SIX WEEKS SUMMER TRAINING REPORT

On

MACHINE LEARNING

Submitted by

Name: Mamidipaka Venkata Sai Nikhil

Registration No: 12013998

Program Name: B. TECH (CSE)

Under the Guidance of

Prince Rajkumar

School of Computer Science & Engineering

Lovely Professional University, Phagwara

(May-July, 2022)

DECLARATION

I hereby declare that I have completed my six weeks summer training at **Techvanto Academy** online platform from **25th May, 2022** to **10th July, 2022** under the guidance of **Prince Rajkumar Sir.** I have declared that I have worked with full dedication during these six weeks of training and my learning outcomes fulfil the requirements of training for the award of degree of **Bachelor of Technology (B.Tech.) in CSE - Data Science (ML & AI)**, Lovely Professional University, Phagwara.

Mamidipaka Venkata Sai Nikhil

12013998

Date: 10th July, 2022

ACKNOWLEDGEMENT

The success and final outcome of learning Machine Learning required a lot of guidance and assistance from many people. I am extremely privileged to have got this all along the completion of my course and few of the projects. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

I would like to express my sincere gratitude to Techvanto Academy, for providing me an opportunity to do the course and project work and giving me all support and guidance, which made me complete the course duty. I would especially thankful to the course advisor Mr. Rajkumar sir and Mr. Mubarak sir.

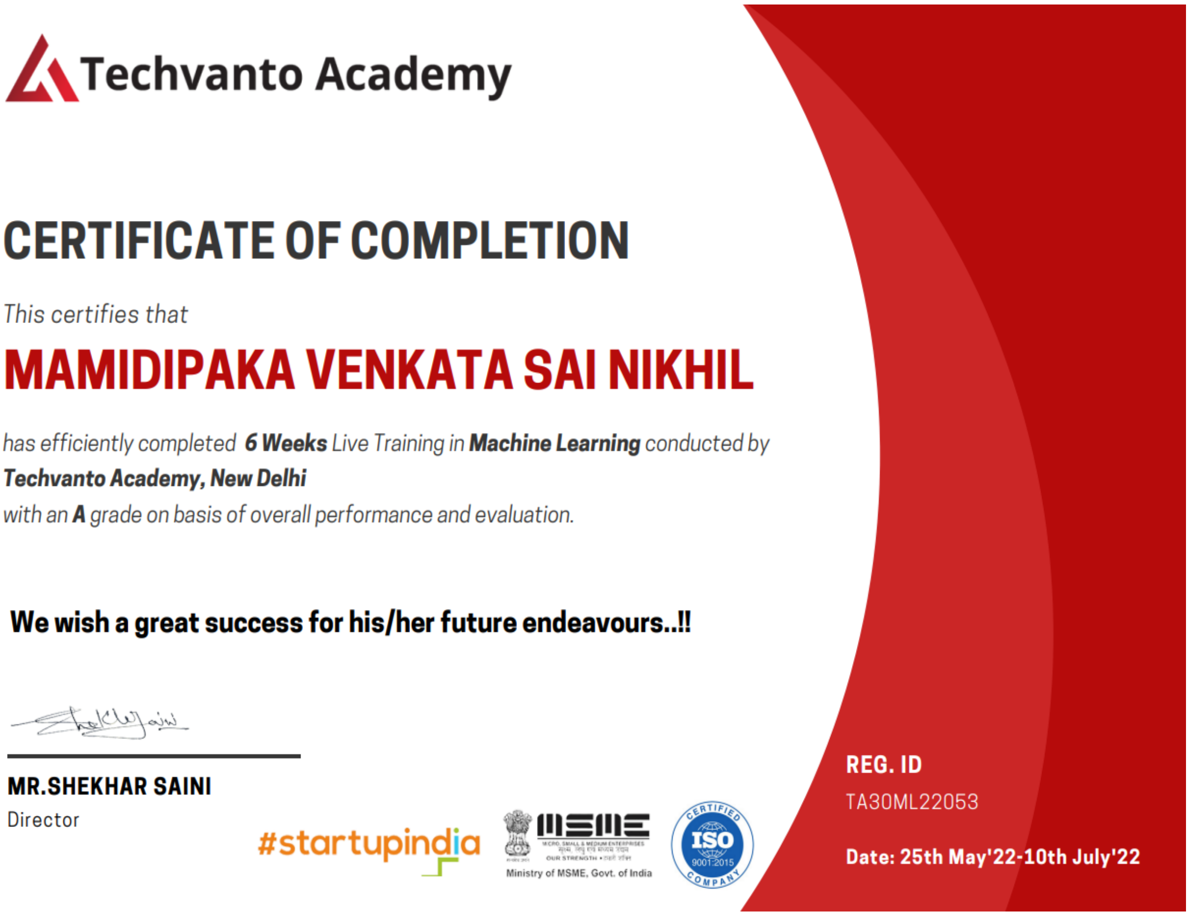
I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance for choosing this course.

Mamidipaka Venkata Sai Nikhil

12013998

Date: 10th July, 2022

Training Certificate



Techvanto Academy

CERTIFICATE OF COMPLETION

This certifies that

MAMIDIPAKA VENKATA SAI NIKHIL

*has efficiently completed **6 Weeks** Live Training in **Machine Learning** conducted by*
Techvanto Academy, New Delhi
*with an **A** grade on basis of overall performance and evaluation.*

We wish a great success for his/her future endeavours..!!

MR.SHEKHAR SAINI
Director

#startupindia

MSME
Ministry of MSME, Govt. of India

CERTIFIED ISO 9001:2015 COMPANY

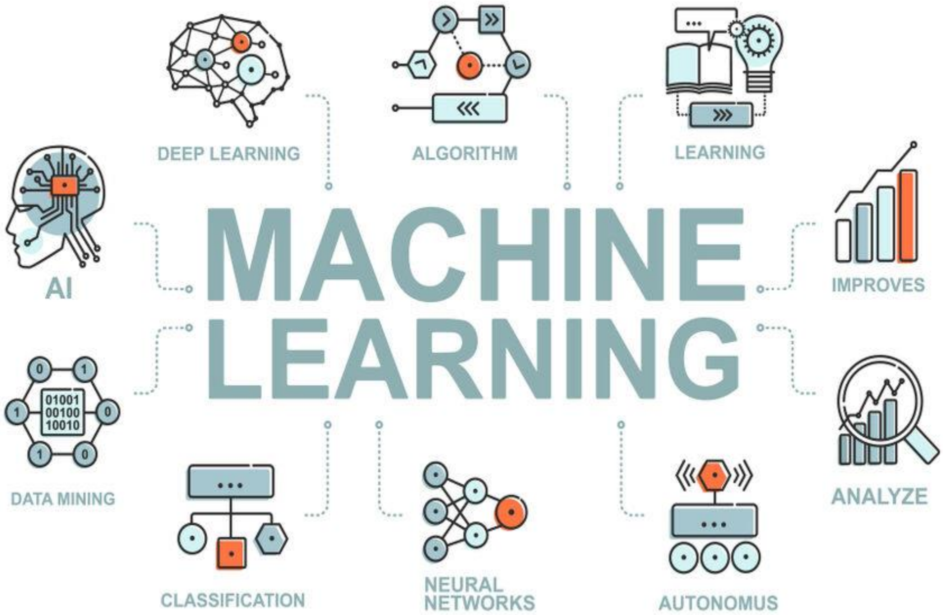
REG. ID
TA30ML22053

Date: 25th May'22-10th July'22

TABLE OF CONTENTS

1. Introduction

Arthur Samuel, an American Pioneer in the field of computer gaming and artificial intelligence coined the term “Machine Learning” in 1959. He defined Machine Learning as a “Field of study that gives computers the capability to learn without being explicitly programmed”. Over the past two decades Machine Learning has become one of the mainstays of information technology. With the ever-increasing amounts of data becoming available there is good reason to believe that smart data analysis will become even more pervasive as a necessary ingredient for technological progress.



Machine learning can be explained as automating and improving the type of learning process of computers based on their experiences without being actually programmed i.e., without any human assistance. Machine Learning is a modern innovation that has enhanced many industrial and professional processes as well as our daily lives. It is a subset of Artificial Intelligence (AI), which focuses on using statistical techniques to build intelligent computer systems to learn from available databases. Machine Learning algorithms build a model based on sample data known as training data in order to make predictions or decisions without being explicitly programmed to do so.

Relation to Data Mining:

Data Mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as “unsupervised learning” or as a pre-processing step to improve learner accuracy.

Relation to Optimization:

Optimization is one of the core components of machine learning. The essence of most machine learning algorithms is to build an optimization model and learn the parameters in the objective function from the given data.

- Function optimization is the reason why we minimize error, cost, or loss when fitting a machine learning algorithm.
- Optimization is also performed during data preparation, hyperparameter tuning, model selection in a predictive modelling project.

Relation to Statistics:

Statistics is a core component of data analytics and machine learning. It helps us analyse and visualize data to find unseen patterns.

Use of statistics in Machine Learning:

- Cleaning and pre-processing the data.
- Asking questions about the data
- Selecting the right features
- Model evaluation
- Model prediction

Future of Machine Learning:

Machine Learning is so versatile and powerful that it’s one of the most exciting technologies of our times. The future of machine learning is exceptionally exciting. At present, almost every common domain is powered by machine learning applications. To name a few such industries- healthcare, search engine, digital marketing, and education are the major beneficiaries.

The discipline of machine learning has the potential to be transformed and innovated by quantum algorithms. Machine learning with Quantum can improve the analysis of data and get more profound insights.

