THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Bank Marketing Analysis

UML501 Machine Learning Project Report

Submitted by:

Siddharth Chaudhary (102003707)

Anirudh Kumar (102003714)

BE Third Year, COE-28

Submitted to: Dr. Harpreet Singh

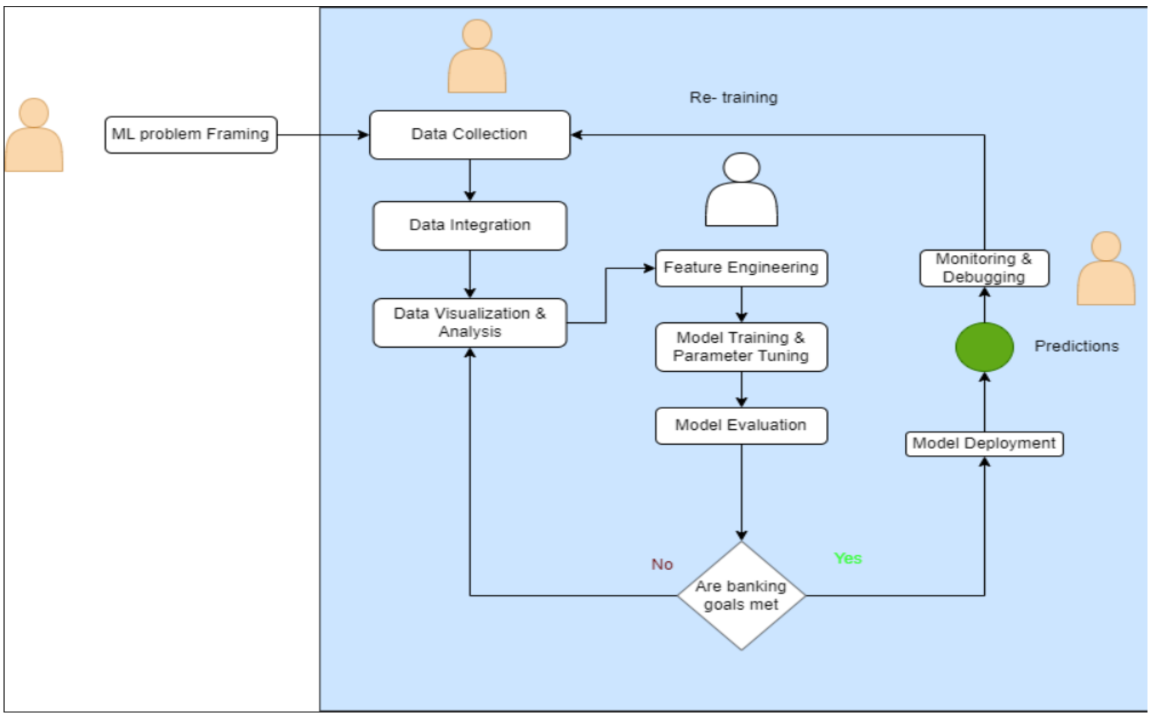
Introduction

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. Getting to know what customers bring value to the business is the key to any organization. Especially for the banks, in cases like predicting whether a customer will be a default or not, whether a customer will take term deposit or not. These types of situations need a proper analysis because a few customers can make a lot of difference for the banks. There are 5 numerical, 11 categorical variables with one target variable. The classification goal is to predict if the client will subscribe a term deposit (variable y).

Machine learning is a subfield of artificial intelligence (AI). The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people. Although machine learning is a field within computer science, it differs from traditional computational approaches. In traditional computing, algorithms are sets of explicitly programmed instructions used by computers to calculate or problem solve. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data to automate decision-making processes based on data inputs. Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Optical character recognition (OCR) technology converts images of text into movable type. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Self-driving cars that rely on machine learning to navigate may soon be available to consumers. Predicting and analyzing student performance are critical to assisting educators in recognizing students' weaknesses while helping them improve their grades. Likewise, students can improve their learning activities, and administrators can improve their operations. The timely prediction of student performance allows educators to identify low-performing individuals and intervene early in the learning process to apply the necessary interventions. ML is a novel approach with numerous applications that can make predictions on data. ML techniques in educational data mining aim to model and detect meaningful hidden patterns and useable information from educational contexts. Moreover, in the academic field, the ML approaches are applied to large datasets to represent a wide range of student characteristics as data points. These strategies can benefit various fields by achieving various goals, including extracting patterns, predicting behavior, or identifying trends, which allow educators to deliver the most effective methods for learning and to track and monitor the students' progress.

Contents of further section includes background in which you need to write about at least 10 articles which have solved the same problem and provide their references. Further explain if some feature selection/ extraction method is applied such as correlation matrix/ PCA etc.

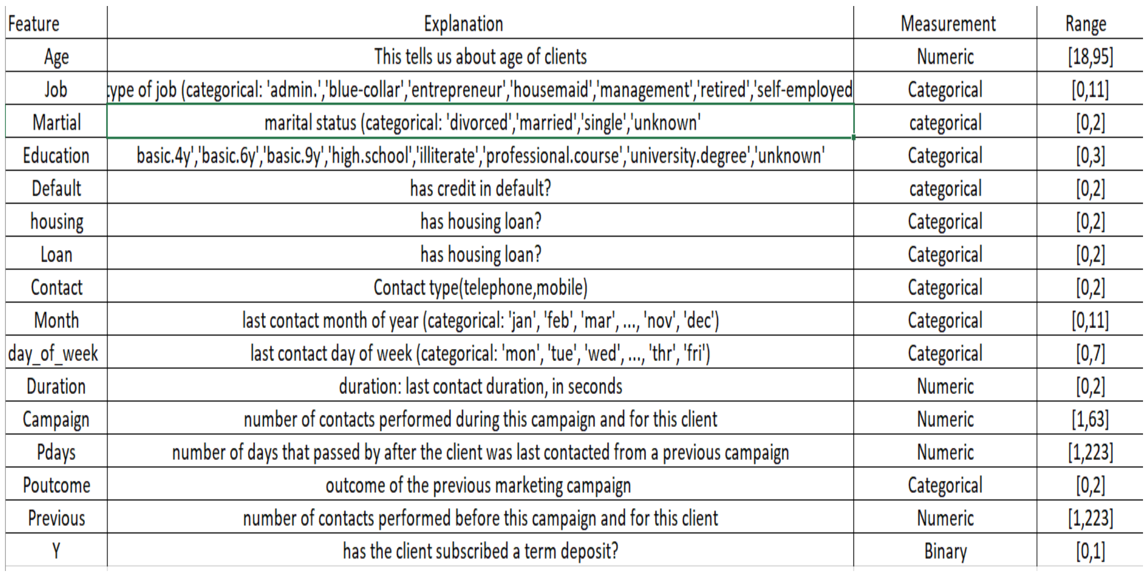
Architecture



Dataset Description

Problem Statement- The data is related with direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict if the client will subscribe a term deposit (variable y).

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.



