizes machine learning to structure its results and for YouTube’s recommendation system, among many other applications. Amazon uses ML to place relevant products in the user’s field of view, maximizing conversion rates by recommending products that the user actually wants to buy.

However, as ML continues to be applied in various fields and use-cases, it becomes more important to know the difference between artificial intelligence and machine learning.

# Chapter 3

3.1 Technologies Studied

3.1.1 Python

 **Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

Python was conceived in the late 1980s as a successor to the [ABC language](https://en.wikipedia.org/wiki/ABC_(programming_language)).

Python 2.0, released in 2000, introduced features like [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a garbage collection system capable of collecting [reference cycles](https://en.wikipedia.org/wiki/Reference_cycle). Python 3.0, released in 2008, was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility), and much Python 2 code does not run unmodified on Python 3.

The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release."[[30]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-30) No more security patches or other improvements will be released for  With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)), only Python 3.5.x and later are supported.

Python [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). A global community of programmers develops and maintains [CPython](https://en.wikipedia.org/wiki/CPython" \o "CPython), an [open source](https://en.wikipedia.org/wiki/Open-source_software)[[34]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-34) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation). A [non-profit organization](https://en.wikipedia.org/wiki/Nonprofit_organization), the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation), manages and directs resources for Python and CPython development.

3.1.2 Data Analysing

**Data analysis** is a process of inspecting, [cleansing](https://en.wikipedia.org/wiki/Data_cleansing), [transforming](https://en.wikipedia.org/wiki/Data_transformation) and [modeling](https://en.wikipedia.org/wiki/Data_modeling) [data](https://en.wikipedia.org/wiki/Data) with the goal of discovering useful information, informing conclusions and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

[Data mining](https://en.wikipedia.org/wiki/Data_mining) is a particular data analysis technique that focuses on statistical modeling and knowledge discovery for predictive rather than purely descriptive purposes, while [business intelligence](https://en.wikipedia.org/wiki/Business_intelligence) covers data analysis that relies heavily on aggregation, focusing mainly on business information.[[2]](https://en.wikipedia.org/wiki/Data_analysis#cite_note-2) In statistical applications, data analysis can be divided into [descriptive statistics](https://en.wikipedia.org/wiki/Descriptive_statistics), [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) (EDA), and [confirmatory data analysis](https://en.wikipedia.org/wiki/Statistical_hypothesis_testing) (CDA). EDA focuses on discovering new features in the data while CDA focuses on confirming or falsifying existing [hypotheses](https://en.wikipedia.org/wiki/Hypotheses). [Predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics) focuses on application of statistical models for predictive forecasting or classification, while [text analytics](https://en.wikipedia.org/wiki/Text_analytics) applies statistical, linguistic, and structural techniques to extract and classify information from textual sources, a species of [unstructured data](https://en.wikipedia.org/wiki/Unstructured_data). All of the above are varieties of data analysis.

3.1.3 Python Libraries

* Numpy

**NumPy** (pronounced [/ˈnʌmpaɪ/](https://en.wikipedia.org/wiki/Help:IPA/English) ([*NUM-py*](https://en.wikipedia.org/wiki/Help:Pronunciation_respelling_key)) or sometimes [/ˈnʌmpi/](https://en.wikipedia.org/wiki/Help:IPA/English)[[3]](https://en.wikipedia.org/wiki/NumPy#cite_note-3)[[4]](https://en.wikipedia.org/wiki/NumPy#cite_note-4) ([*NUM-pee*](https://en.wikipedia.org/wiki/Help:Pronunciation_respelling_key))) is a library for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(math)), along with a large collection of [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [mathematical](https://en.wikipedia.org/wiki/Mathematics) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

* Pandas

In [computer programming](https://en.wikipedia.org/wiki/Computer_programming), **pandas** is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license).[[2]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-2) The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for data sets that include observations over multiple time periods for the same individuals.

Simply, Pandas is an open source, BSD-licensed library providing high-performance, easy to use data structure & data analysis tool for python programming language.

Pandas convert the files into a data frames, Pandas is mainly used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) in form of dataframes. Pandas allow importing data of various file formats such as csv, excel etc. Pandas allows various data manipulation operations such as groupby, join, merge, melt, concatenation as well as [data cleaning](https://en.wikipedia.org/wiki/Data_cleaning) features such as filling, replacing or imputing [null values](https://en.wikipedia.org/wiki/Nullable_type).

* Matplotlib

**Matplotlib** is a [plotting](https://en.wikipedia.org/wiki/Plotter) [library](https://en.wikipedia.org/wiki/Library_(computer_science)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language and its numerical mathematics extension [NumPy](https://en.wikipedia.org/wiki/NumPy). It provides an [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) [API](https://en.wikipedia.org/wiki/API) for embedding plots into applications using general-purpose [GUI toolkits](https://en.wikipedia.org/wiki/GUI_toolkit) like [Tkinter](https://en.wikipedia.org/wiki/Tkinter" \o "Tkinter), [wxPython](https://en.wikipedia.org/wiki/WxPython" \o "WxPython), [Qt](https://en.wikipedia.org/wiki/Qt_(software)), or [GTK+](https://en.wikipedia.org/wiki/GTK%2B). There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB), though its use is discouraged.[[3]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-3) [SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of Matplotlib.

Matplotlib was originally written by [John D. Hunter](https://en.wikipedia.org/wiki/John_D._Hunter), since then it has an active development community,[[4]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-4) and is distributed under a [BSD-style license](https://en.wikipedia.org/wiki/BSD_licenses). Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012,[[5]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-5) and further joined by Thomas Caswell.[[6]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-6)[[7]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-7)

Matplotlib 2.0.x supports Python versions 2.7 through 3.6. Python 3 support started with Matplotlib 1.2. Matplotlib 1.4 is the last version to support Python 2.6.[[8]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-8) Matplotlib has pledged to not support Python 2 past 2020 by signing the Python 3 Statement.

* Pickle

The [pickle](https://docs.python.org/3/library/pickle.html#module-pickle) module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream (from a [binary file](https://docs.python.org/3/glossary.html#term-binary-file) or [bytes-like object](https://docs.python.org/3/glossary.html#term-bytes-like-object)) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as “serialization”, “marshalling,” [1](https://docs.python.org/3/library/pickle.html#id7) or “flattening”; however, to avoid confusion, the terms used here are “pickling” and “unpickling”.

* Scikit-Learn

**Scikit-learn** (formerly **scikits.learn** and also known as **sklearn**) is a [free software](https://en.wikipedia.org/wiki/Free_software) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [library](https://en.wikipedia.org/wiki/Library_(computing)) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [programming language](https://en.wikipedia.org/wiki/Programming_language).[[3]](https://en.wikipedia.org/wiki/Scikit-learn#cite_note-jmlr-3) It features various [classification](https://en.wikipedia.org/wiki/Statistical_classification), [regression](https://en.wikipedia.org/wiki/Regression_analysis) and [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) algorithms including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine), [random forests](https://en.wikipedia.org/wiki/Random_forests), [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting), [*k*-means](https://en.wikipedia.org/wiki/K-means_clustering) and [DBSCAN](https://en.wikipedia.org/wiki/DBSCAN), and is designed to interoperate with the Python numerical and scientific libraries [NumPy](https://en.wikipedia.org/wiki/NumPy) and [SciPy](https://en.wikipedia.org/wiki/SciPy).

 Scikit-learn is largely written in Python, and uses [numpy](https://en.wikipedia.org/wiki/Numpy" \o "Numpy) extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in [Cython](https://en.wikipedia.org/wiki/Cython" \o "Cython) to improve performance. Support vector machines are implemented by a Cython wrapper around [LIBSVM](https://en.wikipedia.org/wiki/LIBSVM); logistic regression and linear support vector machines by a similar wrapper around [LIBLINEAR](https://en.wikipedia.org/wiki/LIBLINEAR). In such cases, extending these methods with Python may not be possible.

