**Exploratory Data Analysis and Machine Learning for Loan Eligibility Prediction**

Internship Semester at NIT Kurukshetra

Submitted by:-

**Akshat Lakhara**

**Roll No. - 11914108**

Under the Guidance of-

**Dr. J.S Lather**

Professor

NIT Kurukshetra



**DEPARTMENT OF ELECTRICAL ENGINEERING**

**NATIONAL INSTITUTE OF TECHNOLOGY KURUKSHETRA**

**KURUKSHETRA - 136119 (INDIA)**

**CERTIFICATE**

This is to certify that project work embodied in this report entitled **“Exploratory Data Analysis and Machine Learning for Loan Eligibility Prediction”** was carried out by **Akshat Lakhara** studying at **National Institute of Technology Kurukshetra.**

This project work has been carried out under my guidance and supervision and it is up to my satisfaction.

Date: -

Place: - Kurukshetra

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Project Mentor

**Dr. J.S. Lather**

Professor

NIT Kurukshetra

**Declaration**

I hereby declare that the project **work “Exploratory Data Analysis and Machine Learning for Loan Eligibility Prediction”**, is an authentic record of my own work which was carried out at NIT KURUKSHETRA.

The work presented has not been submitted by me for the reward of any degree of this or any other institute. I will be liable for full or partial violation copyright or intellectual property right if found at any stage.

These projects are in requirement of my 6th semester that I had done under the able guidance of my faculty and project mentor **Dr. J.S. Lather**, Professor Department of Electrical Engineering National Institute of Technology Kurukshetra.

Date:-

Place: Kurukshetra

(Akshat Lakhara)

Roll no. 11914108

**Acknowledgement**

I would like to express my deep gratitude and appreciation to all the people who have helped and supported me in the process of my internship completion. Without their help and support , I would not have been able to reach this level of satisfaction with what I have learned and accomplished during my internship.

First and foremost, I would like to express my deep sense of respect and gratitude towards my project mentor, Dr. J.S. Lather , Professor , Department of Electrical Engineering , NIT Kurukshetra, for mentoring and guiding me throughout the internship. I am really grateful to be taught the core concepts and the working behind the models of machine learning by him. I am very thankful for his endless support, motivation, patience and encouragement during the internship. His professional knowledge and faith in me were very important and gave me the strength to conclude this work.

On personal note, my heartfelt gratitude for my friends, who have motivated and supported me throughout this uphill task. Without their blessings this work could not have been accomplished. Finally yet importantly, I dedicate my project work to my parents whose support and encouragement has made this work possible.

Place: Kurukshetra

(Akshat Lakhara)

Roll no. 11914108

Date:-

**Abstract**

This report is the summation of all the theoretical and practical knowledge which I gained during my 6-month internship at NIT Kurukshetra. I worked in the “Electrical Department”. The project assigned to me was “Exploratory Data Analysis and Machine Learning for Loan Eligibility Prediction” .

In the first phase of my internship, I learned all the basics and underlying principles of machine learning and its a working. I also learned python programming language to code my understanding of the topics by practically implementing them.

In the second step, I delved into the inner workings of several machine learning models and the mathematics that underpin them. Statistics, Vectors, Linear Algebra, and Probability were among the subjects I studied. In addition, I learned about Numpy, Pandas, Matplotlib, Seaborn, Scikit-learn, and other libraries. The arithmetic and machine learning were then implemented in Python using these packages in Notebooks.

In the final phase, I picked a data collection and applied my knowledge to assess its information. To gain a strong comprehension of the data, I analyzed it and made inferences from it. After that, I experimented with various supervised learning models in order to get the most accurate results from the data.

This report covers all these phases. The aim of my project was to analyse, understand and formulate the best model that can predict the outcome of an unseen test data with high accuracy. The project background involves the knowledge of statistics, linear algebra, matrices etc along with different machine learning libraries like seaborn and matplotlib for analyzing the data, pandas for handling the data set, numpy for numerical calculations and scikit-learn for implementing models.

This project was part of my 6th semester project period. Through this project, I got to learn about the industry used software and their working. I also learned about deployment of the model and how they function under real world situations.

**Table of contents**

**Contents**

Front Page

Certificate

Declaration

Acknowledgement

Abstract

Table of Contents

Unit-1

Chapter 1: Machine Learning

1. Difference between Traditional and Machine Learning Approach

Chapter 2: Types of Machine Learning

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Chapter 3: Challenges of Machine Learning

Unit-2

Chapter 1: Steps to Machine Learning

1. Data Collection
2. Exploratory Data Analysis
3. Data Preprocessing
4. Data Splitting

Unit-3

Chapter 1: Machine Learning Libraries

1. Numpy
2. Pandas
3. Matplotlib
4. Seaborn
5. Scikit-Learn

**Page no.**

i

ii

iii

iv

v

vi

1

1

1

3

3

5

6

8

12

12

12

14

17

21

22

22

22

22

23

23

24

**Table of contents**

Chapter 2: Supervised Machine Learning Models

1. Linear Regression
2. Logistic Regression
3. Naive Bayes Classifier
4. K-Nearest Neighbor
5. Decision Tree
6. Random Forest Classifier

Chapter 3: Unsupervised Machine Learning Models

1. K-Means Clustering
2. Hierarchical Clustering

25

25

27

29

31

33

35

37

37

39

**UNIT - 1**

**CHAPTER 1: MACHINE LEARNING**

Machine Learning is the science (and art) of programming computers so they can learn from data. A more general definition will be that:

**Machine Learning is the field of study that gives computers the ability to learn**

**without being explicitly programmed.**

And a more engineering-oriented definition id:

A computer program is said to learn from experience E with respect to some task T

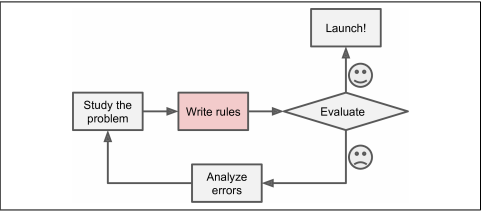
and some performance measure P, if its performance on T, as measured by P, improves with experience E.

In a very layman manner, Machine Learning(ML) can be explained as automation and improving the learning process of computers based on their experiences without being actually programmed that is without any human assistance. The process starts with feeding good quality data and then training our machine by building machine learning models using the data and different algorithms.