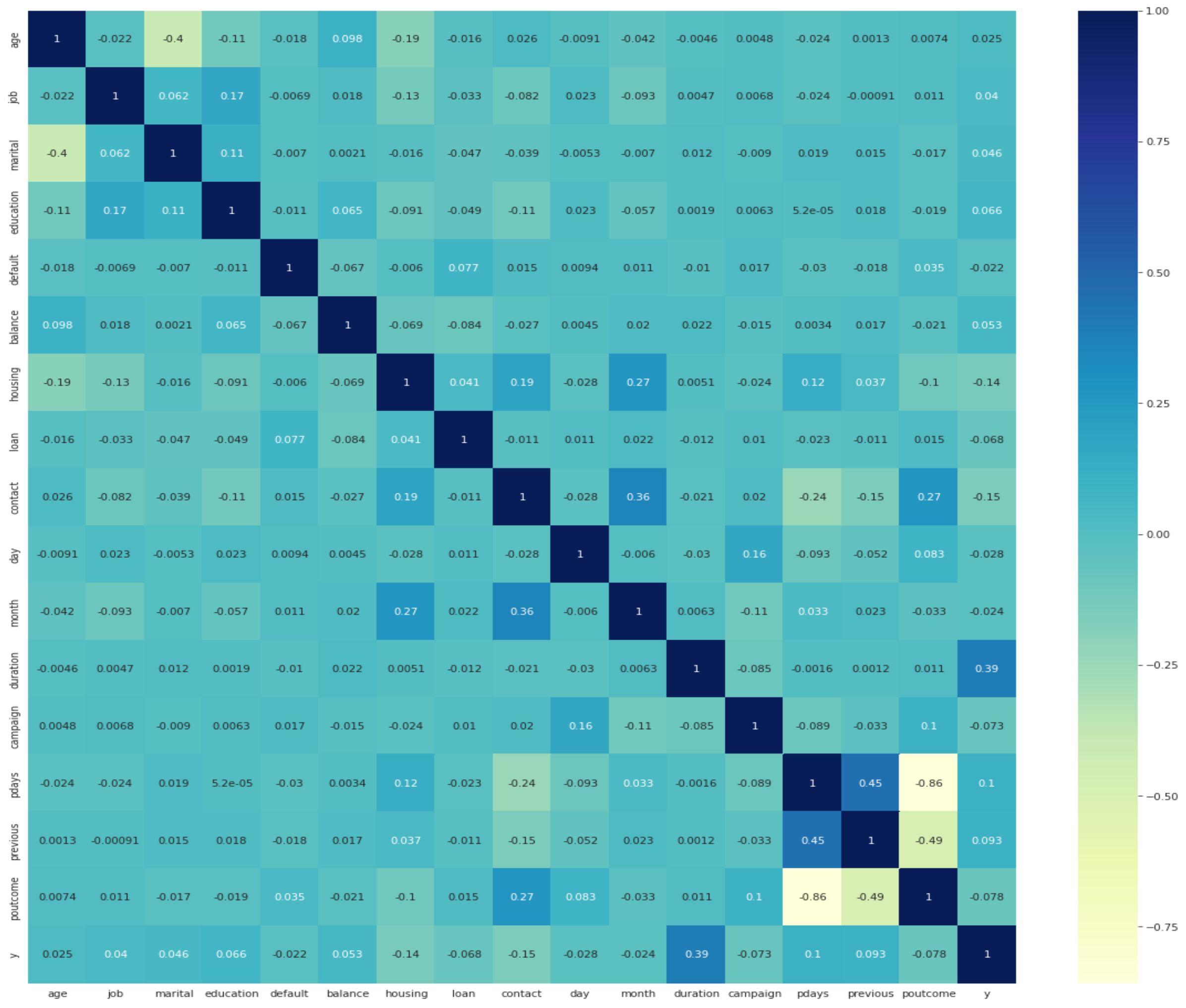
Feature	Explanation	Measurement	Range
Age	This tells us about age of clients	Numeric	[18,95]
Job	ype of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed	Categorical	[0,11]
Martial	marital status (categorical: 'divorced','married','single','unknown'	categorical	[0,2]
Education	basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown'	Categorical	[0,3]
Default	has credit in default?	categorical	[0,2]
housing	has housing loan?	Categorical	[0,2]
Loan	has housing loan?	Categorical	[0,2]
Contact	Contact type(telephone,mobile)	Categorical	[0,2]
Month	last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')	Categorical	[0,11]
day_of_week	last contact day of week (categorical: 'mon', 'tue', 'wed', ..., 'thr', 'fri')	Categorical	[0,7]
Duration	duration: last contact duration, in seconds	Numeric	[0,2]
Campaign	number of contacts performed during this campaign and for this client	Numeric	[1,63]
Pdays	number of days that passed by after the client was last contacted from a previous campaign	Numeric	[1,223]
Poutcome	outcome of the previous marketing campaign	Categorical	[0,2]
Previous	number of contacts performed before this campaign and for this client	Numeric	[1,223]
Y	has the client subscribed a term deposit?	Binary	[0,1]

Pre-processing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data preprocessing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

HEATMAP

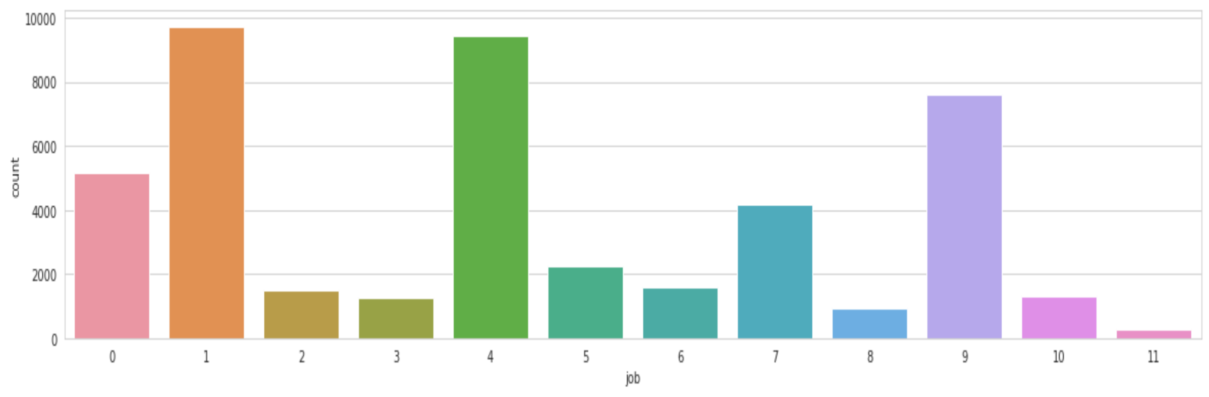
A heat map is a two-dimensional representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of information. More elaborate heat maps allow the viewer to understand complex data sets. There can be many ways to display heat maps, but they all share one thing in common -- they use color to communicate relationships between data values that would be much harder to understand if presented numerically in a spreadsheet.