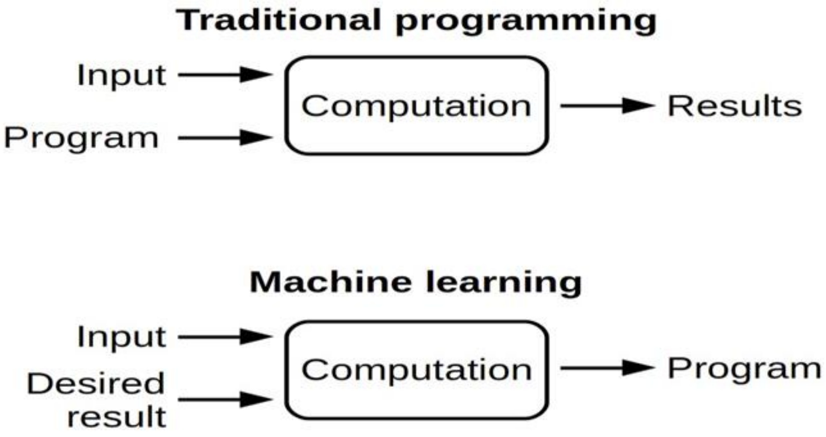
2.Technology Learnt

- **Define Machine Learning**

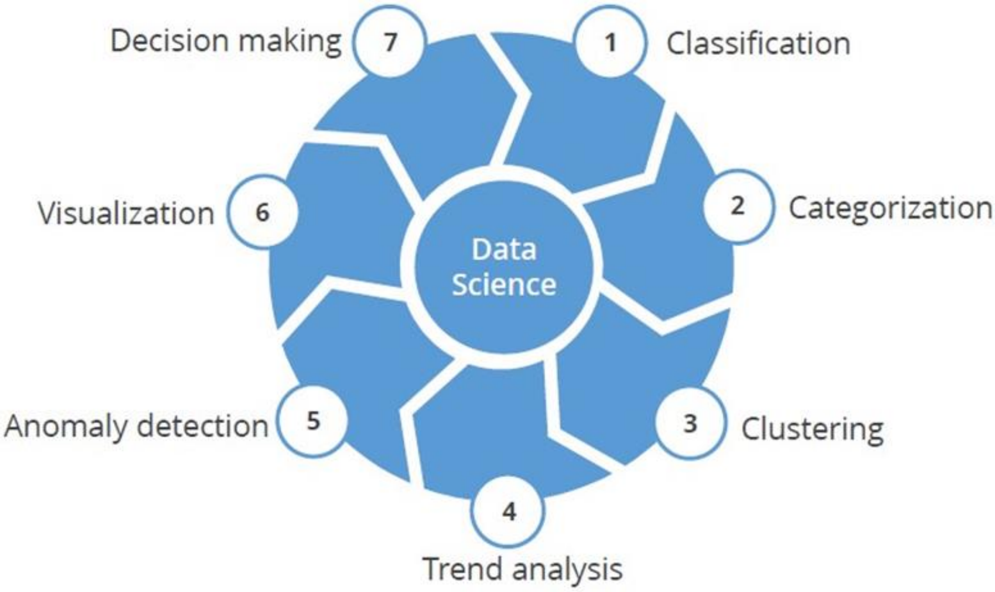
  Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

- **Features of Machine Learning**
  - ✓ Machine Learning is computing-intensive and generally requires a large amount of training data.
  - ✓ It involves repetitive training to improve the learning and decision making of algorithms.
  - ✓ As more data gets added, Machine Learning training can be automated for learning new data patterns and adapting its algorithm.

- **Traditional Programming vs Machine Learning Approach**



- **Machine Learning Approach**



Machine Learning can learn from labelled data (known as supervised learning) or unlabelled data (known as unsupervised learning).

- **Types of Machine Learning**

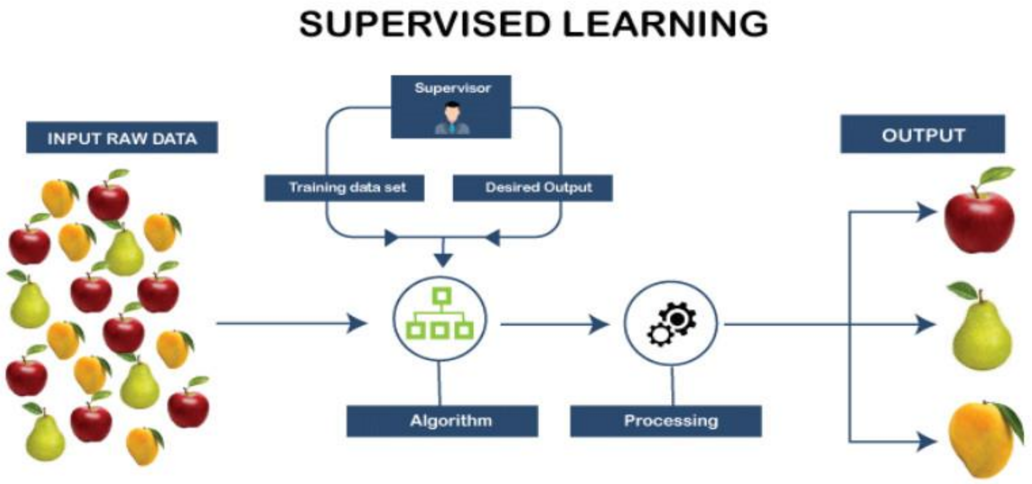
  Machine Learning is classified mainly into four types:

  1) Supervised Learning
  2) Unsupervised Learning
  3) Semi-Supervised Learning
  4) Reinforcement Learning

1. Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The accurate prediction of test data requires large data to have a sufficient understanding of the patterns.

The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y).



Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering etc.

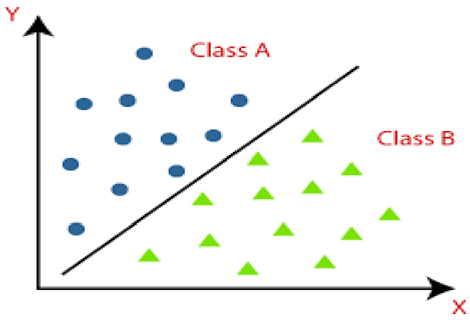
Supervised machine learning can be classified into two types of problems. They are:

a. Classification
b. Regression

a) Classification

Classification is a process of categorizing a given set of data into classes. It can be performed on both structured or unstructured data. The process starts with predicting the class of given data points.

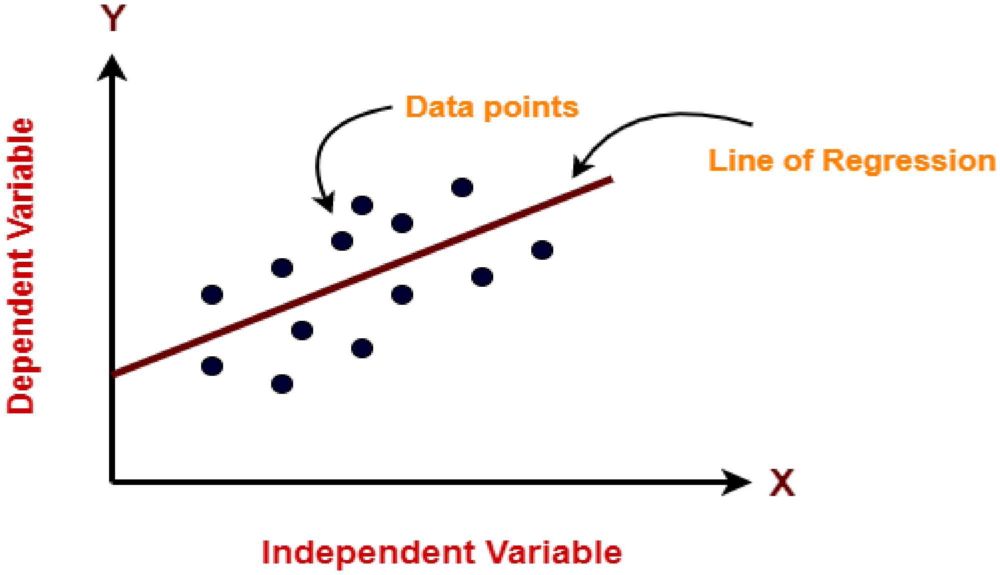
Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as Yes or No, Male or Female, Red or Blue etc. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are spam Detection, Email filtering, etc.



Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbour and random forest.

b) Regression

Regression analysis is a statistical method to model the relationship between a dependent and independent variable with one or more independent variables. It is mainly used for prediction, forecasting, time series modelling and determining the casual-effect relationship between variables. Machine learning regression generally involves plotting a line of best fit through the data points. The distance between each point and the line is minimised to achieve the best fit line.

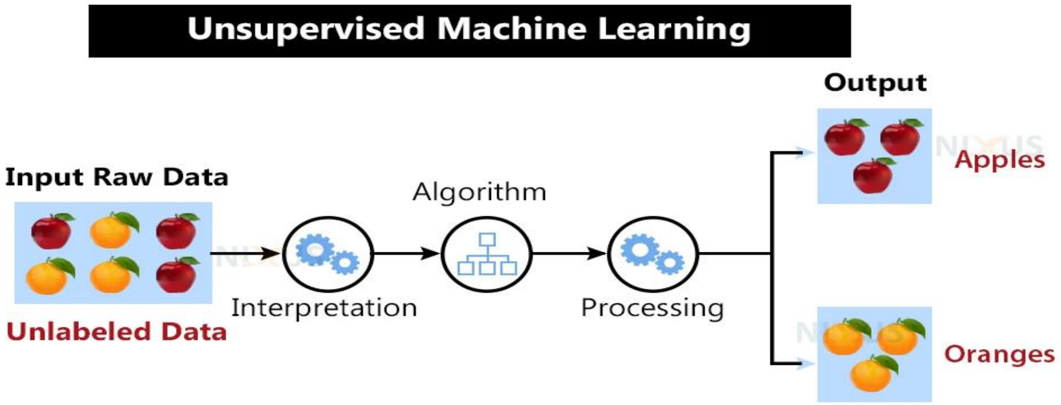


Some common regression algorithms are linear regression, support vector regression, decision tree, lasso regression, random forest regressor etc.

2. Unsupervised Learning

Unsupervised learning is a type of machine learning in which models are trained using unlabelled dataset and are allowed to act on that data without any supervision. Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data.

The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities and represent that dataset in a compressed format.