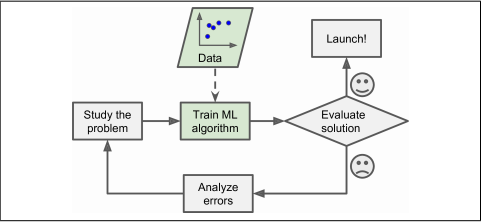
* 1. **Difference between Traditional and machine learning approach**

But machine learning is different from traditional programming as:-

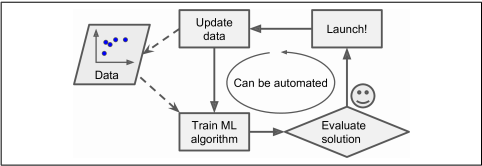
1. **Traditional Programming**: We feed in Data (input) and Program (logic) to get the output from the machine.The traditional approach is that you analyze the data and write set of rules to best define the solution of the data. It can be seen as:-

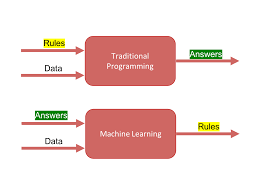


1. **Machine Learning**: We feed in Data (Input) + Output to the machine and the machine creates its own logic, which can further be used to test newer data. The working of a machine learning model can be defined as:-



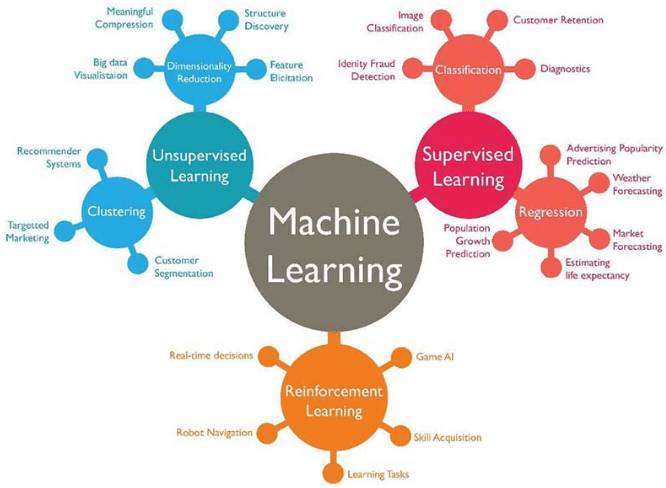
Moreover, newer data feed to traditional programming approach will lead to high error results as the program had no knowledge of the new data and user need to rewrite newer set of rules. On the other hand a machine learning model learns from the data and output and can be updated with newer data.





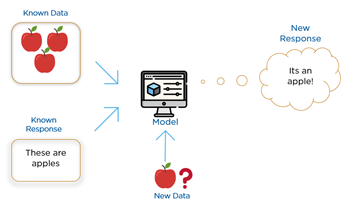
Also , traditional approach can be reliable for small and less complex data , for example data of students heights and weights. But when the data is much more complex , defining the set of rules to becomes difficult. This is where machine learning shines as it sets the rules based on the data input and output.

**CHAPTER 2: TYPES OF MACHINE LEARNING**



Within the field of machine learning, there are majorly three types - supervised, unsupervised and reinforcement learning.

**Supervised Learning**

In supervised learning, you have a known set of inputs (features) and a known set of outputs (labels).Supervised learning is typically done in the context of classification, when we want to map input to output labels, or regression, when we want to map input to a continuous output. Common algorithms in supervised learning include:-

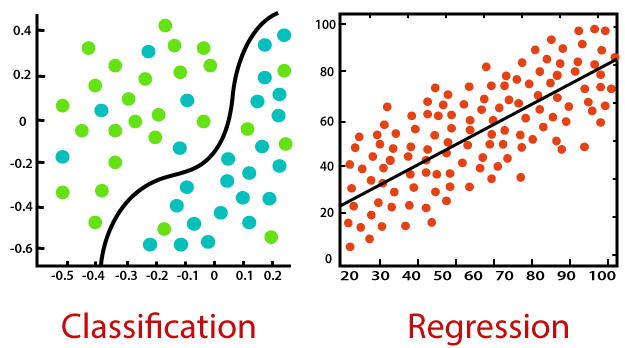
* Linear Regression
* Logistic regression
* Naive Bayes
* Support Vector Machines
* Decision Tree
* Random Forests

In both regression and classification, the goal is to find specific relationships or structure in the input data that allow us to effectively produce correct output data.

****Regression**** models a target prediction value based on independent variables. It is mostly used for finding out the **relationship between variables** and **forecasting**. Regression can be used to estimate/ predict **continuous values** .

****Classification****means to **group** the output into a class. If the data set is **discrete** or **categorical** then it is a classification problem.

Classification separates the data while Regression fits the data.



Some applications of supervised learning are:-

1. **Face Recognition**:- A supervised learning algorithm can predict the face of newer data by comparing it the earlier fed data. Google pictures and Facebook use supervised learning to recognize the face from the posted pictures and tag them accordingly.
2. **Email Spam Classification**:- Modern Email system segregates spam emails from non-spam ones by applying supervised machine learning to it. Data of earlier spam emails are fed in order to understand what all emails can be categorized under spam tag.
3. **Price Prediction**:- Supervised machine learning model can predict the price of an item/object/ house etc if trained to do so. It is seen that machine learning models do much better in predicting the price then a normal human.

**Unsupervised Learning**

In unsupervised learning, problems are approached with little or no idea about the results or the data does not contain feature labels. We can **derive structure** from data where we don’t necessarily know the effect of the variables.

Some of the most important unsupervised learning algorithms are:-

* Clustering

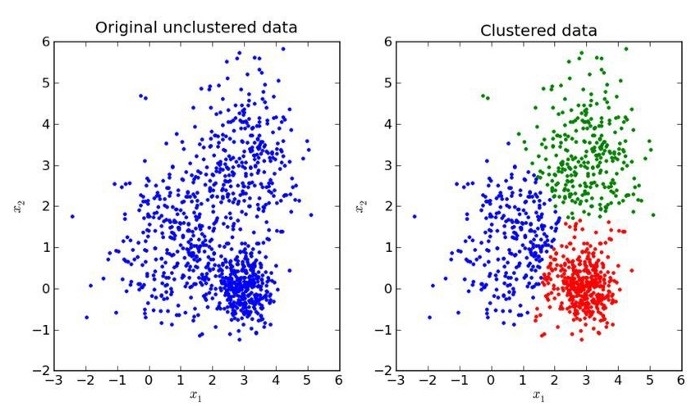
1. K-Means
2. DBSCAN

* Dimensionality reduction

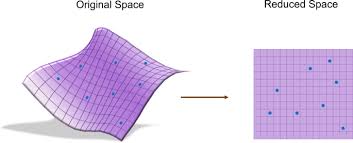
1. Principle Component Analysis
2. Locally-Linear Embedding (LLE)

* Anomaly detection

**Clustering** is the task of dividing the population of data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

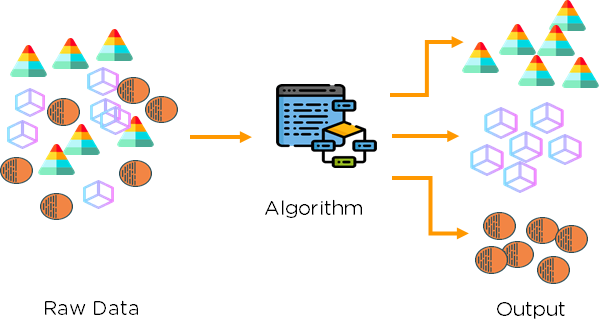


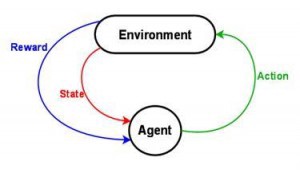
**Dimensionality reduction** refers to techniques that reduce the number of input variables in a data. More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality.