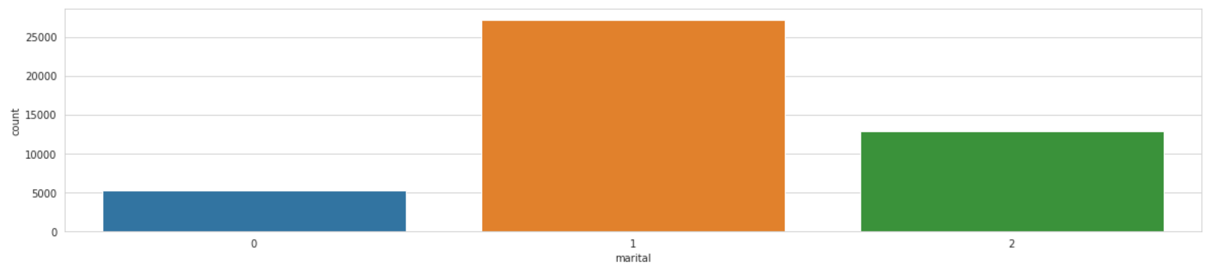


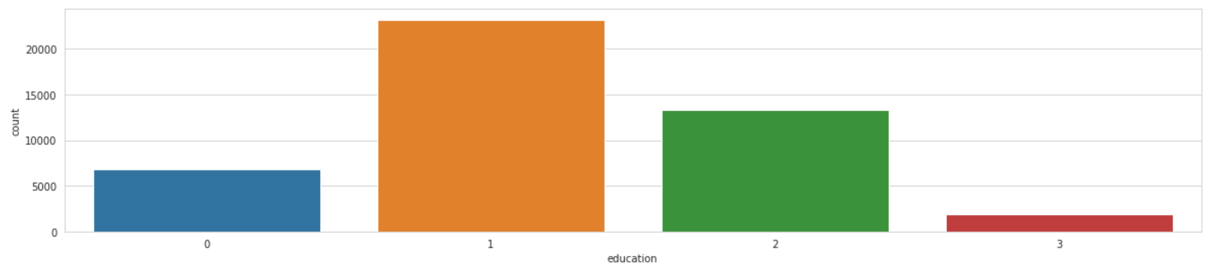
	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
age	1	-0.022	-0.4	-0.11	-0.018	0.098	-0.19	-0.016	0.026	-0.0091	-0.042	-0.0046	0.0048	-0.024	0.0013	0.0074	0.025
job	-0.022	1	0.062	0.17	-0.0069	0.018	-0.13	-0.033	-0.082	0.023	-0.093	0.0047	0.0068	-0.024	-0.00091	0.011	0.04
marital	-0.4	0.062	1	0.11	-0.007	0.0021	-0.016	-0.047	-0.039	-0.0053	-0.007	0.012	-0.009	0.019	0.015	-0.017	0.046
education	-0.11	0.17	0.11	1	-0.011	0.065	-0.091	-0.049	-0.11	0.023	-0.057	0.0019	0.0063	5.2e-05	0.018	-0.019	0.066
default	-0.018	-0.0069	-0.007	-0.011	1	-0.067	-0.006	0.077	0.015	0.0094	0.011	-0.01	0.017	-0.03	-0.018	0.035	-0.022
balance	0.098	0.018	0.0021	0.065	-0.067	1	-0.069	-0.084	-0.027	0.0045	0.02	0.022	-0.015	0.0034	0.017	-0.021	0.053
housing	-0.19	-0.13	-0.016	-0.091	-0.006	-0.069	1	0.041	0.19	-0.028	0.27	0.0051	-0.024	0.12	0.037	-0.1	-0.14
loan	-0.016	-0.033	-0.047	-0.049	0.077	-0.084	0.041	1	-0.011	0.011	0.022	-0.012	0.01	-0.023	-0.011	0.015	-0.068
contact	0.026	-0.082	-0.039	-0.11	0.015	-0.027	0.19	-0.011	1	-0.028	0.36	-0.021	0.02	-0.24	-0.15	0.27	-0.15
day	-0.0091	0.023	-0.0053	0.023	0.0094	0.0045	-0.028	0.011	-0.028	1	-0.006	-0.03	0.16	-0.093	-0.052	0.083	-0.028
month	-0.042	-0.093	-0.007	-0.057	0.011	0.02	0.27	0.022	0.36	-0.006	1	0.0063	-0.11	0.033	0.023	-0.033	-0.024
duration	-0.0046	0.0047	0.012	0.0019	-0.01	0.022	0.0051	-0.012	-0.021	-0.03	0.0063	1	-0.085	-0.0016	0.0012	0.011	0.39
campaign	0.0048	0.0068	-0.009	0.0063	0.017	-0.015	-0.024	0.01	0.02	0.16	-0.11	-0.085	1	-0.089	-0.033	0.1	-0.073
pdays	-0.024	-0.024	0.019	5.2e-05	-0.03	0.0034	0.12	-0.023	-0.24	-0.093	0.033	-0.0016	-0.089	1	0.45	-0.86	0.1
previous	0.0013	-0.00091	0.015	0.018	-0.018	0.017	0.037	-0.011	-0.15	-0.052	0.023	0.0012	-0.033	0.45	1	-0.49	0.093
poutcome	0.0074	0.011	-0.017	-0.019	0.035	-0.021	-0.1	0.015	0.27	0.083	-0.033	0.011	0.1	-0.86	-0.49	1	-0.078
y	0.025	0.04	0.046	0.066	-0.022	0.053	-0.14	-0.068	-0.15	-0.028	-0.024	0.39	-0.073	0.1	0.093	-0.078	1

COUNT PLOT

A univariate plot that shows the comparison of different groups in categorical variables. It shows the number of observations per category using bins. For example, number of males and females in the gender feature of a dataset or number of passengers in different class (class A, class B and class C) of another dataset or as shown below the count of each facility type attribute. To compare and count the number of observations of different groups within a categorical feature.