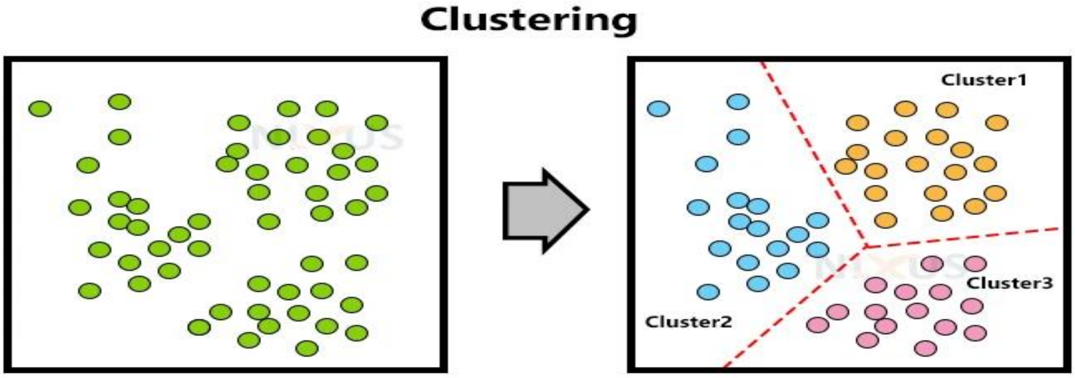


The unsupervised learning can be further categorized into two types of problems. They are:

a) Clustering
b) Association

a) Clustering

Clustering is a method of grouping the objects into clusters such that objects with most similarities remains into a group and has less or no similarities with the objects of another group. Cluster analysis finds the commonalities between the data objects and categories them as per the presence and absence of those commonalities.



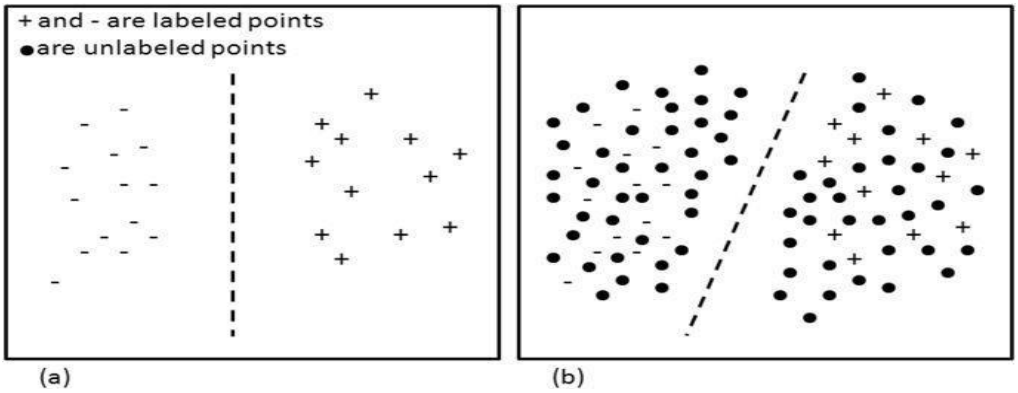
Clustering Anomaly detection can help you find out if there are any unexpected data points in your collection. It's important for detecting shady trades.

b) Association

An association rule is an unsupervised learning method which is used for finding the relationships between variables in the large database. It determines the set of items that occurs together in the dataset. Association rule makes marketing strategy more effective.

3. Semi-Supervised Learning

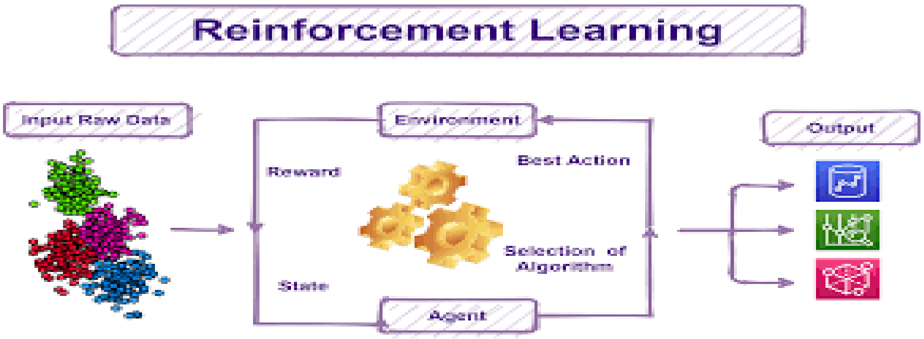
Semi-Supervised learning is a class of machine learning tasks and techniques that also make use of unlabelled data for training- typically a small amount of labelled data with a large amount of unlabelled data.



Semi-Supervised learning falls between Supervised learning (with completely labelled training data) and Unsupervised learning (without any labelled training data).

4. Reinforcement Learning

Reinforcement learning is a type of Machine Learning that allows the learning system to observe the environment and learn the ideal behaviour based on trying to maximize some notion of cumulative reward.



In reinforcement learning, there is no labelled data like supervised learning and agents learn from their experiences only. Reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, multi-agent systems.

3. Technology Implemented

Why Python Is a Perfect Language for Machine Learning?

1. A great library ecosystem-

A great choice of libraries is one of the main reasons Python is the most popular programming language used for AI. A library is a module or a group of modules published by different sources which include a pre-written piece of code that allows users to reach some functionality or perform different actions. Python libraries provide base level items so developers don’t have to code them from the very beginning every time. ML requires continuous data processing, and Python’s libraries let us access, handle and transform data. These are some of the most widespread libraries you can use for ML and AI:

- ✓ Scikit-learn for handling basic ML algorithms like clustering, linear and logistic regressions, regression, classification, and others.

- ✓ Pandas for high-level data structures and analysis. It allows merging and filtering of data, as well as gathering it from other external sources like Excel, for instance.

- ✓ Keras for deep learning. It allows fast calculations and prototyping, as it uses the GPU in addition to the CPU of the computer.

- ✓ TensorFlow for working with deep learning by setting up, training, and utilizing artificial neural networks with massive datasets.

- ✓ Matplotlib for creating 2D plots, histograms, charts, and other forms of visualization.

- ✓ NLTK for working with computational linguistics, natural language recognition, and processing.

- ✓ Scikit-image for image processing.

- ✓ PyBrain for neural networks, unsupervised and reinforcement learning.

- ✓ Stats Models for statistical algorithms and data exploration.

2. A low entry barrier –

Working in the ML and AI industry means dealing with a bunch of data that we need to process in the most convenient and effective way. The low entry barrier allows more data scientists to quickly pick up Python and start using it for AI development without wasting too much effort into learning the language

3. Flexibility –

Python for machine learning is a great choice, as this language is very flexible:

- ✓ It offers an option to choose either to use OOPs or scripting.
- ✓ There's also no need to recompile the source code, developers can implement any changes and quickly see the results.
- ✓ Programmers can combine Python and other languages to reach their goals.

4. Good Visualization Options –

For AI developers, it's important to highlight that in artificial intelligence, deep learning, and machine learning, it's vital to be able to represent data in a human-readable format. Libraries like Matplotlib allow data scientists to build charts, histograms, and plots for better data comprehension, effective presentation, and visualization. Different application programming interfaces also simplify the visualization process and make it easier to create clear reports.

5. Community Support –

It's always very helpful when there's strong community support built around the programming language. Python is an open-source language which means that there's a bunch of resources open for programmers starting from beginners and ending with pros.

6. Growing Popularity –

As a result of the advantages discussed above, Python is becoming more and more popular among data scientists. According to Stack Overflow, the popularity of Python is predicted to grow until 2020, at least.

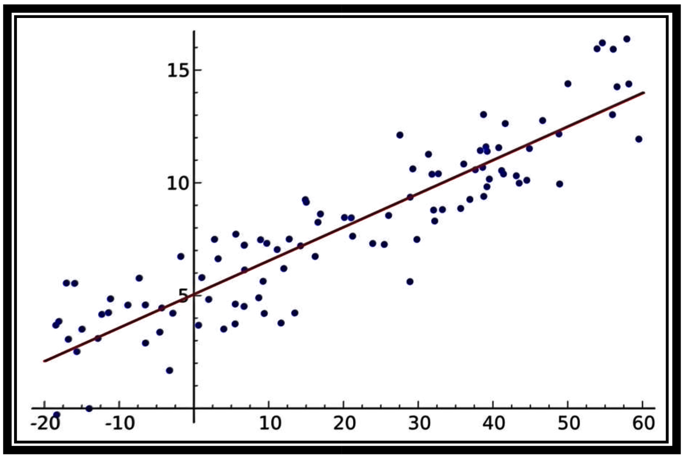
4. Machine Learning Algorithms

There are many types of Machine Learning Algorithms specific to different use cases. As we work with datasets, a machine learning algorithm works in two stages. We usually split the data around 20%-80% between testing and training stages. Under supervised learning, we split a dataset into a training data and test data in ML.

1. Linear Regression-

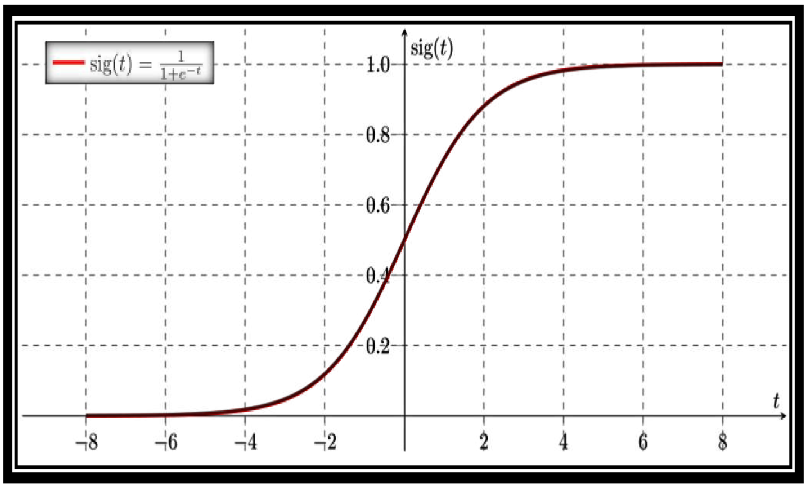
Linear regression is one of the supervised Machine learning algorithms in Python that observes continuous features and predicts an outcome. Depending on whether it runs on a single variable or on many features, we can call it simple linear regression or multiple linear regression.