Some applications of unsupervised machine learning:-

1. **Recommender Systems**:- Softwares reads user usage and activity on its site and recommend them similar types of content. It suggest best content based on the user nature by understanding the data.
2. **Buying Habits**:- Many of the product based companies use user data to understand the usage and buying pattern of its user. This data is fed to an unsupervised learning model and next time when a new user buys a stuff, this model recommends him to check out other products of same type.

****

**Reinforcement learning**

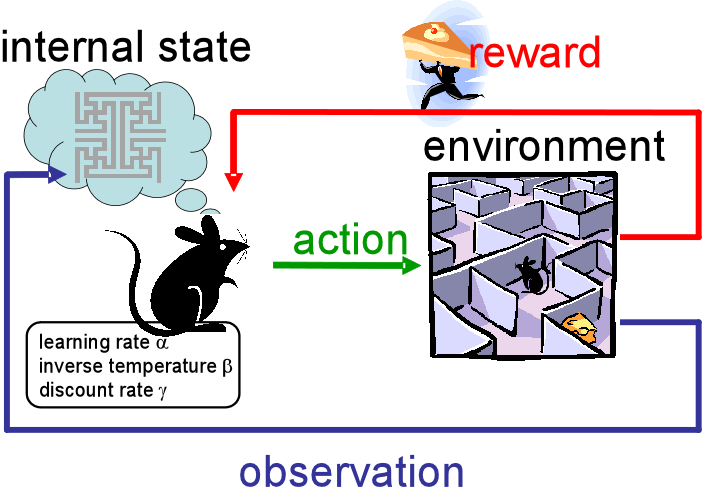
Reinforcement Learning is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machine to find the best possible behaviour or path it should take in a specific situation. Unlike supervised learning , reinforcement learning does not have specific labels rather reinforcement agent decides what to do to perform the given task.

Components of reinforcement learning are:-

1. **Agent:**- It is an object that is perceiving its environment through sensors and act upon the environment.
2. **States**:- The observation, the agent does on the environment after performing an action.
3. **Action**:- An action that the agent performs on the environment based on its observation.
4. **Reward**:- The feedback the agent receives based on the action it performed. If the feedback is positive, it receives a reward and if the feedback is negative, it receives a punishment.
5. **Environment**:- The environment gives agent a particular state according to its position.This learning process continues until the goal is achieved or some other condition is met.

Some applications of reinforcement learning are:-

1. Self-Driving cars and robots.
2. Industry automation.
3. News recommendations
4. Gaming

****

**CHAPTER 3: CHALLENGES OF MACHINE LEARNING**

1. **Insufficient Quantity of Training Data**:- A machine learning model learns from the data it is fed and if the data is small, then the model will not be able to learn the data properly leading to inaccurate results. In order to avoid this challenge, a machine learning model is fed enough data to learn or else the some underlying information is taken out as new features to help model learn.
2. **Non representative Training Data**:- If the training data has large amount of information for a particular type of label and few information about others, then the machine learning model will be able to predict properly for outcomes with large data but for less amount of information outcome it wont be able to learn, and lead to wrong outcomes. If the data is not representative we cant generalize it.
3. **Poor Quality Data**:- If your training data is full of errors, outliers, and noise (e.g., due to poor quality measurements), it will make it harder for the system to detect the underlying patterns, so your system is less likely to perform well. It is often well worth the effort to spend time cleaning up your training data.
4. **Irrelevant Features**:- As the saying goes: garbage in, garbage out. Your system will only be capable of learning if the training data contains enough relevant features and not too many irrelevant ones. If the features are irrelevant , then your model will waste it resources to process and learn from it.

(**Bias**:- Difference between prediction values made by the model and actual values/expected values of the model while training phase is called the bias. It can be defined as the inability of the machine learning algorithm to capture the true relationship between data points.

* **Low Bias**:- A low bias model will make fewer assumptions about the form of the target function.
* **High Bias**:- A model with a high bias makes more assumptions, and the model becomes unable to capture the important features of our dataset. **A high bias model also cannot perform well on new data.**

**Ways to reduce High Bias:-**

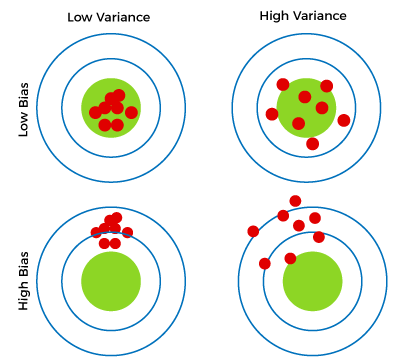
* **Increase the input features as the model is underfitting.**
* **Decrease Regularization term.**
* **Use of complex models, such as including polynomial features.**

**Variance**:- The variance would specify the amount of variation in the prediction if the different training data was used. In simple words, variance tells that how much a random variable is different from its expected value.

* **Low variance**:- **Low variance** means there is a small variation in the prediction of the target function with changes in the training data set.
* **High variance**:- **High variance** shows a large variation in the prediction of the target function with changes in the training dataset.

Ways to reduce High variance:-

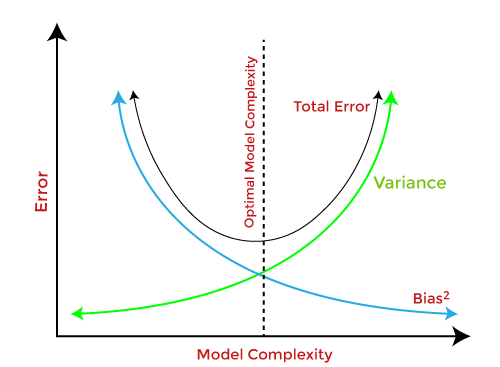
* Reduce the input features or parameters.
* Increase the training data.
* Increase Regularization term.
* Do not use complex model.



Bias-Variance Trade-off

For an accurate prediction of the model, algorithms need a low variance and low bias. But this is not possible because bias and variance are related to each other:

* If we decrease the variance, it will increase the bias.
* If we decrease the bias, it will increase the variance.



Bias-Variance trade-off is a central issue in supervised learning. Ideally, we need a model that accurately captures the regularities in training data and simultaneously generalizes well with the unseen dataset. Unfortunately, doing this is not possible simultaneously. Because a high variance algorithm may perform well with training data, but it may lead to overfitting to noisy data. Whereas, high bias algorithm generates a much simple model that may not even capture important regularities in the data. So, we need to find a sweet spot between bias and variance to make an optimal model.)

1. **Over-fitting the Training Data**:- A statistical model is called overfitted when we train it with a lot of data. When model gets trained with so much data, it also starts learning from the inaccurate data and noise. In short , an overfit model will have high variance and low bias and model has perfectly memorized the data.Techniques to reduce overfitting:-

* Increase Training Data.
* Reduce model complexity.
* Early stoppage during training phase.
* Ridge and Lasso Regularization.