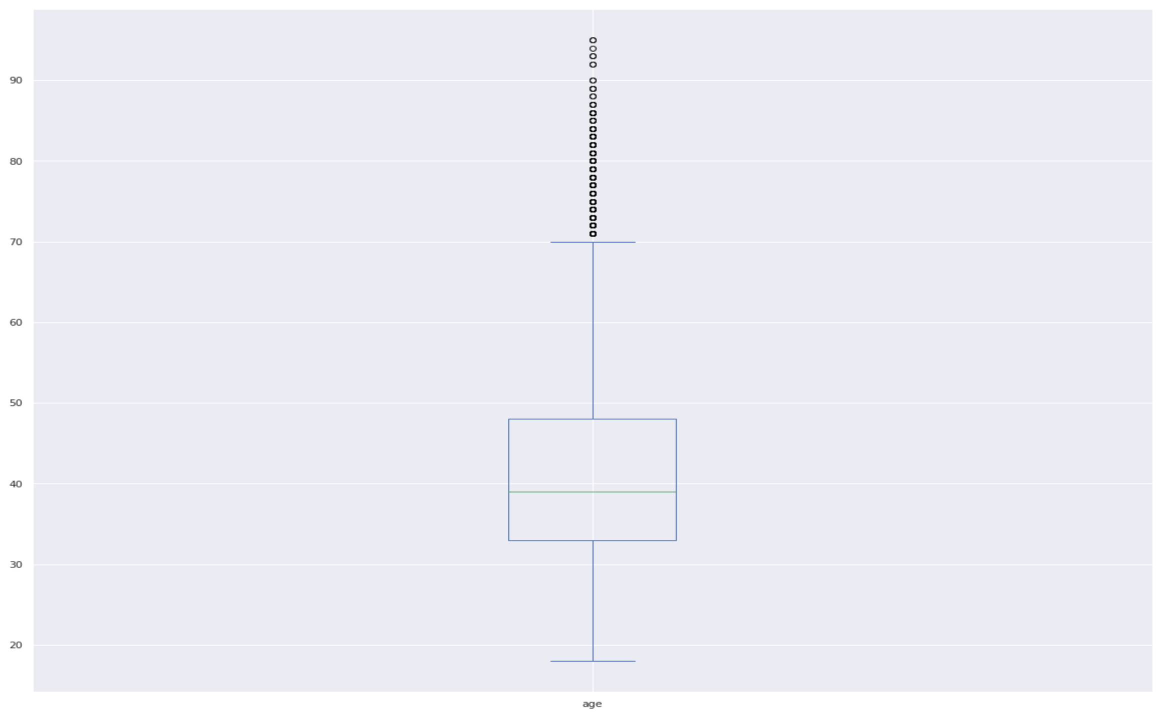




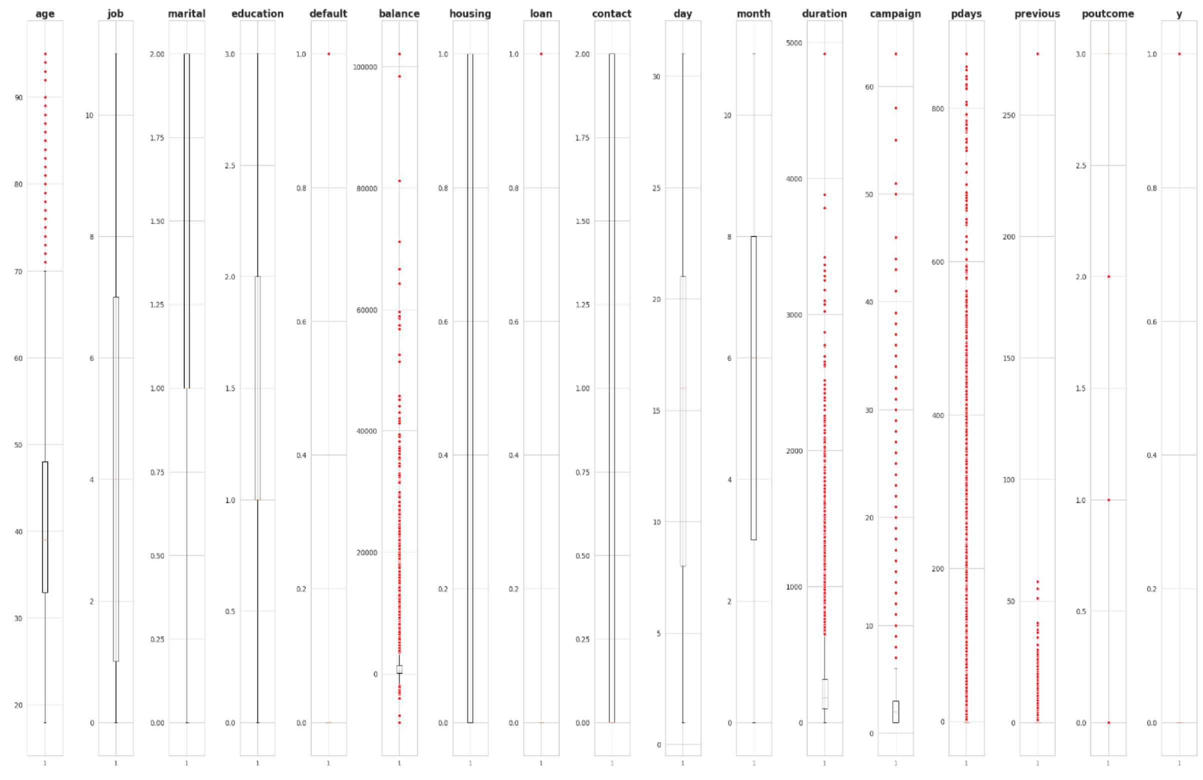


BOX PLOT

A very simple plot which provides important information about the spread and distribution of an attribute. This can be used on individual or to compare 2 or more attributes. Boxplots are easy to read and summarizes a lot of information in a single graph.



90
80
70
60
50
40
30
20
age

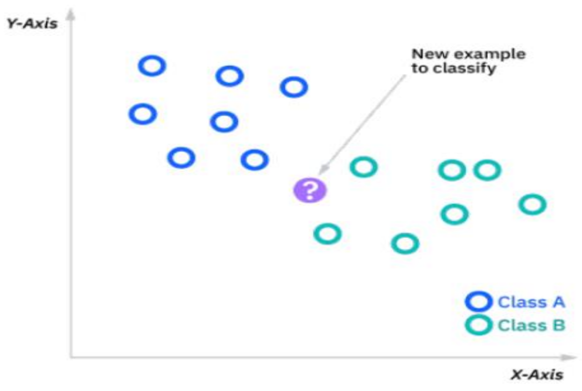


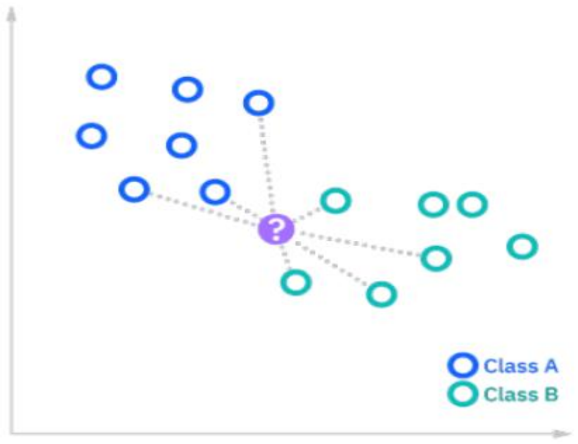
age
job
marital
education
default
balance
housing
loan
contact
day
month
duration
campaign
pdays
previous
poutcome
y

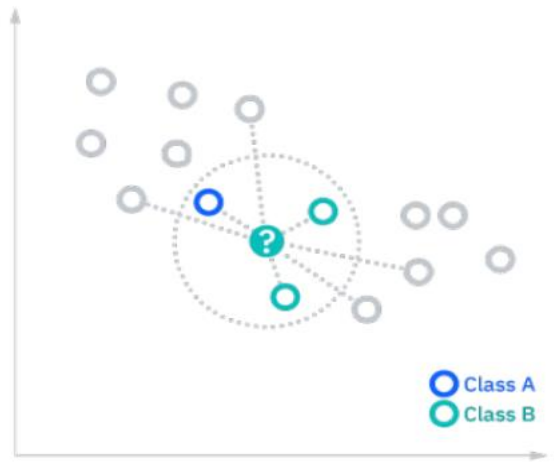
KNN CLASSIFIER

The k-nearest neighbours' algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.

For classification problems, a class label is assigned based on a majority vote—i.e., the label that is most frequently represented around a given data point is used. While this is technically considered "plurality voting", the term, "majority vote" is more commonly used in literature. The distinction between these terminologies is that "majority voting" technically requires a majority of greater than 50%, which primarily works when there are only two categories. When you have multiple classes—e.g., four categories, you do not necessarily need 50% of the vote to make a conclusion about a class; you could assign a class label with a vote of greater than 25%.