Scikit-learn integrates well with many other Python libraries, such as [matplotlib](https://en.wikipedia.org/wiki/Matplotlib) and [plotly](https://en.wikipedia.org/wiki/Plotly" \o "Plotly) for plotting, [numpy](https://en.wikipedia.org/wiki/NumPy" \o "NumPy) for array vectorization, [pandas](https://en.wikipedia.org/wiki/Pandas_(software)) dataframes, [scipy](https://en.wikipedia.org/wiki/SciPy" \o "SciPy), and many more.

3.1.4 Prediction Using Previous data

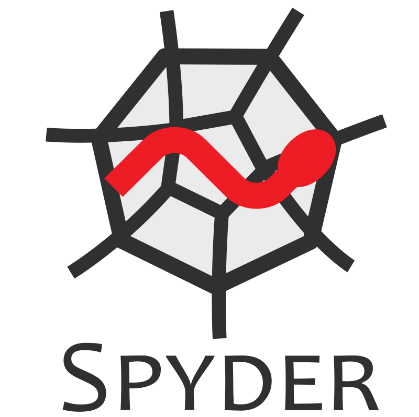
Predictive analytics helps teams in industries as diverse as finance, healthcare, pharmaceuticals, automotive, aerospace, and manufacturing.

* **Automotive** – Breaking new ground with autonomous vehicles  
  Companies developing driver assistance technology and new autonomous vehicles use predictive analytics to analyze sensor data from connected vehicles and to build driver assistance algorithms.
* **Aerospace** – Monitoring aircraft engine health  
  To improve aircraft up-time and reduce maintenance costs, an engine manufacturer created a real-time analytics application to predict subsystem performance for oil, fuel, liftoff, mechanical health, and controls.
* **Energy Production** – Forecasting electricity price and demand  
  Sophisticated forecasting apps use models that monitor plant availability, historical trends, seasonality, and weather.
* **Financial Services** – Developing credit risk models  
  Financial institutions use machine learning techniques and quantitative tools to predict credit risk.
* **Industrial Automation and Machinery** – Predicting machine failures  
  A plastic and thin film producer saves 50,000 Euros monthly using a health monitoring and predictive maintenance application that reduces downtime and minimizes waste.
* **Medical Devices** – Using pattern-detection algorithms to spot asthma and COPD  
  An asthma management device records and analyzes patients' breathing sounds and provides instant feedback via a smart phone app to help patients manage asthma and COPD.



3.1.5 Spyder I.D.E

* **Spyder** is an [open source](https://en.wikipedia.org/wiki/Open-source_software) cross-platform [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for scientific programming in the [Python language](https://en.wikipedia.org/wiki/Python_(programming_language)). Spyder integrates with a number of prominent packages in the scientific Python stack, including [NumPy](https://en.wikipedia.org/wiki/NumPy), [SciPy](https://en.wikipedia.org/wiki/SciPy), [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib), [pandas](https://en.wikipedia.org/wiki/Pandas_(software)), [IPython](https://en.wikipedia.org/wiki/IPython), [SymPy](https://en.wikipedia.org/wiki/SymPy) and [Cython](https://en.wikipedia.org/wiki/Cython), as well as other open source software. It is released under the [MIT license](https://en.wikipedia.org/wiki/MIT_license).
* Initially created and developed by Pierre Raybaut in 2009, since 2012 Spyder has been maintained and continuously improved by a team of scientific Python developers and the community.
* Spyder is extensible with first- and third-party plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, [Pylint](https://en.wikipedia.org/wiki/Pylint" \o "Pylint) and Rope. It is available cross-platform through [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)), on Windows, on macOS through [MacPorts](https://en.wikipedia.org/wiki/MacPorts" \o "MacPorts), and on major Linux distributions such as [Arch Linux](https://en.wikipedia.org/wiki/Arch_Linux), [Debian](https://en.wikipedia.org/wiki/Debian), [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system)), [Gentoo Linux](https://en.wikipedia.org/wiki/Gentoo_Linux), [openSUSE](https://en.wikipedia.org/wiki/OpenSUSE) and [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)).
* Spyder uses [Qt](https://en.wikipedia.org/wiki/Qt_(software)) for its GUI, and is designed to use either of the [PyQt](https://en.wikipedia.org/wiki/PyQt" \o "PyQt) or [PySide](https://en.wikipedia.org/wiki/PySide" \o "PySide) Python bindings. QtPy, a thin abstraction layer developed by the Spyder project and later adopted by multiple other packages, provides the flexibility to use either backend



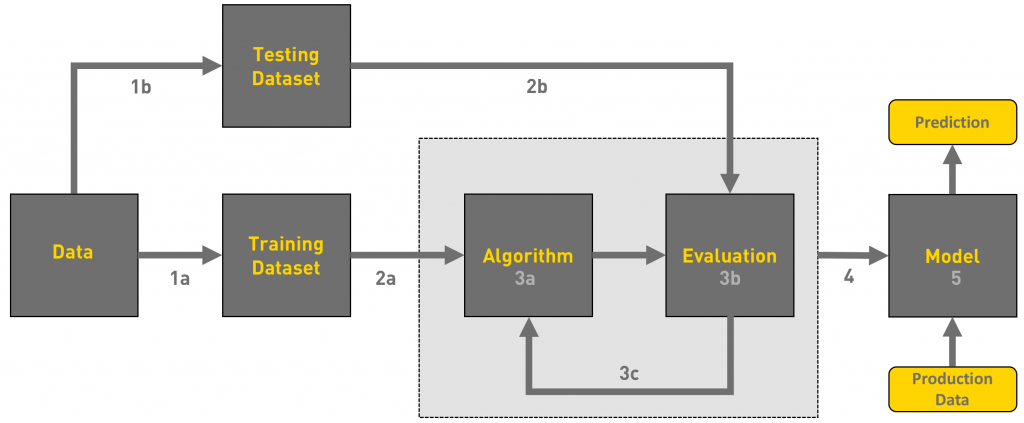
# Chapter 4

4.1 Methodology

In this project, we will discuss the workflow of a Machine learning project this includes all the steps required to build the proper machine learning project from scratch.

We will also go over data pre-processing, data cleaning, feature exploration and feature engineering and show the impact that it has on Machine Learning Model Performance. We will also cover a couple of the pre-modelling steps that can help to improve the model performance.

Python Libraries that would be need to achieve the task:  
1. Numpy  
2. Pandas  
3. Sci-kit Learn  
4. Matplotlib



# Understanding the machine learning workflow

We can define the machine learning workflow in 3 stages.

1. Gathering data
2. Data pre-processing
3. Researching the model that will be best for the type of data
4. Data Visualization
5. Prediction of Data

## ****What is the machine learning Model?****

The machine learning model is nothing but a piece of code; an engineer or data scientist makes it smart through training with data. So, if you give garbage to the model, you will get garbage in return, i.e. the trained model will provide false or wrong predictions.