1. **Underfitting the Training Data**:- A machine learning is called underfit if it cannot capture the underlying trend of the data. It occurs when our model or algorithm does not fit well enough. This mostly happens when you have less data to build a model or when non-linear data. In short , an underfit model will have high bias and low variance and the model is too simple to understand the data. Techniques to reduce underfitting are:-

* Increase model complexity.
* Increase the number of features.
* Remove noise.
* Increase the training phase by increasing duration or epochs.

1. **Hyper parameter Tuning**:- A Machine Leaning model is defined as a mathematical model with a number of parameters that need to be learned from the data. By training a model with existing data, we are able to fit the model parameters. However there is another kind of parameters, known as hyper parameters, that cannot be directly learned from the regular training process. They are usually fixed before the actual training process begins. These parameters express important properties of the model such as its complexity or how fast it should learn. Models can have many hyperparameters and finding the best combination of parameters can be treated as a search problem.Two best strategies for Hyperparameter tuning are:-

* **GridSearchCV:**- In GridSearchCV approach, machine learning model is evaluated for a range of hyperparameter values. This approach is called GridSearchCV. This approach is called GridSearchCV because it searches for best set of hyperparameters from the grid of hyperparameters values.
* **RandomizedSearchCV:**- RandomizedSearchCV solves the drawbacks of GridsearchCV, as it goes through only a fixed number of hyperparameter settings. It moves within the grid in random fashion to find the best set of hyperparameters.

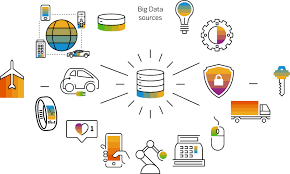
**UNIT - 2**

**CHAPTER 1:STEPS TO MACHINE LEARNING PROJECT**

1. **Data Collection**

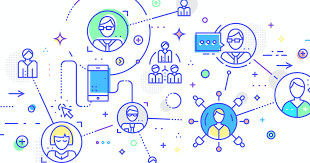
Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Following are some of the problems that can arise in data collection:

* **Inaccurate data:-** The collected data could be unrelated to the problem statement.
* **Missing data:-** Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.
* **Data imbalance:-** Some classes or categories in the data may have a disproportionately high or low number of corresponding samples. As a result, they risk being under-represented in the model.
* **Data bias:-** Depending on how the data, subjects and labels themselves are chosen, the model could propagate inherent biases on gender, politics, age or region, for example. Data bias is difficult to detect and remove.



Several techniques can be applied to address those problems:

* Pre-cleaned freely available datasets. If the problem statement (for example, image classification, object recognition) aligns with a clean, preexisting, properly formulated dataset, then take advantage of existing, open-source expertise.
* Web crawling and scraping. Automated tools, bots and headless browsers can crawl and scrape websites for data.
* Private data. ML engineers can create their own data. This is helpful when the amount of data required to train the model is small and the problem statement is too specific to generalize over an open-source dataset.

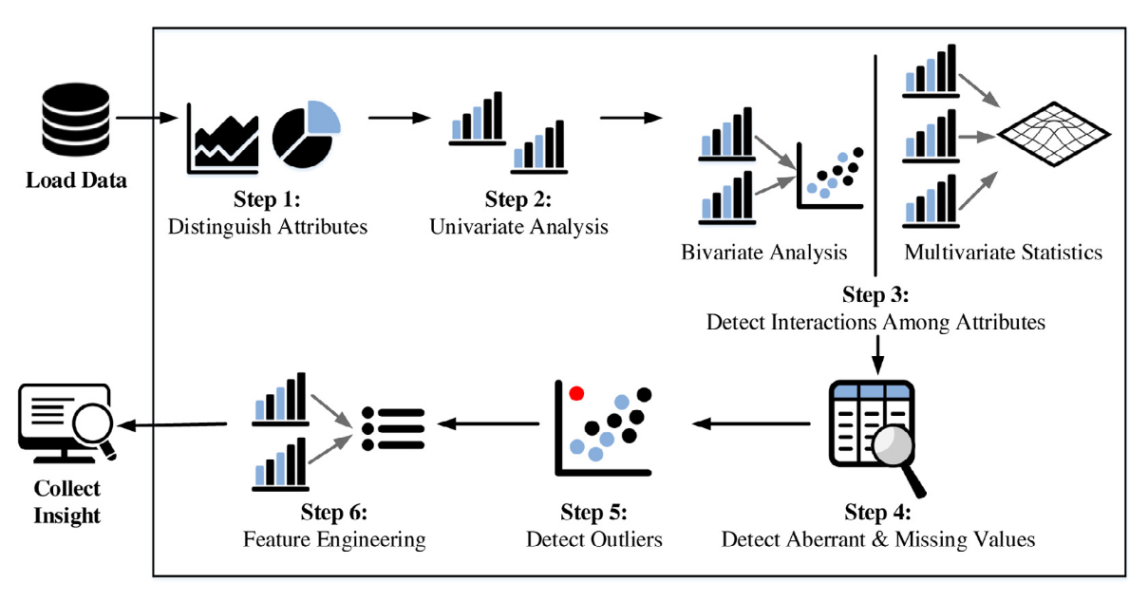


The data you have collected are of different type such as structured and unstructured. Within the structured data we have nominal, ordinal, numerical and time series.

Nominal are mutually exclusive data entries example color of vehicle, car sales etc whereas ordinal have a defined order or rating between them for example rank of student, qualification of an employee.

Numerical data are continuous values where difference between them matters, example house price of 10,000 is more then 5000. Finally time series contains a series of time representing a particular time interval, for example, 2013-2018.

1. **Exploratory Data Analysis**



**Exploratory data analysis** (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

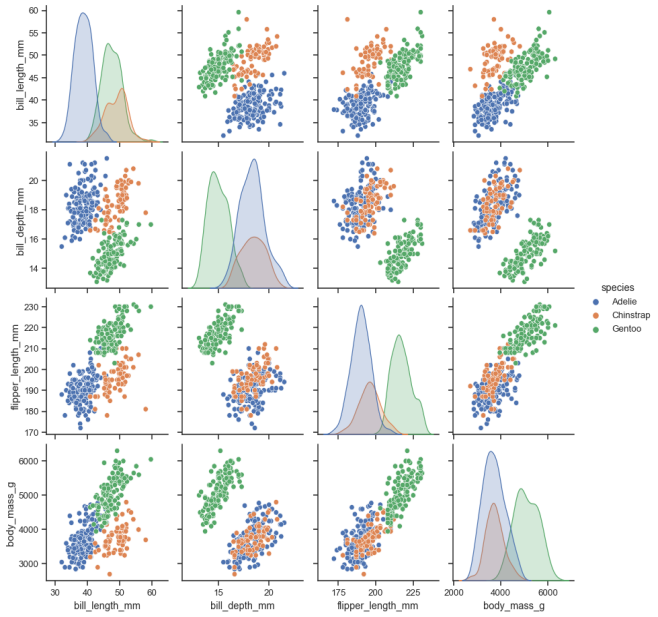
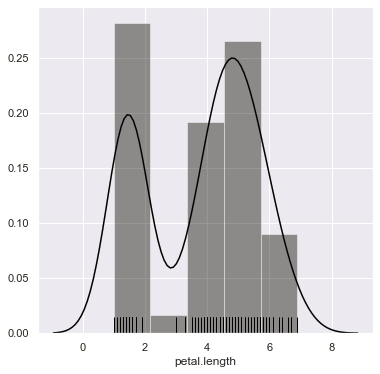
EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate.