This is one of the most popular Python ML algorithms and often under-appreciated. It assigns optimal weights to variables to create a line ax+b to predict the output. We often use linear regression to estimate real values like a number of calls and costs of houses based on continuous variables. The regression line is the best line that fits Y=a*X+b to denote a relationship between independent and dependent variables.



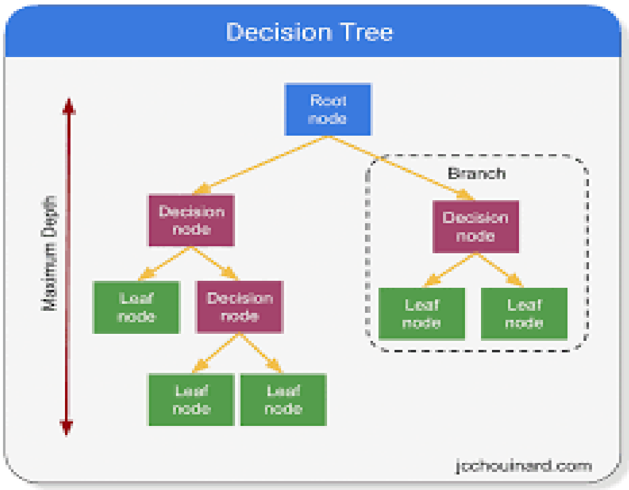
2. Logistic Regression-

Logistic regression is a supervised classification is unique Machine Learning algorithms in Python that finds its use in estimating discrete values like 0/1, yes/no, and true/false. This is based on a given set of independent variables. We use a logistic function to predict the probability of an event and this gives us an output between 0 and 1. Although it says ‘regression’, this is actually a classification algorithm. Logistic regression fits data into a logit function and is also called logit regression.



3. Decision Tree -

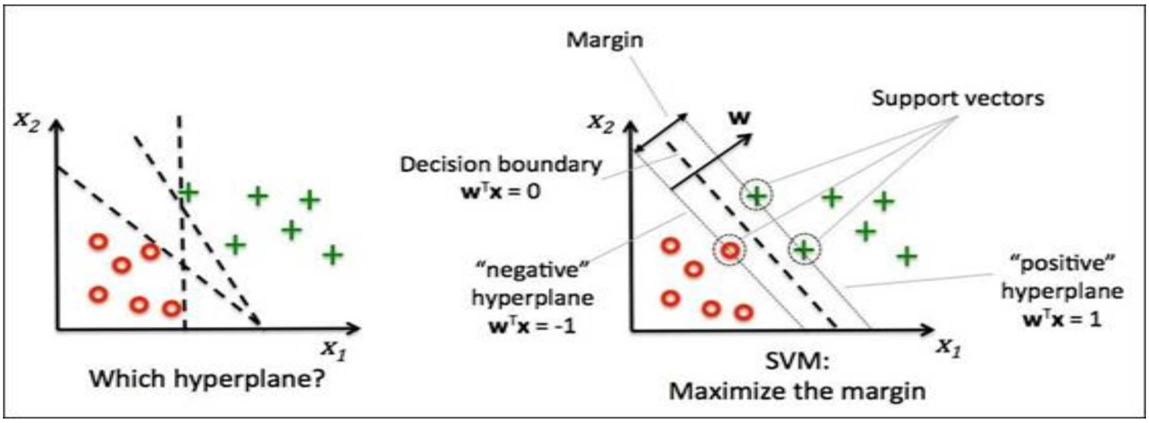
A decision tree falls under supervised Machine Learning Algorithms in Python and comes of use for both classification and regression- although mostly for classification. This model takes an instance, traverses the tree, and compares important features with a determined conditional statement.



Decision Tree, a Machine Learning algorithm in Python can work on both categorical and continuous dependent variables. Here, we split a population into two or more homogeneous sets. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structure, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees.

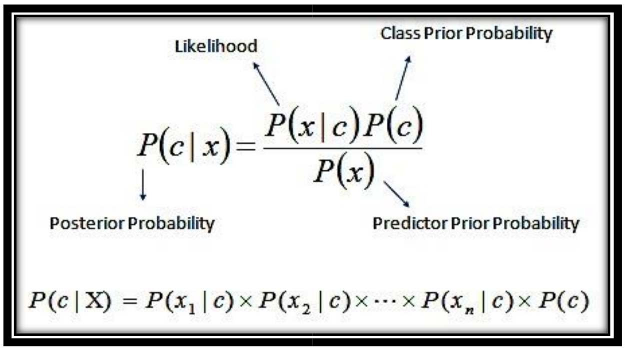
4. Support Vector Machine (SVM) –

SVM is a supervised classification is one of the most important Machines Learning algorithms in Python, that plots a line that divides different categories of your data. In this ML algorithm, we calculate the vector to optimize the line. This is to ensure that the closest point in each group lies farthest from each other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. When data are unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of data to groups, and then map new data to these formed groups.



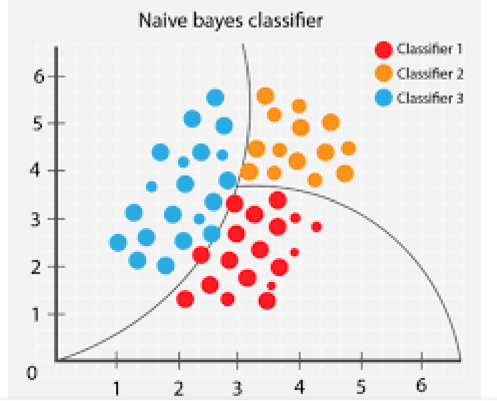
5. Naïve Bayes Algorithm –

Naive Bayes is a classification method which is based on Bayes' theorem. This assumes independence between predictors. A Naive Bayes classifier will assume that a feature in a class is unrelated to any other. Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed form expression which takes linear-time rather than by expensive iterative approximation as used for many other types of classifiers.



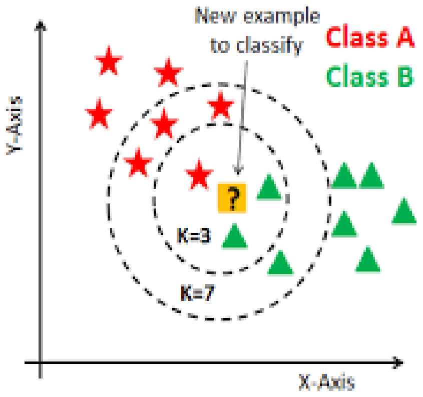
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$



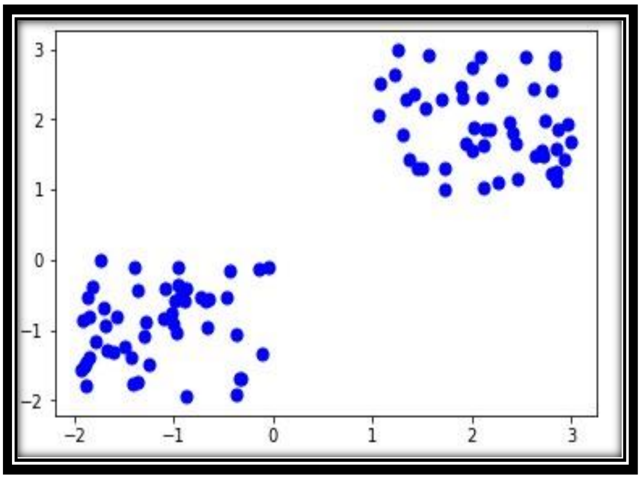
6. K-Nearest Neighbour (KNN) Algorithm-

This is a Python Machine Learning algorithm for classification and regression- mostly for classification. This is a supervised learning algorithm that considers different centroids and uses a usually Euclidean function to compare distance. Then, it analyses the results and classifies each point to the group to optimize it to place with all closest points to it. It classifies new cases using a majority vote of k of its neighbours. The case it assigns to a class is the one most common among its K nearest neighbours. *k*-NN is a special case of a variable bandwidth, kernel density 'balloon' estimator with a uniform kernel.



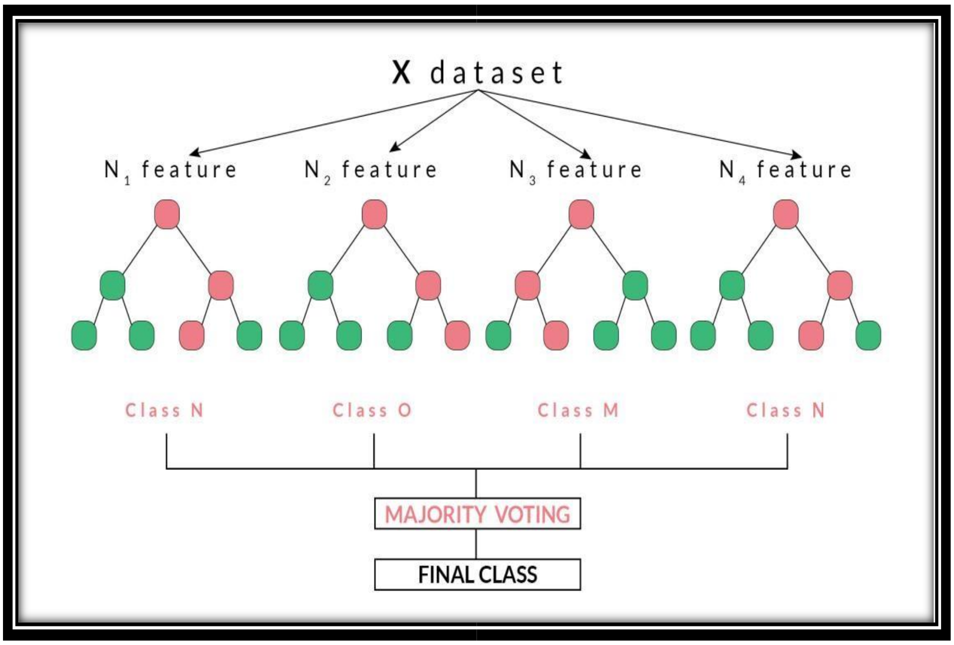
7. K-Means Algorithm-

K-Means is an unsupervised algorithm that solves the problem of clustering. It classifies data using a number of clusters. The data points inside a class are homogeneous and heterogeneous to peer groups. *k*-means clustering is a method of vector quantization originally from signal processing that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into K clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. *k*-means clustering is rather easy to apply to even large data sets, particularly when using heuristics such as Lloyd's algorithm. K-means originates from signal processing, and still finds use in this domain. In cluster analysis, the K-means algorithm can be used to partition the input data set into *k* partitions (clusters). K-means clustering has been used as a feature learning step, in either supervised learning or unsupervised learning.