The goal of the k-nearest neighbours' algorithm is to identify the nearest neighbours of a given query point, so that we can assign a class label to that point. To do this, KNN has a few requirements:

Determine distance metrics-

Euclidean distance (p=2):

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Manhattan distance (p=1):

$$d(x,y) = \left(\sum_{i=1}^{m} \left| x_i - y_i \right| \right)$$

Minkowski distance:

$$\text{Minkowski Distance} = \left(\sum_{i=1}^{n} \left| x_i - y_i \right| \right)^{1/p}$$

Hamming distance:

$$\text{Hamming Distance} = D_H = \left(\sum_{i=1}^{k} \left| x_i - y_i \right| \right)$$

$$\begin{matrix} x=y & D=0 \\ x \neq y & D \neq 1 \end{matrix}$$

The following code is an example of how to create and predict with a KNN model:

```
from sklearn.neighbors import KNeighborsClassifier
model_name = ‘K-Nearest Neighbor Classifier’
knnClassifier = KNeighborsClassifier(n_neighbors = 5, metric = ‘minkowski’, p=2)
knn_model = Pipeline(steps=[(‘preprocessor’, preprocessorForFeatures), (‘classifier’ ,
knnClassifier)])
knn_model.fit(X_train, y_train)
y_pred = knn_model.predict(X_test)
```

Advantages and disadvantages of the KNN algorithm:

Advantages:

-Easy to implement: Given the algorithm’s simplicity and accuracy, it is one of the first classifiers that a new data scientist will learn.

- Adapts easily**:** As new training samples are added, the algorithm adjusts to account for any new data since all training data is stored into memory.

- Few hyperparameters**:** KNN only requires a k value and a distance metric, which is low when compared to other machine learning algorithms.

Disadvantages:

- Does not scale well: Since KNN is a lazy algorithm, it takes up more memory and data storage compared to other classifiers. This can be costly from both a time and money perspective. More memory and storage will drive up business expenses and more data can take longer to compute. While different data structures, such as Ball-Tree, have been created to address the computational inefficiencies, a different classifier may be ideal depending on the business problem.

- Curse of dimensionality: The KNN algorithm tends to fall victim to the curse of dimensionality, which means that it does not perform well with high-dimensional data inputs. This is sometimes also referred to as the peaking phenomenon (PDF, 340 MB) (link resides outside of ibm.com), where after the algorithm attains the optimal number of features, additional features increase the amount of classification errors, especially when the sample size is smaller.

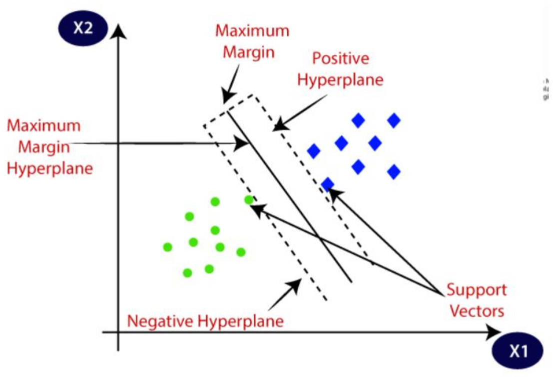
- Prone to overfitting: Due to the “curse of dimensionality”, KNN is also more prone to overfitting. While feature selection and dimensionality reduction techniques are leveraged to prevent this from occurring, the value of k can also impact the model’s behaviour. Lower values of k can overfit the data, whereas higher values of k tend to “smooth out” the prediction values since it is averaging the values over a greater area, or neighbourhood. However, if the value of k is too high, then it can underfit the data.

LINEAR SVM

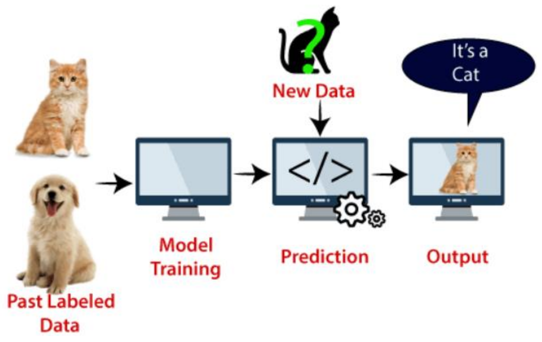
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Example: SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat.



SVM can be of two types:

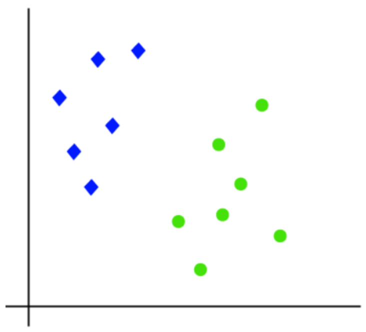
- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- **Non-linear SVM:** Non-Linear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

Hyperplane: There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM. The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane. We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

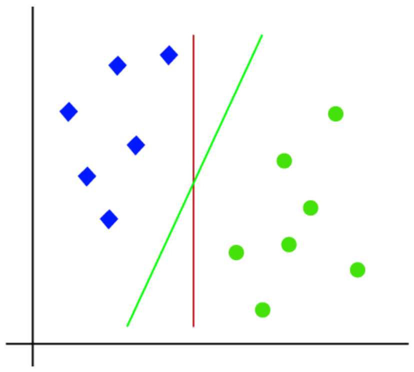
Support Vectors: The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

Linear SVM:

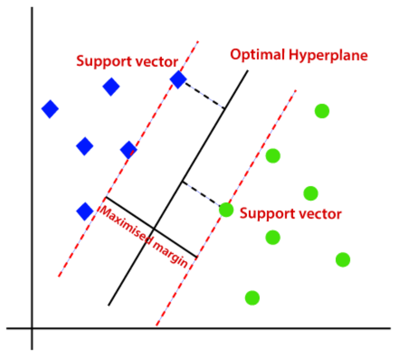
The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So, as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin. The **hyperplane** with maximum margin is called the **optimal hyperplane**.



NAÏVE BAYES CLASSIFIER

This model is easy to build and is mostly used for large datasets. It is a probabilistic machine learning model that is used for classification problems. The core of the classifier depends on the Bayes theorem with an assumption of independence among predictors. That means changing the value of a feature does not change the value of another feature.

Why is it called Naive?

It is called Naive because of the assumption that 2 variables are independent when they may not be. In a real-world scenario, there is hardly any situation where the features are independent. Since it is a probabilistic approach, the predictions can be made really quick. It can be used for both binary and multi-class classification problems.

Conditional Probability for Naive Bayes:

Conditional probability is defined as the likelihood of an event or outcome occurring, based on the occurrence of a previous event or outcome. Conditional probability is calculated by multiplying the probability of the preceding event by the updated probability of the succeeding, or conditional, event.

Mathematically, the conditional probability of event A given event B has already happened is given by:

$$\underbrace{P(A \mid B)}_{\text{Probability of event A given B has occured}} = \frac{\overset{\text{Probability of event A occured and event B occured}}{P(A \cap B)}}{\underbrace{P(B)}_{\text{Probability of event B}}}$$

Bayes Rule

Now we are prepared to state one of the most useful results in conditional probability: Bayes’ Rule. Bayes’ theorem which was given by Thomas Bayes, a British Mathematician, in 1763 provides a means for calculating the probability of an event given some information.