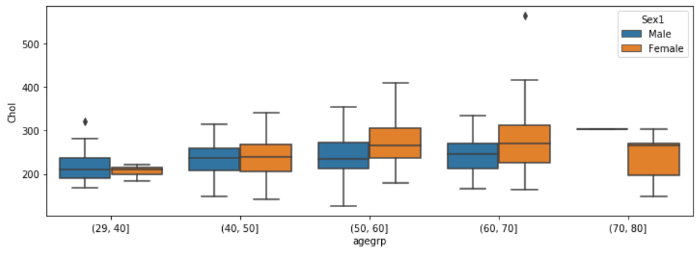
The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

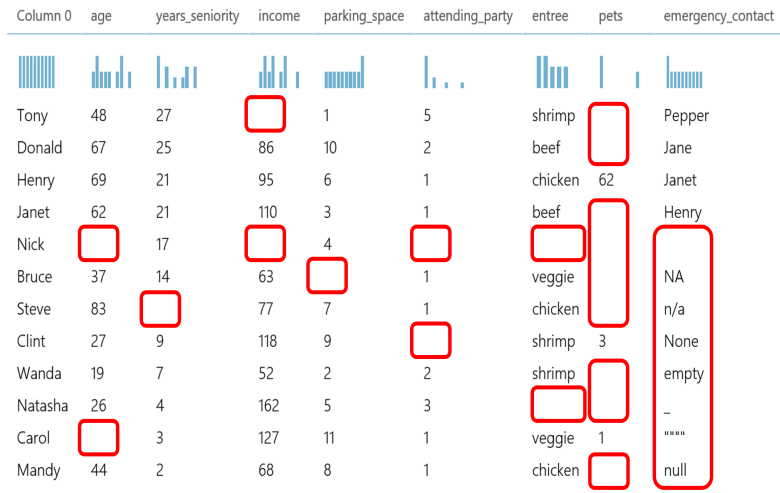
**Data Visualization** is the process of analyzing data in the form of graphs or maps, making it a lot easier to understand the trends or patterns in the data. There are various types of visualizations:-

* **Univariate Analysis**: This type of data consists of only one variable. The analysis of univariate data is thus the simplest form of analysis since the information deals with only one quantity that changes. It does not deal with causes or relationships and the main purpose of the analysis is to describe the data and find patterns that exists within it.
* **Bi-Variate Analysis**: this type of data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to find out the relationship among the two variables.
* **Multi-variate Analysis**: when the data involves three or more variables, it is categorized under multivariate.

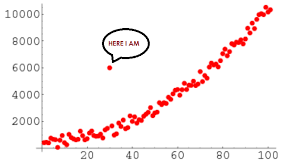




**Missing values**:-In real world data, there are some instances where a particular element is absent because of various reasons, such as, corrupt data, failure to load the information, or incomplete extraction. [Handling](https://analyticsindiamag.com/get-started-preparing-data-machine-learning/) the missing values is one of the greatest challenges faced by analysts, because making the right decision on how to handle it generates robust data models. Handling missing values is one of the major tasks in data preprocessing.



**Detecting outliers**:- An outlier a data-item/object that deviates significantly from the rest of the objects. They can be caused by measurement or execution errors. The analysis for outliers detection id referred outliers mining. There are many ways to detect the outliers, and the removal process is the data frame same as removing a data item from the pandas data-frame.



1. **Data Preprocessing**

Data Preprocessing is a step of data preparation, a data mining technique where raw data is cleaned and transformed into useful and efficient format.

1. **Handling missing values**:- Let us look at different ways of imputing the missing values:-

* **Delete the Rows**:- If your data has small amount of missing values there is no harm in deleting those rows while making sure that there is no addition of bias.
* **Replacing with Central Tendency**:- Applied to numeric kind of data where we impute the missing values to central tendency(mean, median and mode). It might add variance to data but its better then removing rows.
* **Assigning a unique category**:- We can assign an altogether new value to all the missing values of the column so that machine learning algorithm treats it as separate category.
* **Predicting the missing values**:- We can run a machine learning model to find the values of missing data by training it on available data. We can then impute these values to the missing data entries.
* **Using Algorithm which support missing values**:- Algorithms like KNN works well even with missing values as it depends on the distance between points.

1. **Handling Outliers**:- An Outlier is an observation in a given dataset that lies far from the rest of the observations. Outliers can be handled in various ways:-

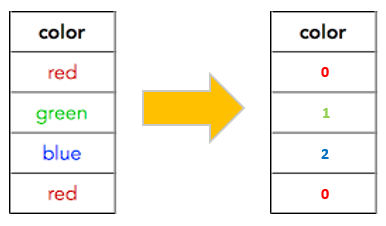
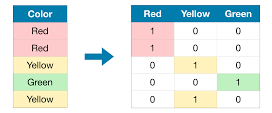
* **Trimming/Remove the outliers**:- If the sample is large enough then removing outliers don’t affect much. Although not a good practice the outliers can be removed to make data more representative.
* **Quantile based Flooring**:- In this method outliers values are capped to certain limit above the 90th percentile value or floored at a factor below the 10th percentile value.
* **Mean/Median imputation**:- We replace outliers value to the mean or median of the data. As mean is generally highly influenced by the outliers, it is advised to replace with the median value.

1. **Feature encoding**:- As models cannot handle categorical variables, we convert categorical variables into numerical variables by using feature encoding techniques.

We encode categorical data numerically because math is generally done using numbers. A big part of natural language processing is converting text to numbers.