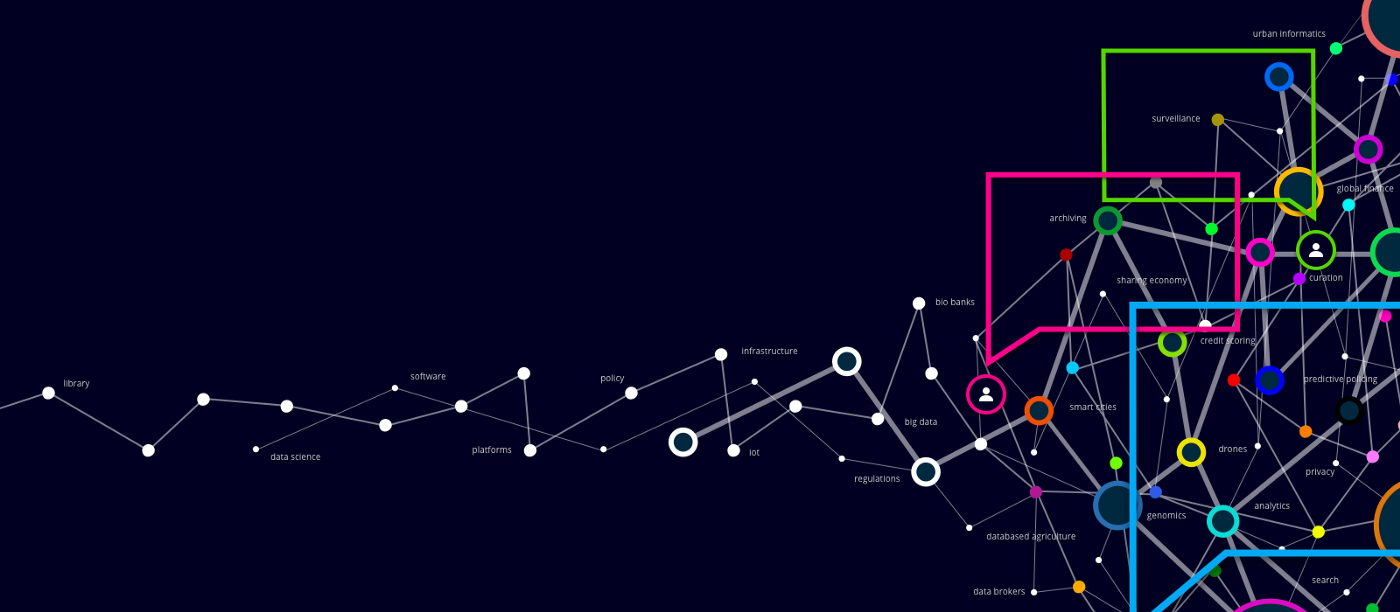
## 1.Gathering Data

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.

We can also use some free data sets which are present on the internet. [**Kaggle**](http://www.kaggle.com/)and [**UCI Machine learning Repository**](https://archive.ics.uci.edu/ml/datasets.html) are the repositories that are used the most for making Machine learning models. Kaggle is one of the most visited websites that is used for practicing machine learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

# 2.Data pre-processing

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis.



**What is data pre-processing?**

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn’t feasible for the analysis.  
Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

## ****Why do we need it?****

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects.

Most of the real-world data is messy, some of these types of data are:

1. **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).

2. **Noisy data:** This type of data is also called outliners, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.

3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

## Three Types of Data

1. Numeric e.g. income, age

2. Categorical e.g. gender, nationality

3. Ordinal e.g. low/medium/high

## How can data pre-processing be performed?

These are some of the basic pre — processing techniques that can be used to convert raw data.

1. **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.

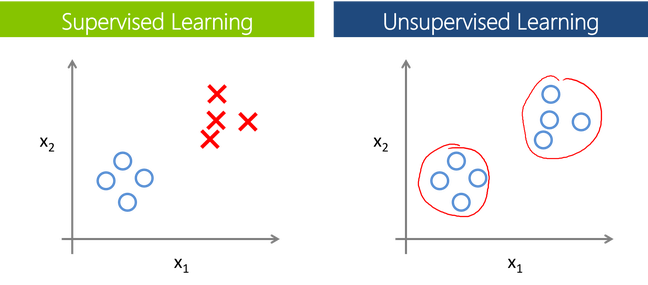
2. **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn’t be performed if there are a lot of missing values in the dataset.

3. **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.

4.**Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.

5. **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set.

# ****3.Researching the model that will be best for the type of data****

Our main goal is to train the best performing model possible, using the pre-processed data.

**Supervised Learning:**

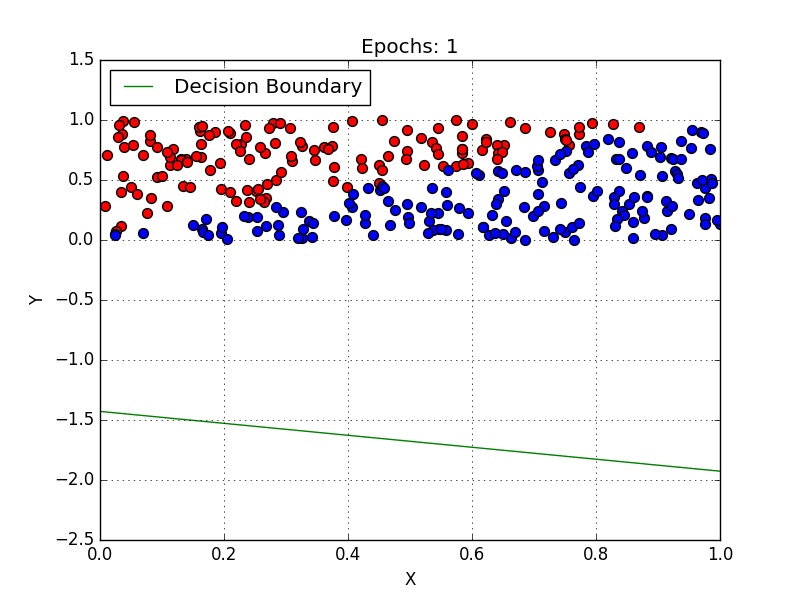
In Supervised learning, an AI system is presented with data which is labelled, which means that each data tagged with the correct label.

The supervised learning is categorized into 2 other categories which are “**Classification**” and “**Regression**”.

## Classification:

**Classification**problem is when the target variable is **categorical**(i.e. the output could be classified into classes — it belongs to either Class A or B or something else).

A classification problem is when the output variable is a category, such as “red” or “blue” , “disease” or “no disease” or “spam” or “not spam”.

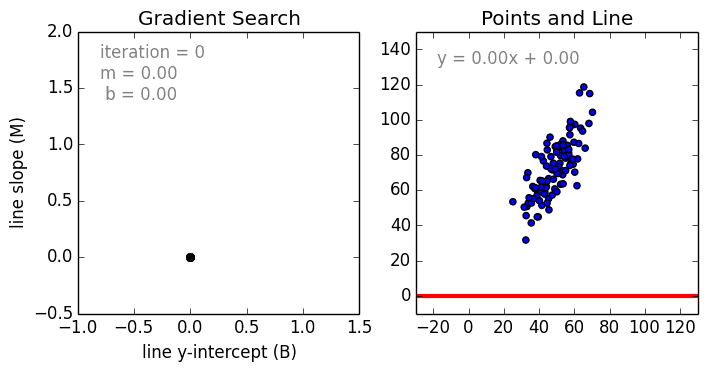
As shown in the above representation, we have 2 classes which are plotted on the graph i.e. red and blue which can be represented as ‘setosa flower’ and ‘versicolor flower’, we can image the X-axis as ther ‘Sepal Width’ and the Y-axis as the ‘Sepal Length’, so we try to create the [best fit line](https://mathbits.com/MathBits/TISection/Statistics1/LineFit.htm) that separates both classes of flowers.

These some most used classification algorithms.

* **K-Nearest Neighbor**
* **Naive Bayes**
* **Decision Trees/Random Forest**
* **Support Vector Machine**
* **Logistic Regression**

## Regression:

While a **Regression**problem is when the target variable is **continuous**(i.e. the output is numeric).