Mathematically Bayes theorem can be stated as:

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

Basically, we are trying to find the probability of event A, given event B is true. Here P(B) is called prior probability which means it is the probability of an event before the evidence P(B|A) is called the posterior probability i.e., Probability of an event after the evidence is seen. Bayes' rule provides us with the formula for the probability of Y given some feature X. In real-world problems, we hardly find any case where there is only one feature. When the features are independent, we can extend Bayes' rule to what is called Naive Bayes which assumes that the features are independent that means changing the value of one feature does not influence the values of other variables and therefore we call this algorithm "*NAIVE*". Naive Bayes can be used for various things like face recognition, weather prediction, Medical Diagnosis, News classification, Sentiment Analysis, and a lot more. When there are multiple X variables, we simplify it by assuming that X's are independent, so

$$P(Y=k|X) = \frac{P(X|Y=k) * P(Y=k)}{P(X)}$$

For n number of X, the formula becomes **Naive Bayes**:

$$P(Y=k|X1,X2\ldots Xn) = \frac{P(X1|Y=k) * P(X2|Y=k)\ldots\ldots * P(Xn|Y=k) * P(Y=k)}{P(X1) * P(X2)\ldots * P(Xn)}$$

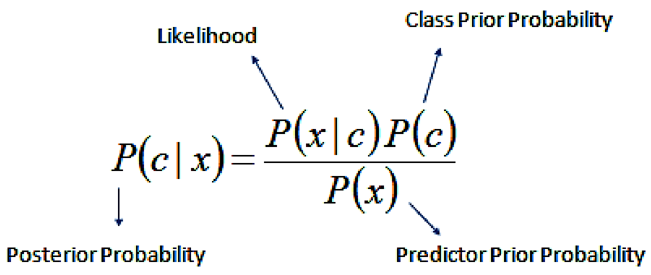
Which can be expressed as:

$$P(Y=k|X1,X2\ldots.Xn) = \frac{\mathrm{P(Y)}\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(Xi|Y)}}{P(X1) * P(X2)\ldots. * P(Xn)}$$

Since the denominator is constant here so we can remove it. It is purely your choice if you want to remove it or not. Removing the denominator will help you save time and calculations.

$$P(Y=k|X1,X2\ldots Xn) \propto \mathrm{P(Y)}\prod_{\mathrm{i}=1}^{\mathrm{n}}\mathrm{P(Xi|Y)}$$

This formula can also be understood as:



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

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

There are a whole lot of formulas mentioned here but worry not we will try to understand all this with the help of an example.

Decision Tree Using Information Gain

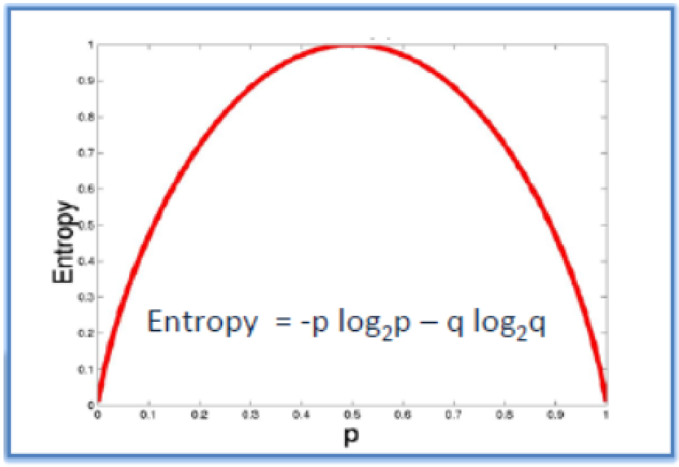
Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Algorithm

The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree. In Zero model there is no predictor, in One model we try to find the single best predictor, naive Bayesian includes all predictors using Bayes' rule and the independence assumptions between predictors, but decision tree includes all predictors with the dependence assumptions between predictors.

Entropy

A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogenous). ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.



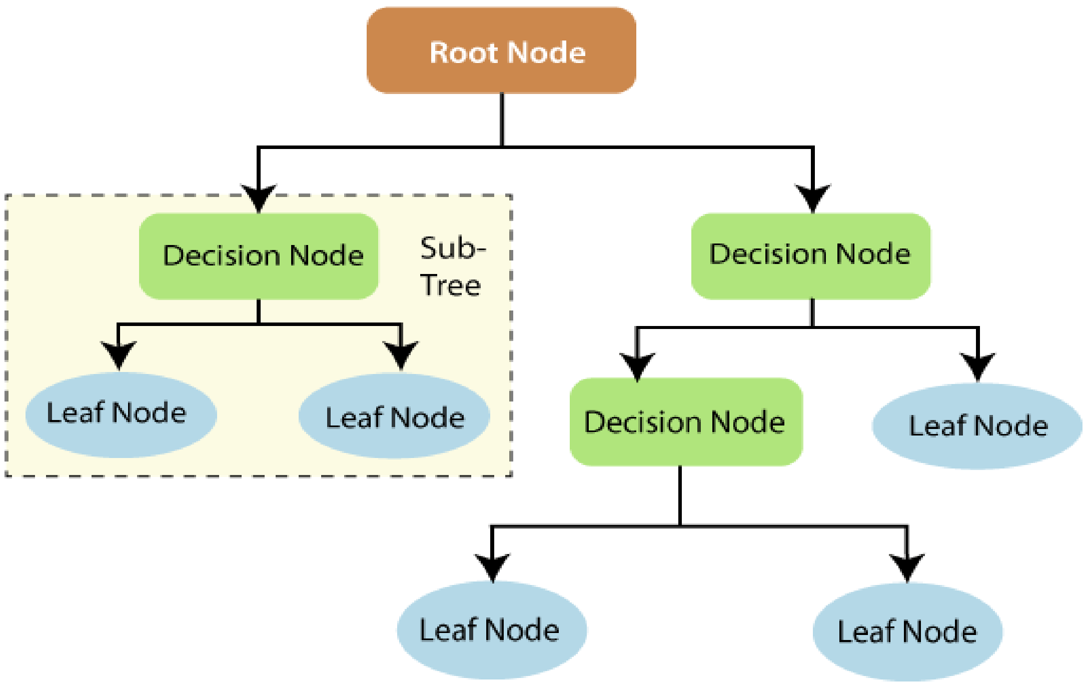
Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

RANDOM FOREST CLASSIFIER

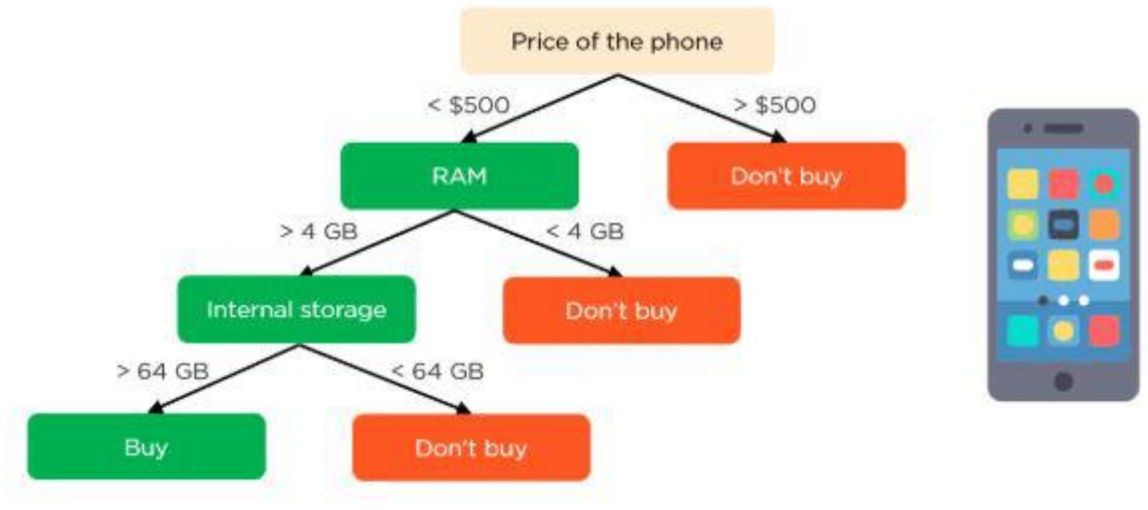
A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome.