Some of these techniques are:-

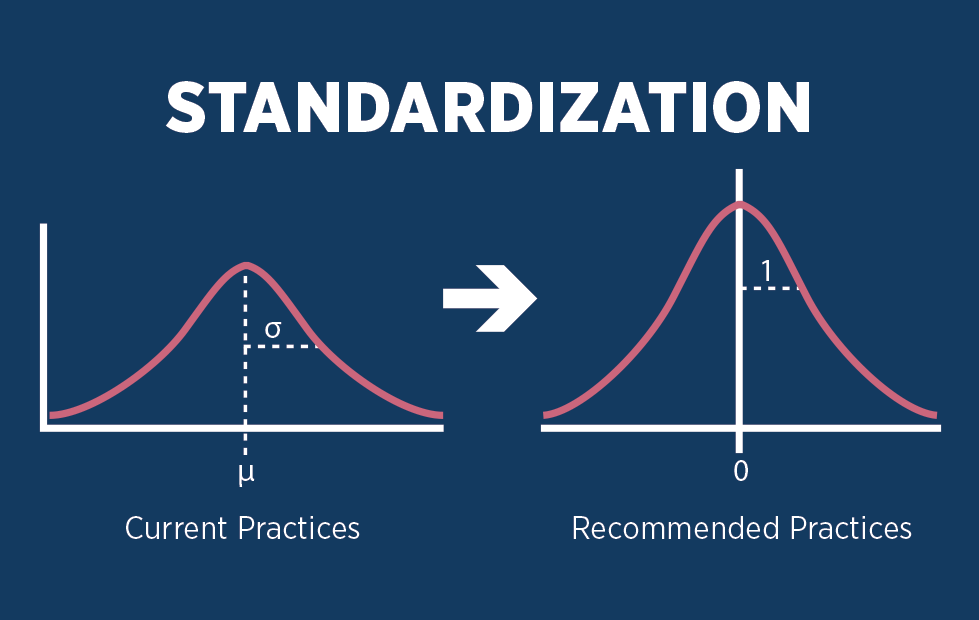
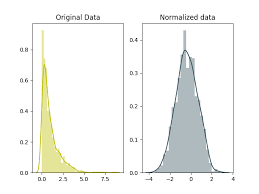
* **One-Hot Encoding**:- One-Hot encoding converts each type of category of a particular feature into separate features. In those new features, wherever that category is encountered is marked as 1 otherwise 0. This method helps machine learning model to learn better , but it also cost resources to compute all those new features.
* **Label Encoding and Ordinal Encoding**:- Label encoding doesn’t add any extra columns to the data but instead assigns a number to each unique value in a feature. We use label encoding when we handle ordinal categories or categories which contains some order. It creates a clean data set but model can learn from the order if your feature does not actually had an order. For example- difference between 2 and 3 is same as 4 and 5 so model might infer this from label encoding.



1. **Feature Scaling**:- Feature Scaling is a technique to standardize the independent features present in the data in a fixed range. It is performed to handle highly varying magnitudes or values or units. If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values.

Types of Feature scaling are:-

* ****Normalization** is a scaling technique in which values are shifted and re-scaled so that they end up ranging between 0 and 1. It is also known as Min-Max scaling.** Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution.
* ****Standardization** is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.** Standardization can be helpful in cases where the data follows a Gaussian distribution.



1. **Feature Engineering**:- Feature engineering is a machine learning technique that leverages data to create new variables that aren’t in the training set. It can produce new features for both supervised and unsupervised learning, with the goal of **simplifying and speeding up data transformations** while also **enhancing model accuracy**. Feature engineering in ML has four processes:-

* ****Feature Creation****: Feature creation is finding the most useful variables to be used in a predictive model.
* ****Transformations****: The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model.
* ****Feature Extraction****: Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling.
* ****Feature Selection:**** While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant.

1. **Imbalanced Dataset**:- Imbalanced data refers to those types of data sets where the target class has an uneven distribution of observations.

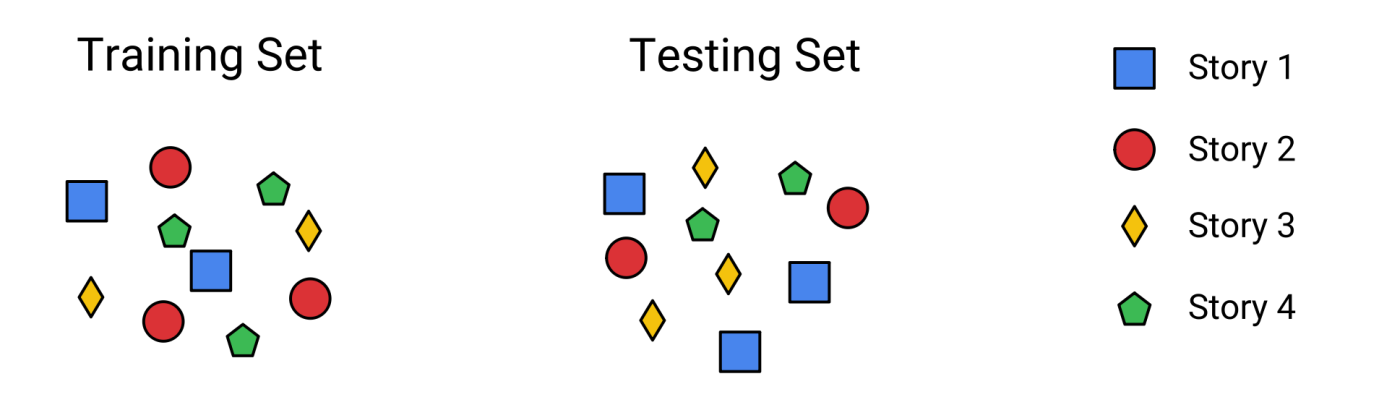
It means that one class label has very high number of observations while other has a very low number of observations. Most common datasets where such imbalance occur are Disease diagnosis, Customer loan prediction, Fraud detection and Natural disaster. Different techniques work well with different problems, so of them are:-

* **Choose proper evaluation metric**:- **The accuracy**of a classifier is the total number of correct predictions by the classifier divided by the total number of predictions. This may be good enough for a well-balanced class but not ideal for the imbalanced class problem. The other metrics such as **precision**is the measure of how accurate the classifier’s prediction of a specific class and **recall**is the measure of the classifier’s ability to identify a class.
* **Resampling (Oversampling and Undersampling):**- This technique is used to upsample or downsample the minority or majority class. When we are using an imbalanced dataset, we can oversample the minority class using replacement. This technique is called oversampling. Similarly, we can randomly delete rows from the majority class to match them with the minority class which is called undersampling. After sampling the data we can get a balanced dataset for both majority and minority classes. So, when both classes have a similar number of records present in the dataset, we can assume that the classifier will give equal importance to both classes.
* **SMOTE**:- **Synthetic Minority Oversampling Technique** or **SMOTE**is another technique to oversample the minority class. Simply adding duplicate records of minority class often don’t add any new information to the model. In SMOTE new instances are synthesized from the existing data. If we explain it in simple words, SMOTE looks into minority class instances and use *k* nearest neighbor to select a random nearest neighbor, and a synthetic instance is created randomly in feature space.

1. **Data splitting**

One of the first decisions to make when starting a modeling project is how to utilize the existing data. One common technique is to split the data into two groups typically referred to as the training and testing sets.

The training set is used to develop models and feature sets; they are the substrate for estimating parameters, comparing models, and all of the other activities required to reach a final model. The test set is used only at the conclusion of these activities for estimating a final, unbiased assessment of the model’s performance. It is critical that the test set not be used prior to this point. Looking at the test sets results would bias the outcomes since the testing data will have become part of the model development process.



**UNIT-3**

**CHAPTER 1: MACHINE LEARNING LIBRARIES**

1. **NumPy**

Numpy is general purpose array processing package. It provides a high performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with python. It is an open-source software. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated functions
* Tools for integrating C/C++ and FORTRAN code
* Useful linear algebra, Fourier Transform, and random number capabilities.