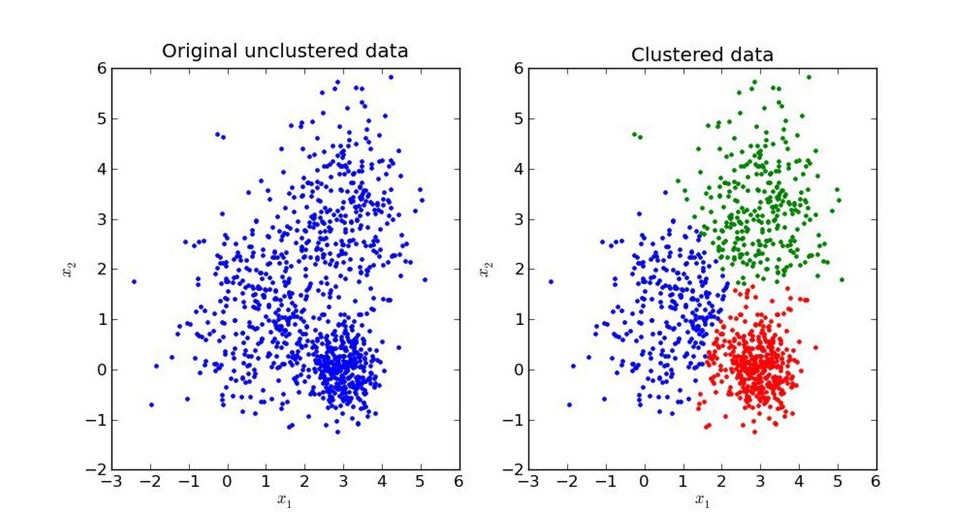


As shown in the above representation, we can imagine that the graph’s X-axis is the ‘Test scores’ and the Y-axis represents ‘IQ’. So we try to create the [best fit line](https://mathbits.com/MathBits/TISection/Statistics1/LineFit.htm) in the given graph so that we can use that line to predict any approximate IQ that isn’t present in the given data.

These some most used regression algorithms.

* **Linear Regression**
* **Support Vector Regression**
* **Decision Tress/Random Forest**
* **Gaussian Progresses Regression**
* **Ensemble Methods**

## Unsupervised Learning:

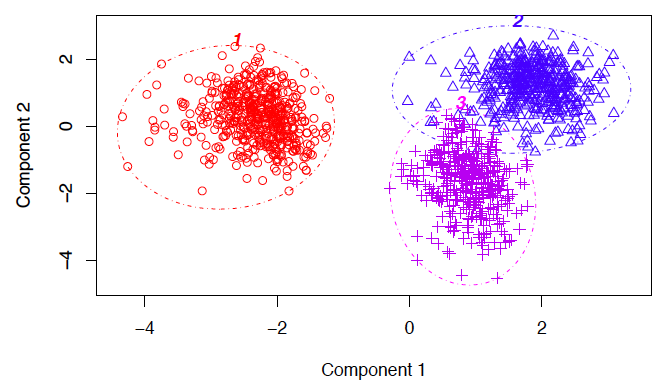


In unsupervised learning, an AI system is presented with unlabeled, un-categorized data and the system’s algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.

The unsupervised learning is categorized into 2 other categories which are “**Clustering**” and “**Association**”.

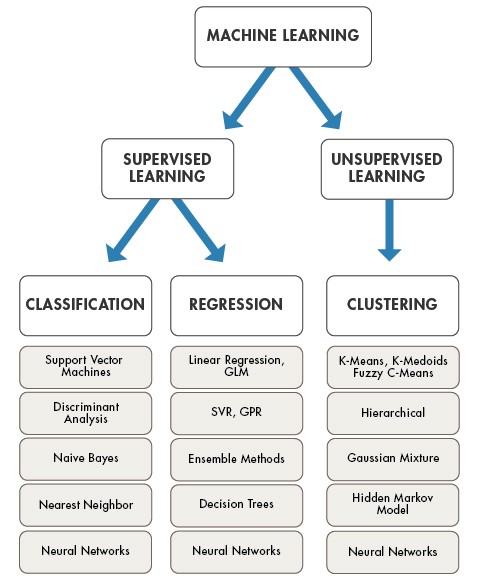
## Clustering:

A set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.



Methods used for clustering are:

* **Gaussian mixtures**
* **K-Means Clustering**
* **Boosting**
* **Hierarchical Clustering**
* **K-Means Clustering**
* **Spectral Clustering**

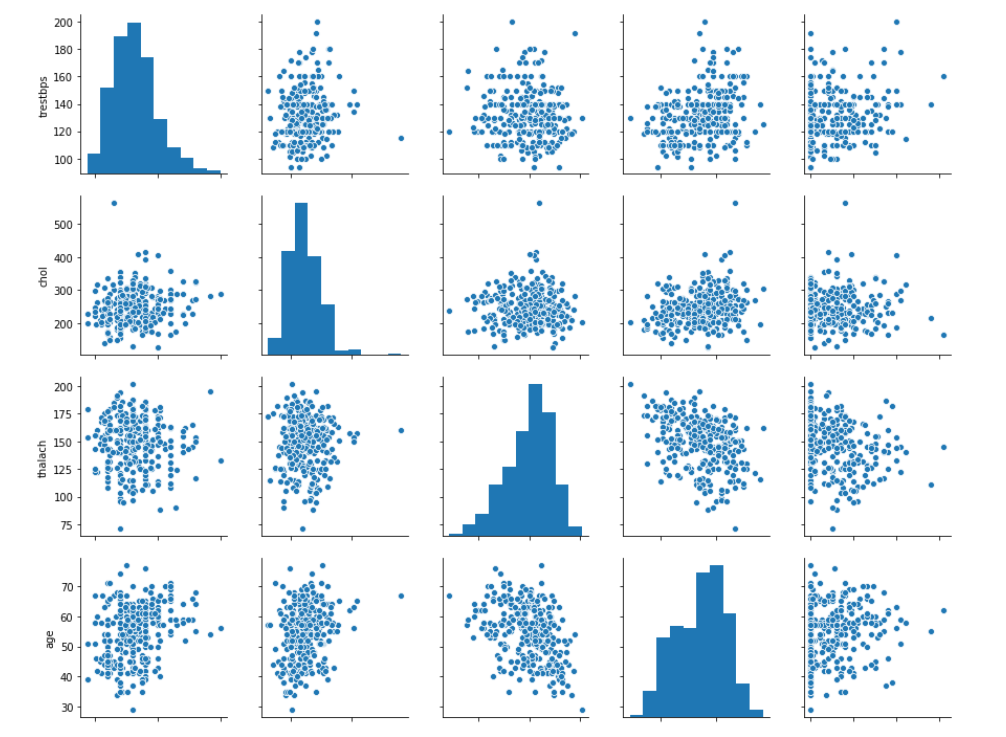
Overview of models under categories:

# 4. Data Visulaizatoin

**Data visualization** is the [graphic](https://en.wikipedia.org/wiki/Graphics) [representation](https://en.wikipedia.org/wiki/Representation_(arts)) of [data](https://en.wikipedia.org/wiki/Data). It involves producing images that communicate relationships among the represented data to viewers of the images. This communication is achieved through the use of a systematic [mapping](https://en.wikipedia.org/wiki/Map_(mathematics)) between graphic marks and data values in the creation of the visualization. This mapping establishes how data values will be represented visually, determining how and to what extent a property of a graphic mark, such as size or color, will change to reflect changes in the value of a datum.