How random forest algorithm works

Decision trees are the building blocks of a random forest algorithm. A decision tree is a decision support technique that forms a tree-like structure. An overview of decision trees will help us understand how random forest algorithms work. A decision tree consists of three components: decision nodes, leaf nodes, and a root node. A decision tree algorithm divides a training dataset into branches, which further segregate into other branches. This sequence continues until a leaf node is attained. The leaf node cannot be segregated further. The nodes in the decision tree represent attributes that are used for predicting the outcome. Decision nodes provide a link to the leaves. The following diagram shows the three types of nodes in a decision tree.



The information theory can provide more information on how decision trees work. Entropy and information gain are the building blocks of decision trees. An overview of these fundamental concepts will improve our understanding of how decision trees are built. Entropy is a metric for calculating uncertainty. Information gain is a measure of how uncertainty in the target variable is reduced, given a set of independent variables. The information gain concept involves using independent variables (features) to gain information about a target variable (class). The entropy of the target variable (Y) and the conditional entropy of Y (given X) are used to estimate the information gain. In this case, the conditional entropy is subtracted from the entropy of Y. Information gain is used in the training of decision trees. It helps in reducing uncertainty in these trees. A high information gain means that a high degree of uncertainty (information entropy) has been removed. Entropy and information gain are important in splitting branches, which is an important activity in the construction of decision trees. Let us take a simple example of how a decision tree works. Suppose we want to predict if a customer will purchase a mobile phone or not. The features of the phone form the basis of his decision. This analysis can be presented in a decision tree diagram. The root node and decision nodes of the decision represent the features of the phone mentioned above. The leaf node represents the final output, either *buying* or *not buying*. The main features that determine the choice include the price, internal storage, and Random Access Memory (RAM). The decision tree will appear as follows.



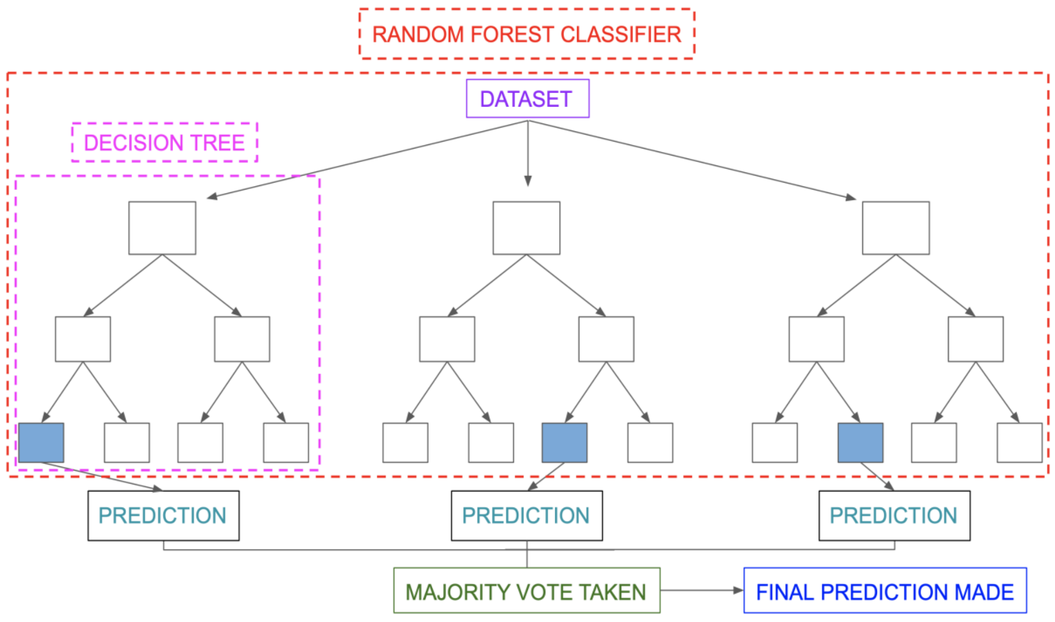
Applying decision trees in random forest

The main difference between the decision tree algorithm and the random forest algorithm is that establishing root nodes and segregating nodes is done randomly in the latter. The random forest employs the bagging method to generate the required prediction. Bagging involves using different samples of data (training data) rather than just one sample. A training dataset comprises observations and features that are used for making predictions. The decision trees produce different outputs, depending on the training data fed to the random forest algorithm. These outputs will be ranked, and the highest will be selected as the final output.

Our first example can still be used to explain how random forests work. Instead of having a single decision tree, the random forest will have many decision trees. Let us assume we have only four decision trees. In this case, the training data comprising the phone's observations and features will be divided into four root nodes. The root nodes could represent four features that could influence the customer's choice (price, internal storage, camera, and RAM). The random forest will split the nodes by selecting features randomly. The final prediction will be selected based on the outcome of the four trees. The outcome chosen by most decision trees will be the final choice. If three trees predict *buying*, and one tree predicts *not buying*, then the final prediction will be *buying*. In this case, it's predicted that the customer will buy the phone.