Numpy’s main object is homogeneous multidimensional array, where dimensions are called axes. Numpy array class is called ndarray. Numpy is used in machine learning to get fast results when arrays are involved as python array(lists) are generally slow.

1. **Pandas**

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis/manipulation tool available in language. Some features of pandas are:-

* Easy handling of missing values
* Size mutability
* Powerful, flexible groupby functionality to perform split-apply-combine operation
* Intelligent label-based slicing, fancy indexing and sub-setting
* Flexible reshaping and pivoting of data sets.
* Intuitive merging and joining data sets.
* Make it easy to convert ragged, differently-indexed data in other Python and Numpy data structures into DataFrame objects.

1. **Matplotlib**

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. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

* Create [publication quality plots](https://ieeexplore.ieee.org/document/4160265/citations?tabFilter=papers).
* Make [interactive figures](https://mybinder.org/v2/gh/matplotlib/mpl-brochure-binder/main?labpath=MatplotlibExample.ipynb) that can zoom, pan, update.
* Customize [visual style](https://matplotlib.org/stable/gallery/style_sheets/style_sheets_reference.html) and [layout](https://matplotlib.org/stable/tutorials/provisional/mosaic.html).
* Export to [many file formats](https://matplotlib.org/stable/api/figure_api.html" \l "matplotlib.figure.Figure.savefig) .
* Embed in [JupyterLab and Graphical User Interfaces](https://matplotlib.org/stable/gallery/" \l "embedding-matplotlib-in-graphical-user-interfaces).
* Use a rich array of [third-party packages](https://matplotlib.org/mpl-third-party/) built on Matplotlib.

1. **Seaborn**

Seaborn is a Python data visualization library based on [matplotlib](https://matplotlib.org/). It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

Plots are basically used for visualizing the relationship between variables. Those variables can be either be completely numerical or a category like a group, class or division. Seaborn divides plot into below categories-

* Relational plots: This plot is used to understand the relation between two variables.
* Categorical plots: This plot is used for examining univariate and bivariate distributions.
* Regression plots: They are primarily intended to add a visual guide that helps to emphasize patters in a dataset during exploratory data analysis.
* Multi-plot grids: It is an useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

1. **Scikit-learn**

Scikit-learn is an open source Python library that implements a range of machine learning, preprocessing, cross-validation, and visualization algorithms using a unified interface. It is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency. It has minimal dependencies and is distributed under the simplified BSD license, encouraging its use in both academic and commercial settings.

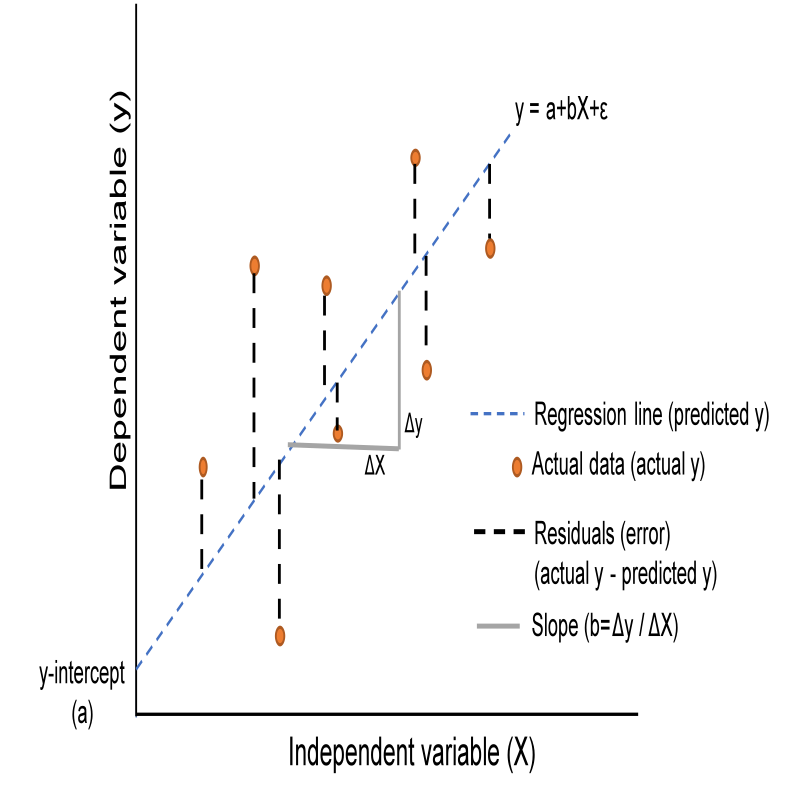
Important features of scikit-learn:

* Simple and efficient told for data mining and data analysis. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means etc.
* Accessible to everybody and reusable in various contexts.
* Built on the top of Numpy, SciPy, and matplotlib.
* Open source, commercially usable.

**CHAPTER 2: SUPERVISED MACHINE LEARNING MODELS**

1. **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 algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as 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.

Mathematically, we can represent a linear regression as:

y= a0+a1x+ ε

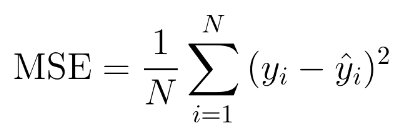
Y= Dependent Variable (Target Variable)  
X= Independent Variable (predictor Variable)  
a0= intercept of the line (Gives an additional degree of freedom)  
a1 = Linear regression coefficient (scale factor to each input value).  
ε = random error

**Finding the best fit line:**- When working with linear regression, our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error. The different values for weights or the coefficient of lines (a0, a1) gives a different line of regression, so we need to calculate the best values for a0 and a1 to find the best fit line, so to calculate this we use cost function.

**Cost function**:-

* The different values for weights or coefficient of lines (a0, a1) gives the different line of regression, and the cost function is used to estimate the values of the coefficient for the best fit line.
* Cost function optimizes the regression coefficients or weights. It measures how a linear regression model is performing.
* We can use the cost function to find the accuracy of the **mapping function**, which maps the input variable to the output variable. This mapping function is also known as **Hypothesis function**.

For Linear Regression, we use the ****Mean Squared Error (MSE)**** cost function, which is the average of squared error occurred between the predicted values and actual values. It can be written as:



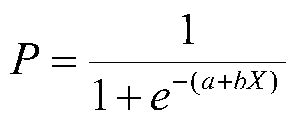
**Gradient descent** is used to minimize the MSE by calculating the gradient of the cost function. A regression model uses gradient descent to update the coefficients of the line by reducing the cost function. It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.

1. **LOGISTIC REGRESSION**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. 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 the 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 function:

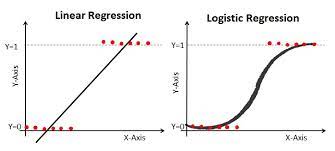


Here, P is the probability of 1, e is the base of natural logarithm and a and b are parameters of the model.