To communicate information clearly and efficiently, data visualization uses [statistical graphics](https://en.wikipedia.org/wiki/Statistical_graphics), [plots](https://en.wikipedia.org/wiki/Plot_(graphics)), [information graphics](https://en.wikipedia.org/wiki/Infographic) and other tools. Numerical data may be encoded using dots, lines, or bars, to visually communicate a quantitative message.[[1]](https://en.wikipedia.org/wiki/Data_visualization#cite_note-ReferenceA-1) Effective visualization helps users analyze and reason about data and evidence. It makes complex data more accessible, understandable and usable. Users may have particular analytical tasks, such as making comparisons or understanding [causality](https://en.wikipedia.org/wiki/Causality), and the design principle of the graphic (i.e., showing comparisons or showing causality) follows the task. Tables are generally used where users will look up a specific measurement, while charts of various types are used to show patterns or relationships in the data for one or more variables.

Data visualization is both an art and a science.[[2]](https://en.wikipedia.org/wiki/Data_visualization#cite_note-2) It is viewed as a branch of [descriptive statistics](https://en.wikipedia.org/wiki/Descriptive_statistics) by some, but also as a [grounded theory](https://en.wikipedia.org/wiki/Grounded_theory) development tool by others. Increased amounts of data created by Internet activity and an expanding number of sensors in the environment are referred to as "[big data](https://en.wikipedia.org/wiki/Big_data)" or [Internet of things](https://en.wikipedia.org/wiki/Internet_of_things). Processing, analyzing and communicating this data present ethical and analytical challenges for data visualization. The field of [data science](https://en.wikipedia.org/wiki/Data_science) and practitioners called data scientists help address this challenge.[[3]](https://en.wikipedia.org/wiki/Data_visualization#cite_note-3)

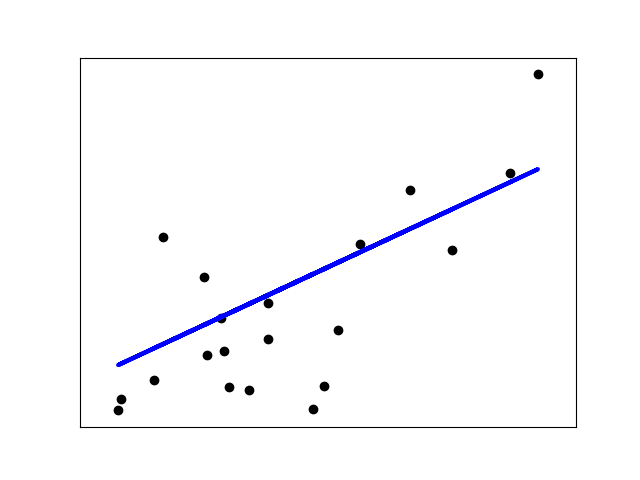


## Prediction of Data

[Supervised learning](https://scikit-learn.org/stable/supervised_learning.html#supervised-learning) consists in learning the link between two datasets: the observed data X and an external variable y that we are trying to predict, usually called “target” or “labels”. Most often, y is a 1D array of length n\_samples.

All supervised [estimators](https://en.wikipedia.org/wiki/Estimator) in scikit-learn implement a fit(X, y) method to fit the model and a predict(X) method that, given unlabeled observations X, returns the predicted labels y.

[**LinearRegression**](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression), in its simplest form, fits a linear model to the data set by adjusting a set of parameters in order to make the sum of the squared residuals of the model as small as possible.

[](https://scikit-learn.org/stable/auto_examples/linear_model/plot_ols.html)

Linear models: y=Xβ+ϵ

* X: data
* y: target variable
* β: Coefficients
* ϵ: Observation noise

## 

### 4.2 Data Collection & Data Required

## 0. How to collect data for machine learning if you don’t have any

The line dividing those who can play with ML and those who can’t is drawn by years of collecting information. Some organizations have been hoarding records for decades with such great success that now [they need trucks to move it](https://aws.amazon.com/snowmobile/) to the cloud as conventional broadband is just not broad enough.

For those who’ve just come on the scene, lack of data is expected, but fortunately, there are ways to turn that minus into a plus.

First, rely on open source data to initiate ML execution. There are mountains of data for machine learning around and some companies (like Google) are ready to give it away. We’ll talk about public dataset opportunities a bit later. While those opportunities exist, usually the real value comes from internally collected golden data nuggets mined from the business decisions and activities of your own company.

Second – and not surprisingly – now you have a chance to collect data the right way. The companies that started data collection with paper ledgers and ended with .xlsx and .csv files will likely have a harder time with data preparation than those who have a small but proud ML-friendly dataset. If you know the tasks that machine learning should solve, you can tailor a data-gathering mechanism in advance.

What about big data? It’s so buzzed, it seems like the thing everyone should be doing. Aiming at big data from the start is a good mindset, but big data isn’t about petabytes. It’s all about the ability to process them the right way. The larger your dataset, the harder it gets to make the right use of it and yield insights. Having tons of lumber doesn’t necessarily mean you can convert it to a warehouse full of chairs and tables. So, the general recommendation for beginners is to start small and reduce the complexity of their data.

## 1. Articulate the problem early

Knowing what you want to predict will help you decide which data may be more valuable to collect. When formulating the problem, conduct data exploration and try to think in the categories of classification, clustering, regression, and ranking that we talked about in our whitepaper on [business application of machine learning](https://www.altexsoft.com/whitepapers/machine-learning-bridging-between-business-and-data-science/). In layman’s terms, these tasks are differentiated in the following way:

**Classification**. You want an algorithm to answer binary yes-or-no questions (cats or dogs, good or bad, sheep or goats, you get the idea) or you want to make a multiclass classification (grass, trees, or bushes; cats, dogs, or birds etc.) You also need the right answers labeled, so an algorithm can learn from them.

**Clustering**. You want an algorithm to find the rules of classification and the number of classes. The main difference from classification tasks is that you don’t actually know what the groups and the principles of their division are. For instance, this usually happens when you need to segment your customers and tailor a specific approach to each segment depending on its qualities.

**Regression**. You want an algorithm to yield some numeric value. For example, if you spend too much time coming up with the right price for your product since it depends on many factors, regression algorithms can aid in estimating this value.

**Ranking**. Some machine learning algorithms just rank objects by a number of features. Ranking is actively used to recommend movies in video streaming services or show the products that a customer might purchase with a high probability based on his or her previous search and purchase activities.

It’s likely, that your business problem can be solved within this simple segmentation and you may start adapting a dataset accordingly. The rule of thumb on this stage is to avoid over-complicated problems.

## 2. Establish data collection mechanisms

Creating a data-driven culture in an organization is perhaps the hardest part of the entire initiative. We briefly covered this point in our story on [machine learning strategy](https://www.altexsoft.com/blog/datascience/machine-learning-strategy-7-steps/). If you aim to use ML for predictive analytics, the first thing to do is combat data fragmentation.

For instance, if you look at [travel tech](https://www.altexsoft.com/travel-technology/) – one of AltexSoft’s key areas of expertise – data fragmentation is one of the top analytics problems here. In hotel businesses, the departments that are in charge of physical property get into pretty intimate details about their guests. Hotels know guests’ credit card numbers, types of amenities they choose, sometimes home addresses, room service use, and even drinks and meals ordered during a stay. The website where people book these rooms, however, may treat them as complete strangers.

This data gets siloed in different departments and even different tracking points within a department. Marketers may have access to a CRM but the customers there aren’t associated with web analytics. It’s not always possible to converge all data streams if you have many channels of engagement, acquisition, and retention, but in most cases it’s manageable.

Another point here is the human factor. Data collection may be a tedious task that burdens your employees and overwhelms them with instructions. If people must constantly and manually make records, the chances are they will consider these tasks as yet another bureaucratic whim and let the job slide. For instance, Salesforce provides a decent toolset to track and analyze salespeople activities but manual data entry and activity logging alienates salespeople.