8. Random Forest Algorithm-

A random forest is an ensemble of decision trees. In order to classify every new object based on its attributes, trees vote for class- each tree provides a classification. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.



X dataset

N_1 feature, N_2 feature, N_3 feature, N_4 feature

Class N, Class O, Class M, Class N

MAJORITY VOTING

FINAL CLASS

5. Data Preprocessing, Analysis & Visualization

Machine Learning algorithms don't work so well with processing raw data. Before we can feed such data to an ML algorithm, we must preprocess it. We must apply some transformations on it. With data preprocessing, we convert raw data into a clean data set. To perform data this, there are 7 techniques –

1. Rescaling Data -

For data with attributes of varying scales, we can rescale attributes to possess the same scale. We rescale attributes into the range 0 to 1 and call it normalization. We use the MinMaxScaler class from scikitlearn. This gives us values between 0 and 1.

2. Standardizing Data -

With standardizing, we can take attributes with a Gaussian distribution and different means and standard deviations and transform them into a standard Gaussian distribution with a mean of 0 and a standard deviation of 1.

3. Normalizing Data -

In this task, we rescale each observation to a length of 1 (a unit norm). For this, we use the Normalizer class.

4. Binarizing Data -

Using a binary threshold, it is possible to transform our data by marking the values above it 1 and those equal to or below it, 0. For this purpose, we use the Binarize class.

5. Mean Removal -

We can remove the mean from each feature to center it on zero.

6. One Hot Encoding -

When dealing with few and scattered numerical values, we may not need to store these. Then, we can perform One Hot Encoding. For k distinct values, we can transform the feature into a k-dimensional vector with one value of 1 and 0 as the rest values.

7. Label Encoding -

Some labels can be words or numbers. Usually, training data is labelled with words to make it readable. Label encoding converts word labels into numbers to let algorithms work on them.

6. Advantages & Disadvantages of Machine Learning

Advantages of Machine Learning

1. Automation of Everything

Machine Learning is responsible for cutting the workload and time. By automating things, we let the algorithm do the hard work for us. Automation is now being done almost everywhere. The reason is that it is very reliable. Also, it helps us to think more creatively.

2. Wide Range of Applications

ML has a wide variety of applications. This means that we can apply ML on any of the major fields. ML has its role everywhere from medical, business, banking to science and tech. This helps to create more opportunities. It plays a major role in customer interactions. Machine Learning can help in the detection of diseases more quickly. It is helping to lift up businesses. That is why investing in ML technology is worth it.

3. Scope of Improvement

Machine Learning is the type of technology that keeps on evolving. There is a lot of scope in ML to become the top technology in the future. This helps us to improve both hardware and software.

4. Efficient Handling of Data

Machine Learning has many factors that make it reliable. One of them is data handling. ML plays the biggest role when it comes to data at this time. It can handle any type of data. Machine learning can be multidimensional or different types of data.

5. Best for Education and Online Shopping

ML would be the best tool for education in the future. It provides very creative techniques to help students study. In online shopping, the ML model studies your searches. Based on your search history, it would provide advertisements. These will be about your search preferences in previous searches. In this, the search history is the data for the model. This is a great way to improve e-commerce with ML.

Disadvantages of Machine Learning

1. Possibility of High Error

In ML, we can choose the algorithms based on accurate results. For that, we have to run the results on every algorithm. The main problem occurs in the training and testing of data. The data is huge, so sometimes removing errors becomes nearly impossible. These errors can cause headache to users.

2. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

3. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for us.

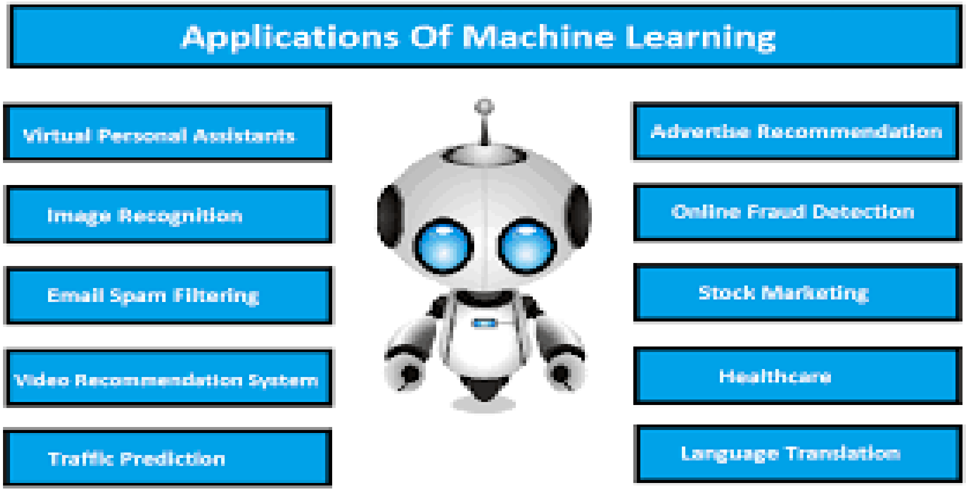
4. Algorithm Selection

The selection of an algorithm in Machine Learning is still a manual job. We have to run and test our data in all the algorithms. After that only we can decide what algorithm, we want. We choose them on the basis of result accuracy. The process is very time-consuming.

7. Applications of Machine Learning

Machine learning is a buzzword for today's technology, and it is growing very rapidly day by day. Some most trending real-world applications of Machine Learning are:

- Image Recognition
- Speech Recognition
- Traffic Prediction
- Self-driving cars
- Product recommendations
- Online Fraud Detection
- Email Spam and Malware Filtering
- Stock Market trading
- Medical Diagnosis
- Automatic Language Translation
- Virtual Personal Assistant



8. Problem Description

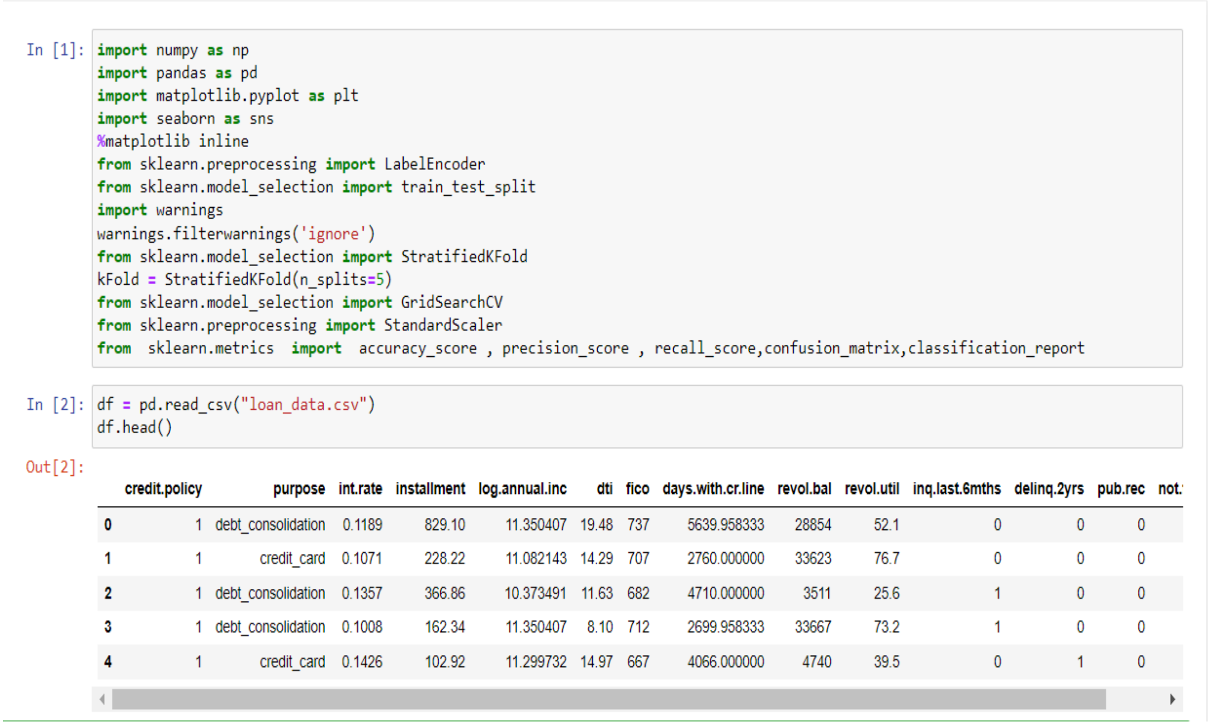
Use loan_data.csv to train the model. Predicts whether the bank should approve the loan of an applicant based on his profit using any two machine learning algorithms.

The libraries used in this project are:

- NumPy
- Pandas
- Matplotlib
- Seaborn
- SciKit Learn
- GridSearch CV

In this project I use two Algorithms namely

- Decision Tree Classifier Algorithm
- Random Forest Classifier Algorithm.