* The sigmoid function is a mathematical function used to map the predicted values to probabilities.
* It maps any real value into another value within a range of 0 and 1.
* The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form.
* In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

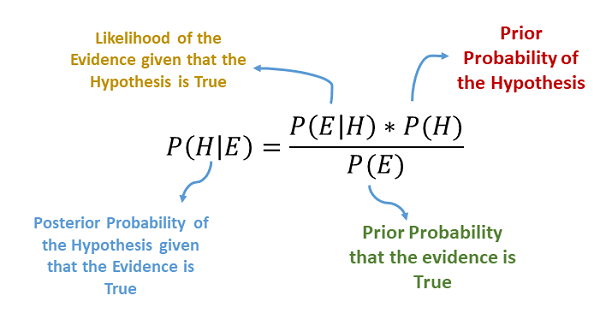
**Loss Function**: A loss function is a measure of fit between a mathematical model of data and the actual data. We choose the parameters of our model to minimize the badness-of-fit or to maximize the goodness-of-fit of the model to the data. With linear or curvilinear models, there is a mathematical solution to the problem that will minimize the sum of squares, that is,

**b = (X'X)-1X'y**



1. **NAIVE BAYES CLASSIFIER**

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, where 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 bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. It is called Bayes because it depends on the principle of [Bayes' Theorem](https://www.javatpoint.com/bayes-theorem-in-artifical-intelligence" \t "https://www.javatpoint.com/_blank).



**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

IMG_256

**Where,**

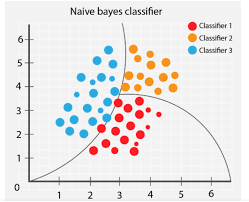
**P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.

**P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.

**P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.

**P(B) is Marginal Probability**: Probability of Evidence.

Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.



### Advantages of Naïve Bayes Classifier:

* Naïve Bayes is one of the fast and easy ML algorithms to predict a class of datasets.
* It can be used for Binary as well as Multi-class Classifications.
* It performs well in Multi-class predictions as compared to the other Algorithms.
* It is the most popular choice for **text classification problems**.

### Disadvantages of Naïve Bayes Classifier:

* Naive Bayes assumes that all features are independent or unrelated, so it cannot learn the relationship between features.

1. **K-NEAREST NEIGHBOR**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories and algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.



## **How to select the value of K in the K-NN Algorithm?**

Below are some points to remember while selecting the value of K in the K-NN algorithm:

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

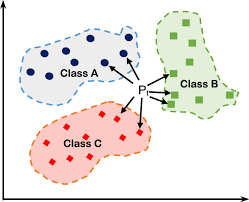
## Advantages of KNN Algorithm:

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

## Disadvantages of KNN Algorithm:

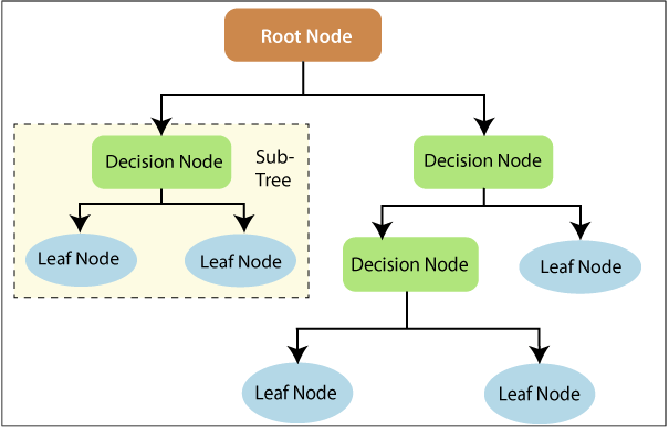
* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5. A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.



1. **DECISION TREE**

Decision Tree is a **Supervised learning technique**that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where**internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.** In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into sub-trees.



Decision Trees are preferred in classification problems as:-

* Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
* The logic behind the decision tree can be easily understood because it shows a tree-like structure.

## Decision Tree Terminologies

* ****Root Node**:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
* ****Leaf Node**:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
* ****Splitting**:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
* ****Branch/Sub Tree**:** A tree formed by splitting the tree.
* ****Pruning**:** Pruning is the process of removing the unwanted branches from the tree.
* ****Parent/Child node**:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree.

## Advantages of the Decision Tree

* It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
* It can be very useful for solving decision-related problems.
* It helps to think about all the possible outcomes for a problem.
* There is less requirement of data cleaning compared to other algorithms.

## Disadvantages of the Decision Tree

* The decision tree contains lots of layers, which makes it complex.
* It may have an overfitting issue, which can be resolved using the **Random Forest algorithm.**
* For more class labels, the computational complexity of the decision tree may increase.

1. **RANDOM FOREST CLASSIFIER**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

**Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**



Random Forest is preferred over other classification algorithms as:-

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.
* It is used in various sectors like Banking, Medicine, Marketing and Land Use.

## Advantages of Random Forest

* Random Forest is capable of performing both Classification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

## Disadvantages of Random Forest

* Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

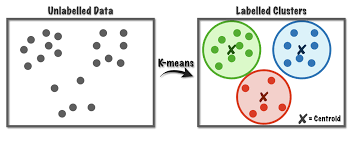


Random decision forests correct for decision trees' habit of [overfitting](https://en.wikipedia.org/wiki/Overfitting" \o "Overfitting) to their [training set](https://en.wikipedia.org/wiki/Test_set" \o "Test set).  Random forests generally outperform [decision trees](https://en.wikipedia.org/wiki/Decision_tree_learning" \o "Decision tree learning), but their accuracy is lower than gradient boosted. However, data characteristics can affect their performance.

**CHAPTER 3: UNSUPERVISED MACHINE LERNING MODELS**

1. **K-MEANS**
2. Means Clustering is an [Unsupervised Learning algorithm](https://www.javatpoint.com/unsupervised-machine-learning), which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on. It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.



The k-means [clustering](https://www.javatpoint.com/clustering-in-machine-learning) algorithm mainly performs two tasks:

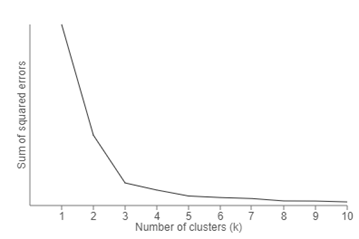
* Determines the best value for K center points or centroids by an iterative process.
* Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

### **Elbow Method**

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2



To find the optimal value of clusters, the elbow method follows the below steps:

* It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
* For each value of K, calculates the WCSS value.
* Plots a curve between calculated WCSS values and the number of clusters K.
* The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

1. **HIERARCHICAL CLUSTERING**

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as **hierarchical cluster analysis** or HCA.

The hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the **dendrogram**.

In the K-means clustering that there are some challenges with this algorithm, which are a predetermined number of clusters, and it always tries to create the clusters of the same size. To solve these two challenges, we can opt for the hierarchical clustering algorithm because, in this algorithm, we don't need to have knowledge about the predefined number of clusters. In this technique, you can decide the optimal number of clusters by noticing which vertical lines can be cut by horizontal line without intersecting a cluster and covers the maximum distance.