## 3. Format data to make it consistent

Data formatting is sometimes referred to as the file format you’re using. And this isn’t much of a problem to convert a dataset into a file format that fits your machine learning system best.

We’re talking about format consistency of records themselves. If you’re aggregating data from different sources or your dataset has been manually updated by different people, it’s worth making sure that all variables within a given attribute are consistently written. These may be date formats, sums of money (4.03 or $4.03, or even 4 dollars 3 cents), addresses, etc. The input format should be the same across the entire dataset.

## 4. Reduce data

It’s tempting to include as much data as possible, because of… well, big data! That’s wrong-headed. Since you know what the target attribute (what value you want to predict) is, common sense will guide you further. You can assume which values are critical and which are going to add more dimensions and complexity to your dataset without any predictive contribution. This approach is called **attribute sampling**.

For example, you want to predict which customers are prone to make large purchases in your online store. The age of your customers, their location, and gender can be better predictors than their credit card numbers. But this also works another way. Consider which other values you may need to collect to uncover more dependencies. For instance, adding bounce rates may increase accuracy in predicting conversion.

That’s the point where domain expertise plays a big role. Returning to our beginning story, not all data scientists know that asthma can cause pneumonia complications. The same works with reducing large datasets. If you haven’t employed a unicorn who has one foot in healthcare basics and the other in data science, it’s likely that a data scientist might have a hard time understanding which values are of real significance to a dataset.

Another approach is called **record sampling**. This implies that you simply remove records (objects) with missing, erroneous, or less representative values to make prediction more accurate. The technique can also be used in the later stages when you need a model prototype to understand whether a chosen machine learning method yields expected results.

You can also reduce data by **aggregating** it into broader records by dividing the entire attribute data into multiple groups and drawing the number for each group. Instead of exploring the most purchased products of a given day through five years of online store existence, aggregate them to weekly or monthly scores. This will help reduce data size and computing time without tangible prediction losses.

## 5. Complete data cleaning

Since missing values can tangibly reduce prediction accuracy, make this issue a priority. In terms of machine learning, assumed or approximated values are “more right” for an algorithm than just missing ones.  Even if you don’t know the exact value, methods exist to better “assume” which value is missing or bypass the issue. How to сlean data? Choosing the right approach also heavily depends on data and the domain you have:

* Substitute missing values with dummy values, e.g. n/a for categorical or 0 for numerical values
* Substitute the missing numerical values with mean figures
* For categorical values, you can also use the most frequent items to fill in.

If you use some ML as a service platform, data cleaning can be automated. For instance, Azure Machine Learning allows you to choose among available techniques, while Amazon ML will do it without your involvement at all. Have a look at our [MLaaS systems comparison](https://www.altexsoft.com/blog/datascience/comparing-machine-learning-as-a-service-amazon-microsoft-azure-google-cloud-ai-ibm-watson/) to get a better idea about systems available on the market.

## 6. Decompose data

Some values in your data set can be complex and decomposing them into multiple parts will help in capturing more specific relationships. This process is actually the opposite to reducing data as you have to add new attributes based on the existing ones.

For example, if your sales performance varies depending on the day of a week, segregating the day as a separate categorical value from the date (Mon; 06.19.2017) may provide the algorithm with more relevant information.

## 7. Rescale data

Data rescaling belongs to a group of **data normalization** procedures that aim at improving the quality of a dataset by reducing dimensions and avoiding the situation when some of the values overweight others. What does this mean?

Imagine that you run a chain of car dealerships and most of the attributes in your dataset are either categorical to depict models and body styles (sedan, hatchback, van, etc.) or have 1-2 digit numbers, for instance, for years of use. But the prices are 4-5 digit numbers ($10000 or $8000) and you want to predict the average time for the car to be sold based on its characteristics (model, years of previous use, body style, price, condition, etc.)  While the price is an important criterion, you don’t want it to overweight the other ones with a larger number.

In this case, **min-max normalization** can be used. It entails transforming numerical values to ranges, e.g. from 0.0 to 5.0 where 0.0 represents the minimal and 5.0 the maximum values to even out the weight of the price attribute with other attributes in a dataset.

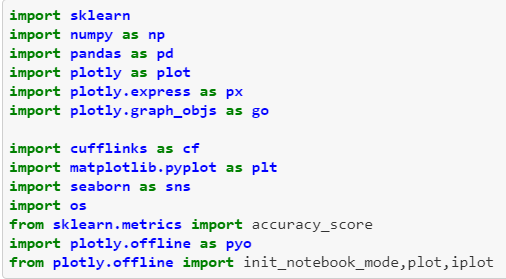
A bit simpler approach is **decimal scaling**. It consists of scaling data by moving a decimal point in either direction for the same purposes.

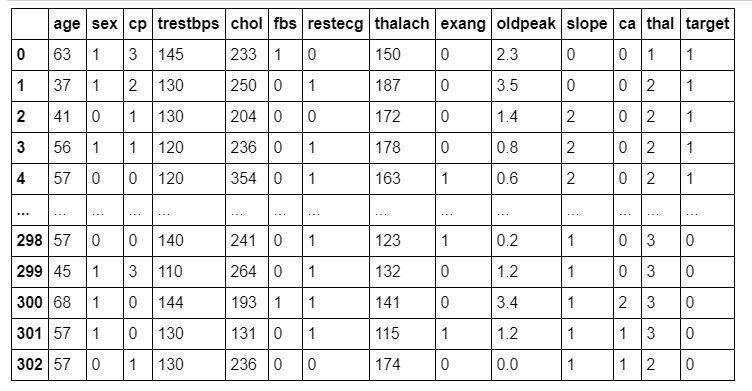
## 8. Discretize data

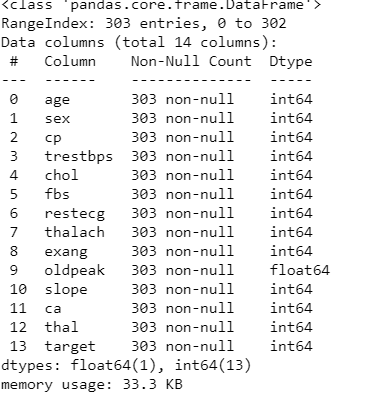
Sometimes you can be more effective in your predictions if you turn numerical values into categorical values. This can be achieved, for example, by dividing the entire range of values into a number of groups.If you track customer age figures, there isn’t a big difference between the age of 13 and 14 or 26 and 27. So these can be converted into relevant age groups. Making the values categorical, you simplify the work for an algorithm and essentially make prediction more relevant.