In [1]:

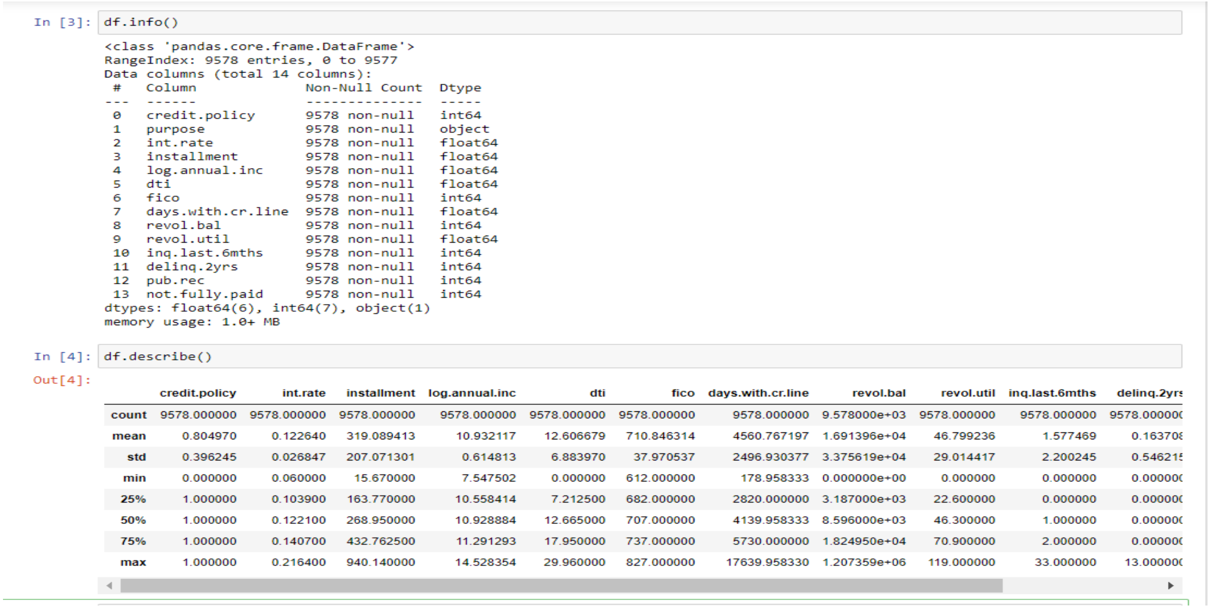
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import StratifiedKFold
kFold = StratifiedKFold(n_splits=5)
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from  sklearn.metrics  import  accuracy_score , precision_score , recall_score,confusion_matrix,classification_report
```

In [2]:

```
df = pd.read_csv("loan_data.csv")
df.head()
```

Out[2]:

	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs	pub.rec	not.
0	1	debt_consolidation	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	52.1	0	0	0	
1	1	credit_card	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	76.7	0	0	0	
2	1	debt_consolidation	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	25.6	1	0	0	
3	1	debt_consolidation	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	73.2	1	0	0	
4	1	credit_card	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	39.5	0	1	0	



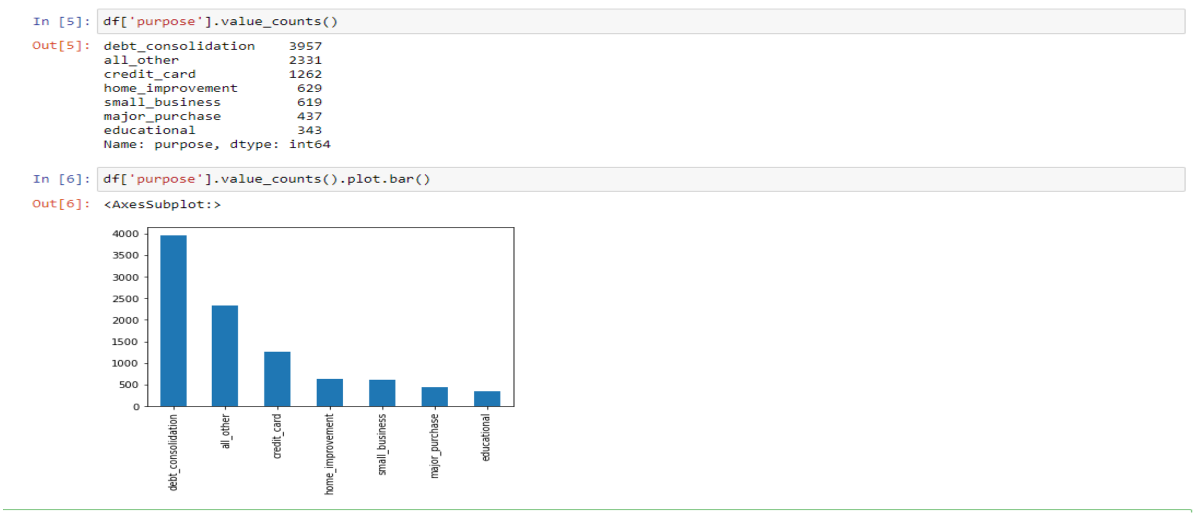
In [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9578 entries, 0 to 9577
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   credit.policy      9578 non-null   int64  
 1   purpose            9578 non-null   object 
 2   int.rate           9578 non-null   float64
 3   installment        9578 non-null   float64
 4   log.annual.inc     9578 non-null   float64
 5   dti                9578 non-null   float64
 6   fico               9578 non-null   int64  
 7   days.with.cr.line  9578 non-null   float64
 8   revol.bal          9578 non-null   int64  
 9   revol.util         9578 non-null   float64
 10  inq.last.6mths     9578 non-null   int64  
 11  delinq.2yrs        9578 non-null   int64  
 12  pub.rec            9578 non-null   int64  
 13  not.fully.paid     9578 non-null   int64  
dtypes: float64(6), int64(7), object(1)
memory usage: 1.0+ MB
```

In [4]:
```
df.describe()
```

Out[4]:

	credit.policy	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs
count	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9578.000000	9.578000e+03	9578.000000	9578.000000	9578.000000
mean	0.804970	0.122640	319.089413	10.932117	12.606679	710.846314	4560.767197	1.691396e+04	46.799236	1.577469	0.163708
std	0.396245	0.026847	207.071301	0.614813	6.883970	37.970537	2496.930377	3.375619e+04	29.014417	2.200245	0.546215
min	0.000000	0.060000	15.670000	7.547502	0.000000	612.000000	178.958333	0.000000e+00	0.000000	0.000000	0.000000
25%	1.000000	0.103900	163.770000	10.558414	7.212500	682.000000	2820.000000	3.187000e+03	22.600000	0.000000	0.000000
50%	1.000000	0.122100	268.950000	10.928884	12.665000	707.000000	4139.958333	8.596000e+03	46.300000	1.000000	0.000000
75%	1.000000	0.140700	432.762500	11.291293	17.950000	737.000000	5730.000000	1.824950e+04	70.900000	2.000000	0.000000
max	1.000000	0.216400	940.140000	14.528354	29.960000	827.000000	17639.958330	1.207359e+06	119.000000	33.000000	13.000000

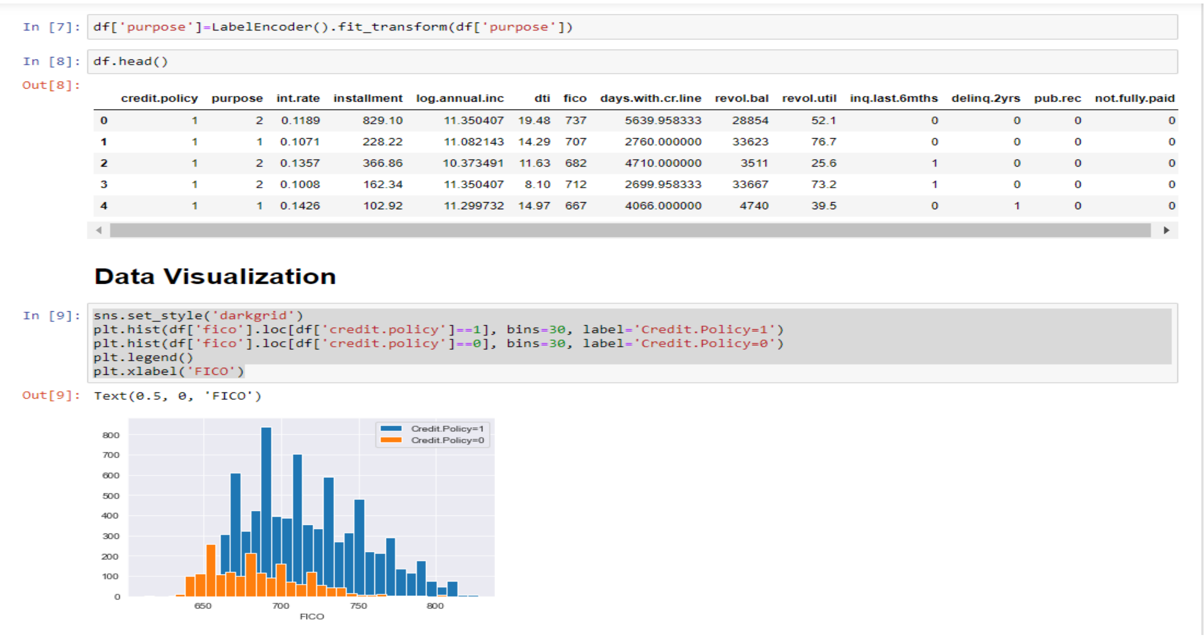


In [5]:
```
df['purpose'].value_counts()
```

Out[5]:
```
debt_consolidation    3957
all_other             2331
credit_card           1262
home_improvement       629
small_business         619
major_purchase         437
educational            343
Name: purpose, dtype: int64
```

In [6]:
```
df['purpose'].value_counts().plot.bar()
```

Out[6]:
```
<AxesSubplot:>
```



In [7]:
```
df['purpose']=LabelEncoder().fit_transform(df['purpose'])
```

In [8]:
```
df.head()
```

Out[8]:

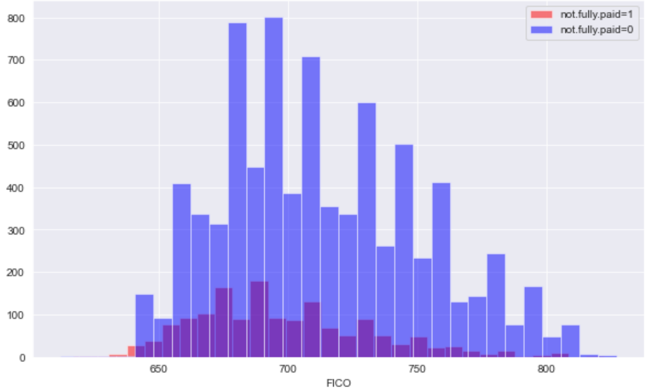
	credit.policy	purpose	int.rate	installment	log.annual.inc	dti	fico	days.with.cr.line	revol.bal	revol.util	inq.last.6mths	delinq.2yrs	pub.rec	not.fully.paid
0	1	2	0.1189	829.10	11.350407	19.48	737	5639.958333	28854	52.1	0	0	0	0
1	1	1	0.1071	228.22	11.082143	14.29	707	2760.000000	33623	76.7	0	0	0	0
2	1	2	0.1357	366.86	10.373491	11.63	682	4710.000000	3511	25.6	1	0	0	0
3	1	2	0.1008	162.34	11.350407	8.10	712	2699.958333	33667	73.2	1	0	0	0
4	1	1	0.1426	102.92	11.299732	14.97	667	4066.000000	4740	39.5	0	1	0	0

Data Visualization

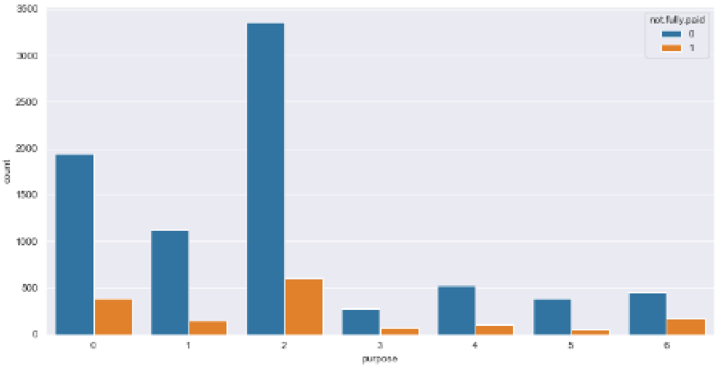
In [9]:
```
sns.set_style('darkgrid')
plt.hist(df['fico'].loc[df['credit.policy']==1], bins=30, label='Credit.Policy=1')
plt.hist(df['fico'].loc[df['credit.policy']==0], bins=30, label='Credit.Policy=0')
plt.legend()
plt.xlabel('FICO')
```

Out[9]:
```
Text(0.5, 0, 'FICO')
```

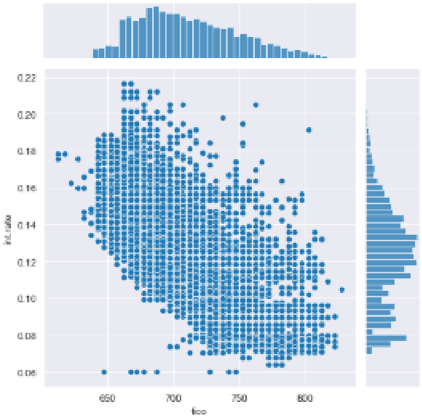
```
In [10]: #sns.set_style('white')
         plt.figure(figsize=(10,6))
         df[df['not.fully.paid']==1]['fico'].hist(bins=30, alpha=0.5, color='red', label='not.fully.paid=1')
         df[df['not.fully.paid']==0]['fico'].hist(bins=30, alpha=0.5, color='blue', label='not.fully.paid=0')
         plt.legend()
         plt.xlabel('FICO')

Out[10]: Text(0.5, 0, 'FICO')
```



```
In [11]: #creating a countplot to see the counts of purpose of loans by not.fully.paid
         plt.figure(figsize=(12,6))
         sns.countplot(data=df, x='purpose', hue='not.fully.paid')

Out[11]: <AxesSubplot:xlabel='purpose', ylabel='count'>
```

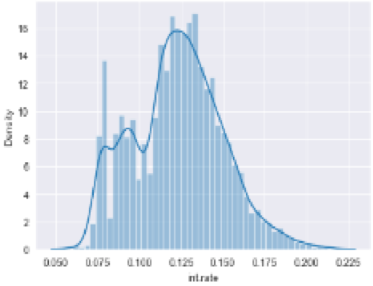


```
In [12]: #checking the trend between FICO and the interest rate
         plt.figure(figsize=(10,6))
         sns.jointplot(x='fico', y='int.rate', data=df)

Out[12]: <seaborn.axisgrid.JointGrid at 0x21846bbc6d0>

         <Figure size 720x432 with 0 Axes>
```

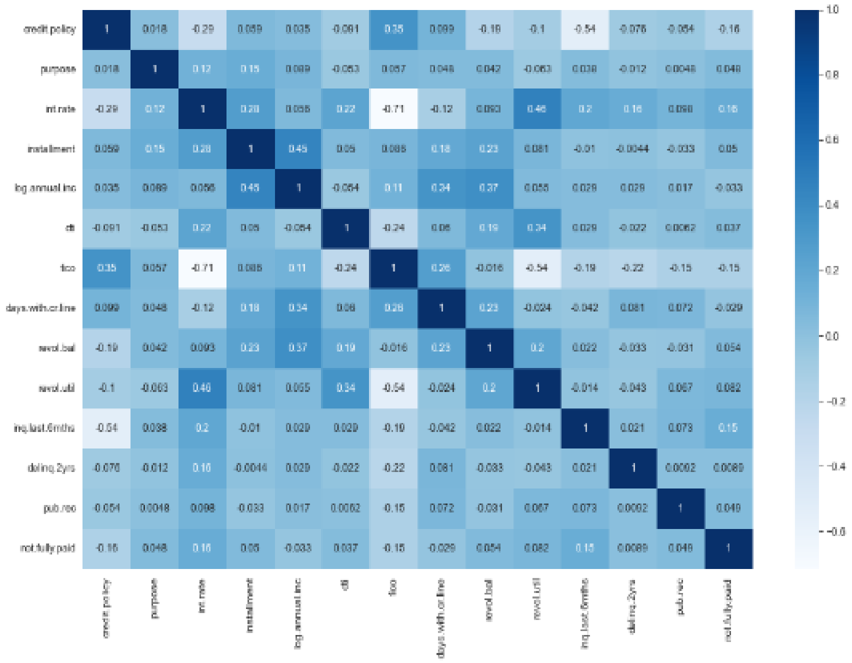


```
In [13]: sns.distplot(df["int.rate"])
```

```
Out[13]: <AxesSubplot:xlabel='int.rate', ylabel='Density'>
```



```
In [14]: plt.figure(figsize = (15, 9))
         sns.heatmap(df.corr(), cmap='Blues', annot=True)
         plt.show()
```



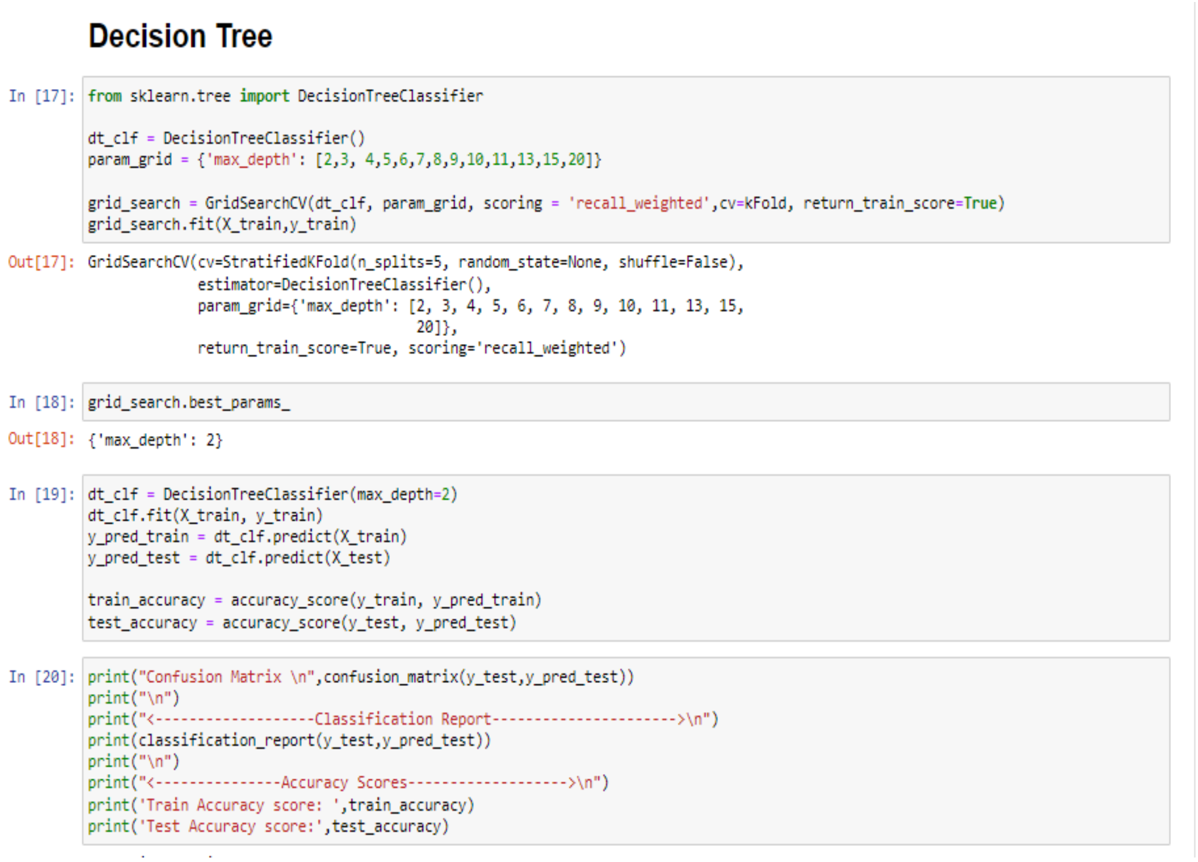
Train-Test Split

Splitting the dataset for training and testing purpose.

```
In [15]: # Dropping target class

         X = df.drop('not.fully.paid',axis=1)
         y = df['not.fully.paid']
```

```
In [16]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,random_state=101)
```

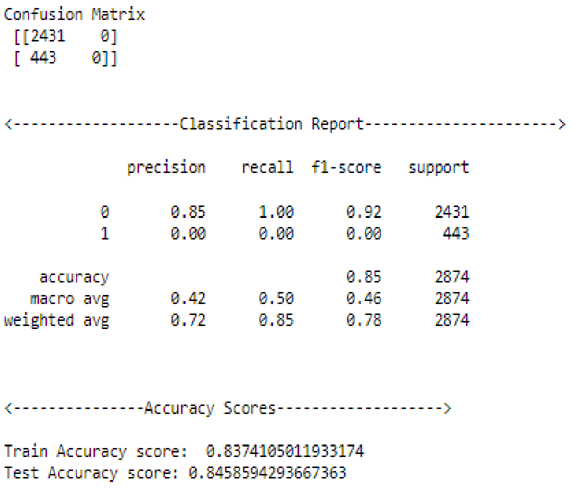
- ## Using Decision Tree Classifier Algorithm