# Chapter 5 : Overview of Project

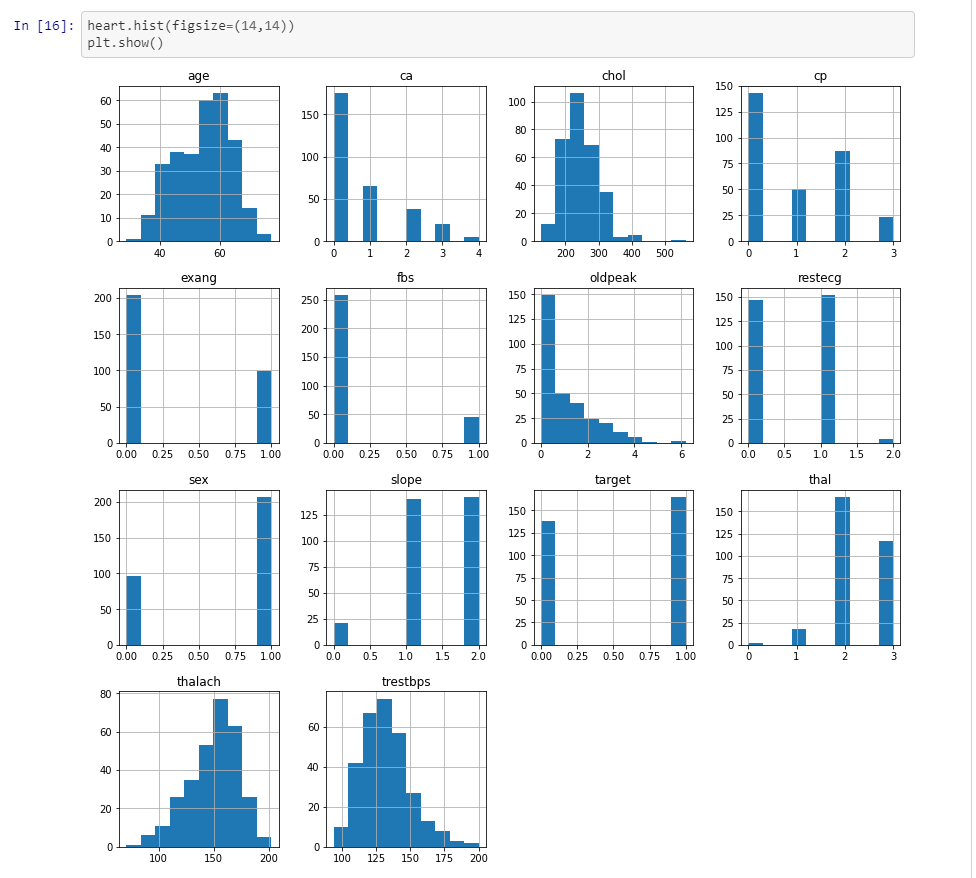
* **Mentioning all the libraries :**

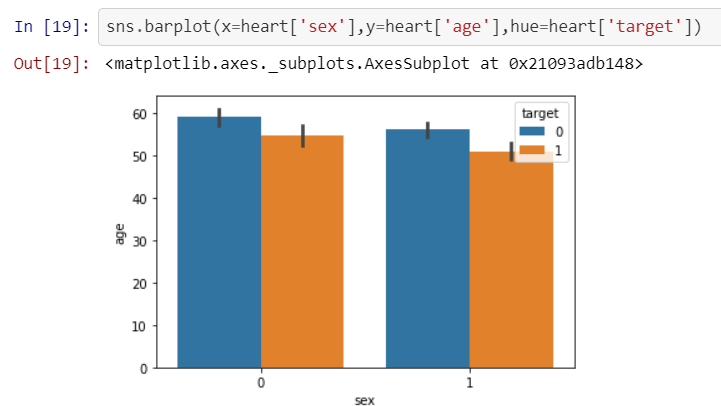


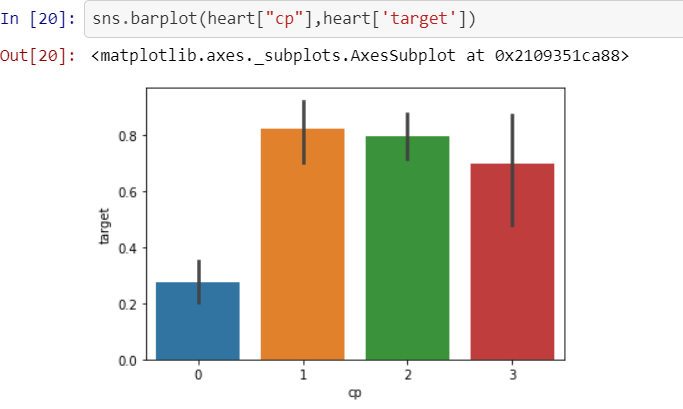
* **Data abstraction from the comma separated values :**
* **Finding the null values in data :**

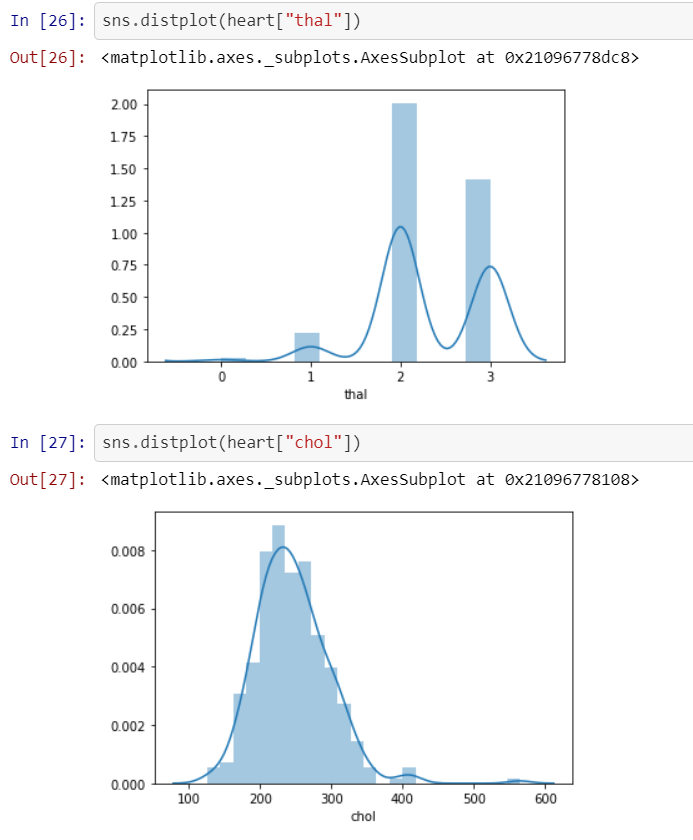


* **Data Visualization :**

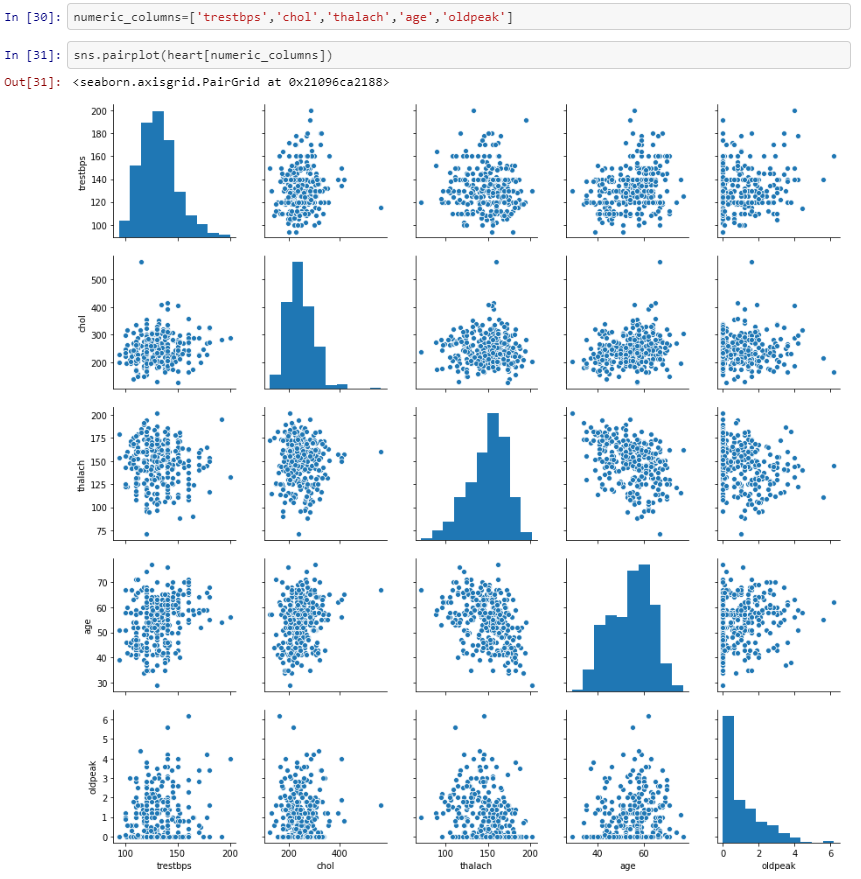


using parameter sex, age, target and cp:



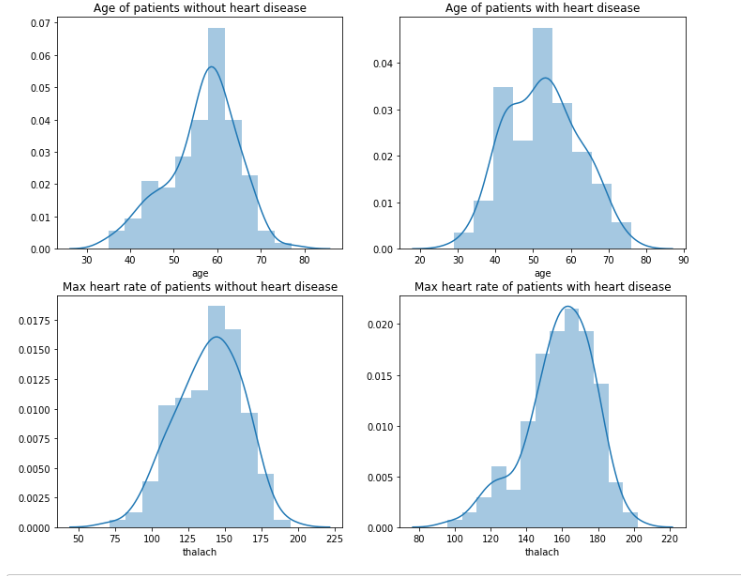
using thal parameter and chol :

dotted and bar representation using trestbps, chol, thalach, age and oldpeaks

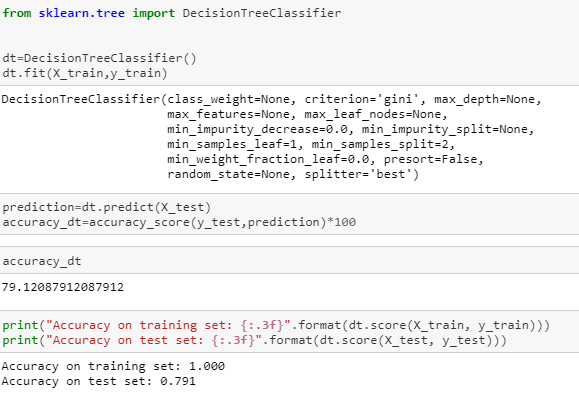


Represtation using :

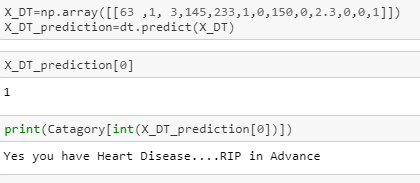
1. age of patients without heart diseases
2. age of patients with heart diseases
3. maximum heart rate without heart diseases
4. maximum heart rate with heart diseases



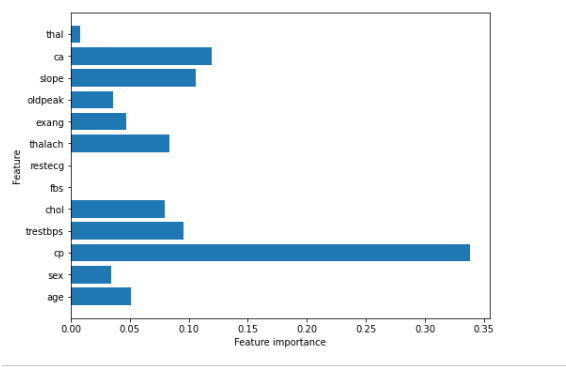
* Training the model using data:



* Prediction the outcomes:



* Factors that affect the prediction most:



That is cp

# Chapter 7

# Conclusion

* This project has introduced you to Machine Learning. Now, you know that Machine Learning is a technique of training machines to perform the activities a human brain can do, albeit bit faster and better than an average human-being. Today we have seen that the machines can beat human champions in games such as Chess, AlphaGO, which are considered very complex. You have seen that machines can be trained to perform human activities in several areas and can aid humans in living better lives.
* Machine Learning can be a Supervised or Unsupervised. If you have lesser amount of data and clearly labelled data for training, opt for Supervised Learning. Unsupervised Learning would generally give better performance and results for large data sets. If you have a huge data set easily available, go for deep learning techniques.
* In this project, we will get a salary according to your test score, Interview Score and Age of Experience

# REFERENCES

1. Baldi, P. and Brunak, S. (2002). Bioinformatics: A Machine Learning Approach. Cambridge, MA: MIT Press.  
   This book offers a good coverage of machine learning approaches - especially neural networks and hidden Markov models in bioinformatics.
2. Baldi, P., Frasconi, P., Smyth, P. (2003). Modeling the Internet and the Web - Probabilistic Methods and Algorithms. New York: Wiley.  
   A good introduction to machine learning approaches to text mining and related applications on the web.
3. Bishop, C. M. Neural Networks for Pattern Recognition. New York: Oxford University Press (1995).  
   This book offers a good coverage of neural networks
4. Chakrabarti, S. (2003). Mining the Web, Morgan Kaufmann.
5. Cohen, P.R. (1995) [Empirical Methods in Artificial Intelligence](http://babs.cs.umass.edu/emai.html). Cambridge, MA: MIT Press. This is an excellent reference on experiment design, and hypothesis testing, and related topics that are essential for empirical machine learning research.
6. Cowell, R.G., Dawid, A.P., Lauritzen, S.L., and Spiegelhalter,D.J. (1999). Graphical Models and Expert Systems.Berlin: Springer.  
   This is a very good introduction to probabilistic graphical models.
7. Cristianini, N. and Shawe-Taylor, J. (2000). An Introduction to Support Vector Machines. London: Cambridge University Press.  
   This is an excellent introduction to kernel methods for pattern classification.
8. Duda, R., Hart, P., and Stork, D. (2001). Pattern Classification. New York: Wiley.  
   This is a good text with primary emphasis on statistical methods for pattern classification.
9. Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of Statistical Learning - Data Mining, Inference, and Prediction. Berlin: Springer-Verlag.  
   This is an excellent text that explains some of the key ideas in machine learning within a statistical framework.
10. Jordan, M. (2003). Probabilistic Graphical Models. Professor Jordan has kindly shared a pre-publication draft.  
    This text has an excellent coverage of generative and discriminative probabilistic models for classification.
11. Kearns, M. and Vazirani, U. (1994). Computational Learning Theory. Cambridge, MA: MIT Press.  
    This, although a bit dated, is an excellent introduction to learning theory.
12. Mitchell, T. (1997). [Machine Learning](http://www-2.cs.cmu.edu/~tom/mlbook.html). New York: Mc Graw-Hill.  
    This is, although a bit dated, an excellent introduction to Machine Learning.
13. Russel, S. and Norvig, P. (2003). Artifiical Intelligence: A Modern Approach. 2nd Edition. New York: Prentice-Hall.  
    This is an excellent text on Artificial Intelligence, with several introductory chapters on Machine Learning.
14. Tan, P-N., Steinbach, M., and Kumar, V. (2004). Introduction to Data Mining. New York: Addison-Vesley.