Decision Tree

```
In [17]: from sklearn.tree import DecisionTreeClassifier

         dt_clf = DecisionTreeClassifier()
         param_grid = {'max_depth': [2,3, 4,5,6,7,8,9,10,11,13,15,20]}

         grid_search = GridSearchCV(dt_clf, param_grid, scoring = 'recall_weighted',cv=kFold, return_train_score=True)
         grid_search.fit(X_train,y_train)

Out[17]: GridSearchCV(cv=StratifiedKFold(n_splits=5, random_state=None, shuffle=False),
                      estimator=DecisionTreeClassifier(),
                      param_grid={'max_depth': [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15,
                                                20]},
                      return_train_score=True, scoring='recall_weighted')

In [18]: grid_search.best_params_

Out[18]: {'max_depth': 2}

In [19]: dt_clf = DecisionTreeClassifier(max_depth=2)
         dt_clf.fit(X_train, y_train)
         y_pred_train = dt_clf.predict(X_train)
         y_pred_test = dt_clf.predict(X_test)

         train_accuracy = accuracy_score(y_train, y_pred_train)
         test_accuracy = accuracy_score(y_test, y_pred_test)

In [20]: print("Confusion Matrix \n",confusion_matrix(y_test,y_pred_test))
         print("\n")
         print("<------------------Classification Report--------------------->\n")
         print(classification_report(y_test,y_pred_test))
         print("\n")
         print("<--------------Accuracy Scores----------------->\n")
         print('Train Accuracy score: ',train_accuracy)
         print('Test Accuracy score:',test_accuracy)
```

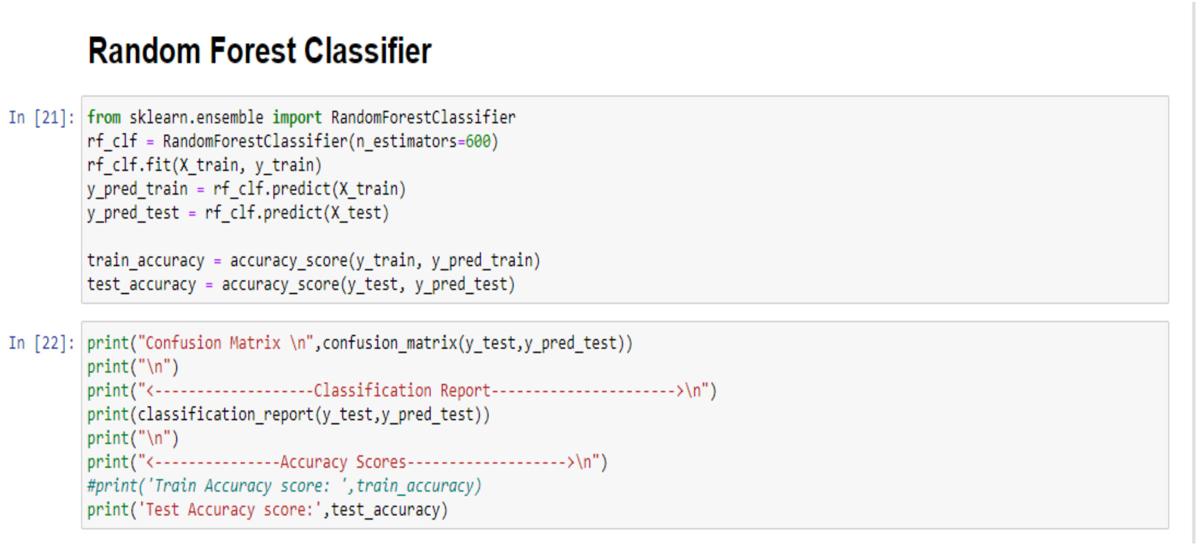
Output:



```
Confusion Matrix
 [[2431    0]
 [ 443    0]]


<------------------Classification Report--------------------->

              precision    recall  f1-score   support

           0       0.85      1.00      0.92      2431
           1       0.00      0.00      0.00       443

    accuracy                           0.85      2874
   macro avg       0.42      0.50      0.46      2874
weighted avg       0.72      0.85      0.78      2874



<--------------Accuracy Scores----------------->

Train Accuracy score:  0.8374105011933174
Test Accuracy score: 0.8458594293667363
```

We got Accuracy of 84.58% using Decision Tree Classifier.

- # Random Forest Classifier Algorithm

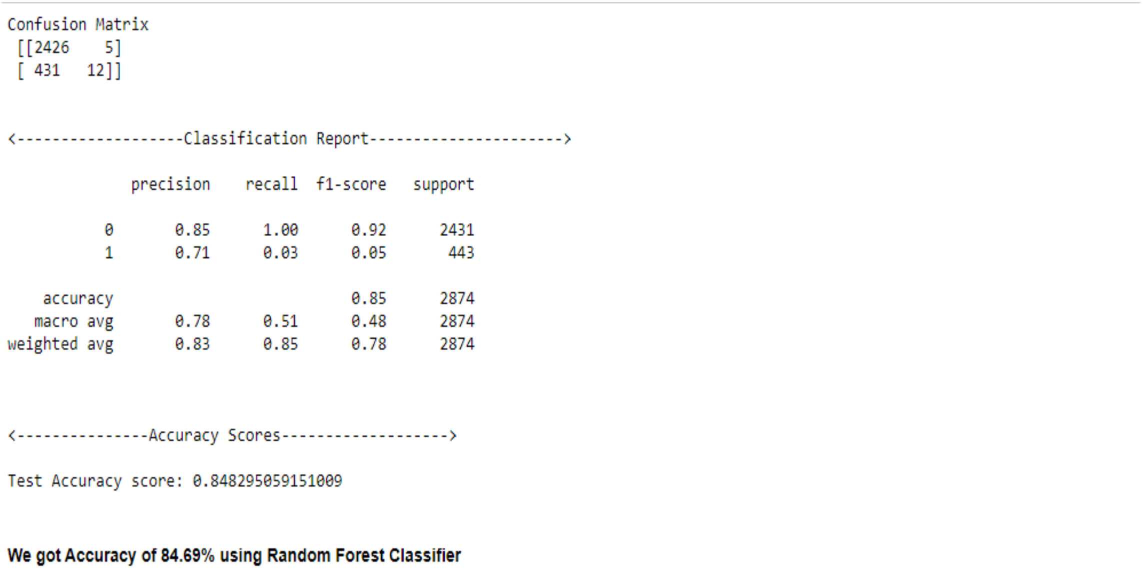


Random Forest Classifier

```
In [21]: from sklearn.ensemble import RandomForestClassifier
         rf_clf = RandomForestClassifier(n_estimators=600)
         rf_clf.fit(X_train, y_train)
         y_pred_train = rf_clf.predict(X_train)
         y_pred_test = rf_clf.predict(X_test)

         train_accuracy = accuracy_score(y_train, y_pred_train)
         test_accuracy = accuracy_score(y_test, y_pred_test)
```

```
In [22]: print("Confusion Matrix \n",confusion_matrix(y_test,y_pred_test))
         print("\n")
         print("<------------------Classification Report--------------------->\n")
         print(classification_report(y_test,y_pred_test))
         print("\n")
         print("<--------------Accuracy Scores------------------>\n")
         #print('Train Accuracy score: ',train_accuracy)
         print('Test Accuracy score:',test_accuracy)
```

Output:



```
Confusion Matrix
 [[2426    5]
 [ 431   12]]


<------------------Classification Report--------------------->

              precision    recall  f1-score   support

           0       0.85      1.00      0.92      2431
           1       0.71      0.03      0.05       443

    accuracy                           0.85      2874
   macro avg       0.78      0.51      0.48      2874
weighted avg       0.83      0.85      0.78      2874



<--------------Accuracy Scores------------------>

Test Accuracy score: 0.848295059151009
```

We got Accuracy of 84.69% using Random Forest Classifier

On comparing Decision Tree Classifier Algorithm and Random Forest Classifier Algorithm I got the highest accuracy in Random Forest Classifier Algorithm of **84.69%.** So, the best approach for this dataset is Random Forest Classifier Algorithm.

9. Reason for Choosing Machine Learning

- **Learning machine learning brings in better career opportunities**
  - Machine learning is the shining star of the moment.
  - Every industry looking to apply AI in their domain, studying machine learning opens world of opportunities to develop cutting edge machine learning applications in various verticals – such as cyber security, image recognition, medicine or face recognition.
  - Several machine learning companies on the verge of skilled ML engineers, it is becoming the brain behind business intelligence.
- **Machine Learning Jobs on the rise**
  - The major hiring is happening in all top tech companies in search of those special kind of people (machine learning engineers) who can build a hammer (machine learning algorithms).
  - The job market for machine learning engineers is not just hot but it's sizzling.
- **Machine Learning jobs come with big pay checks**
  - Being one of the most in-demand skills right now, ML jobs pay very well. According to Glassdoor, the average annual salary of ML jobs in India is anywhere between Rs. 4,60,000 and Rs. 11,00,000.
- **Machine Learning helps you to understand customers better**
  - ML technologies and solutions can dig into customer data and understand individual needs, preferences, and pain points of each customer segment. This, in turn, allows companies to create highly personalized products/services, offers and discounts, and marketing strategies to cater to individual customer needs.
  - In the long run, a company can maintain long-term relationships with happy and satisfied customers. This has a significant bearing on company ROI.

10. Learning Outcomes

- ✓ Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.

- ✓ Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.

- ✓ Have an understanding of the strengths and weaknesses of many popular machine learning approaches.

- ✓ Ability to integrate machine learning libraries and mathematical and statistical tools with modern technologies.

- ✓ Be able to design and implement various machine learning algorithms in a range of real-world applications.

- ✓ Ability to understand and apply scaling up machine learning techniques and associated computing techniques and technologies.