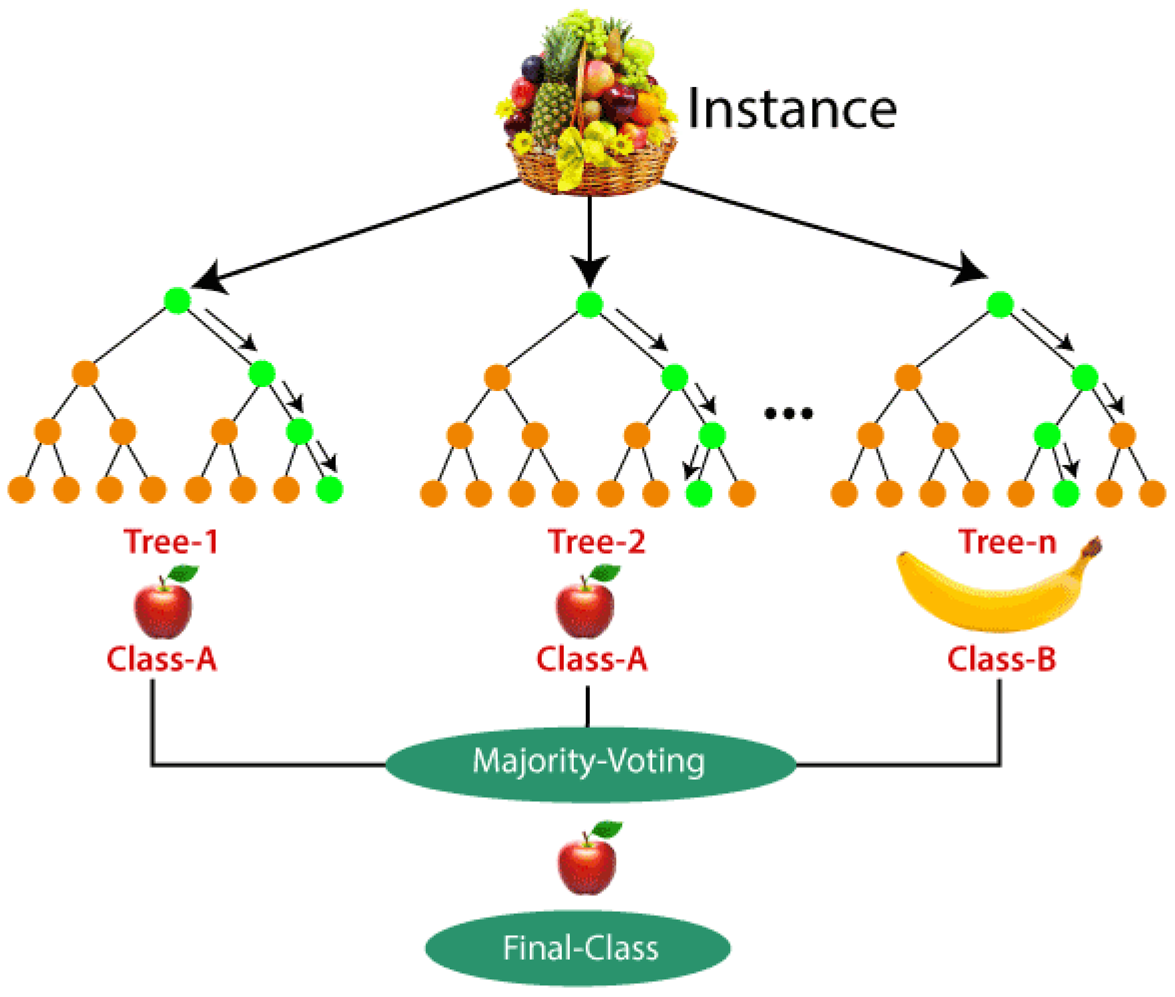
Classification in random forests

Classification in random forests employs an ensemble methodology to attain the outcome. The training data is fed to train various decision trees. This dataset consists of observations and features that will be selected randomly during the splitting of nodes. A rain forest system relies on various decision trees. Every decision tree consists of decision nodes, leaf nodes, and a root node. The leaf node of each tree is the final output produced by that specific decision tree. The selection of the final output follows the majority-voting system. In this case, the output chosen by the majority of the decision trees becomes the final output of the rain forest system. The diagram below shows a simple random forest classifier.



Let us take an example of a training dataset consisting of various fruits such as bananas, apples, pineapples, and mangoes. The random forest classifier divides this dataset into subsets. These subsets are given to every decision tree in the random forest system. Each decision tree produces its specific output. For example, the prediction for trees 1 and 2 is *apple*.

Another decision tree (n) has predicted *banana* as the outcome. The random forest classifier collects the majority voting to provide the final prediction. The majority of the decision trees have chosen *apple* as their prediction. This makes the classifier choose *apple* as the final prediction.


Instance
Tree-1
Tree-2
Tree-n
Class-A
Class-A
Class-B
Majority-Voting
Final-Class

Result

In results section, we have created this following table by running the below mentioned classifiers.

		Accuracy	Precision	Recall	F1 score	Sensitivity	Specificity	Error rate	TPR	FPR
1.	Logistic regression with regularization	0.906	0.9191	0.9795	0.9483	0.9795	0.3721	0.0937	0.3768	0.104
2.	KNeighborsClassifier	0.87	0.876	0.894	0.863	0.879	0.8756	0.123	0.82	0.18
3.	Linear SVM	0.99	0.99	1.0	0.9998	0.997	0.997	0.001	0.99	0.01435
4.	Naive Bayes	0.84	0.924	0.8932	0.4506	0.8932	0.4506	0.1587	0.5493	0.1067
5.	Bernoulli Naïve Bayes	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0
6.	Decision Tree (Gini index)	1.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0
7.	Random Forest	0.88	0.88	0.89	0.93	0.12	0.88	0.117	0.88	0.12

\

Confusion Matrix:

KNN:

```
array ([[7661, 319], [ 794, 269]])
```

Random Forest

```
array ([[7980, 0], [1063, 0]])
```

Naïve Bayes

```
array ([[7128, 852], [ 584, 479]])
```

Bernoulli’s Naïve Bayes

```
array([[9994, 0], [ 0, 1306]])
```

Linear SVM

```
array([[9994, 0], [ 3, 1306]])
```

Logistic Regression with Regularisation

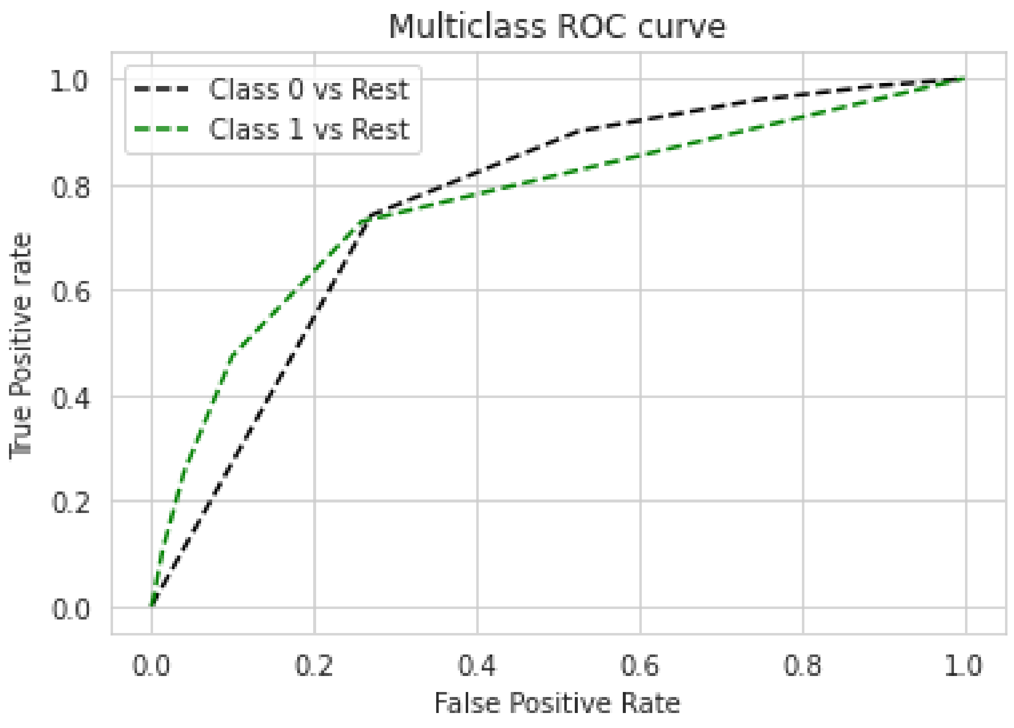
```
array([[7789, 163], [ 685, 406]])
```

Decision Tree using Gini Index

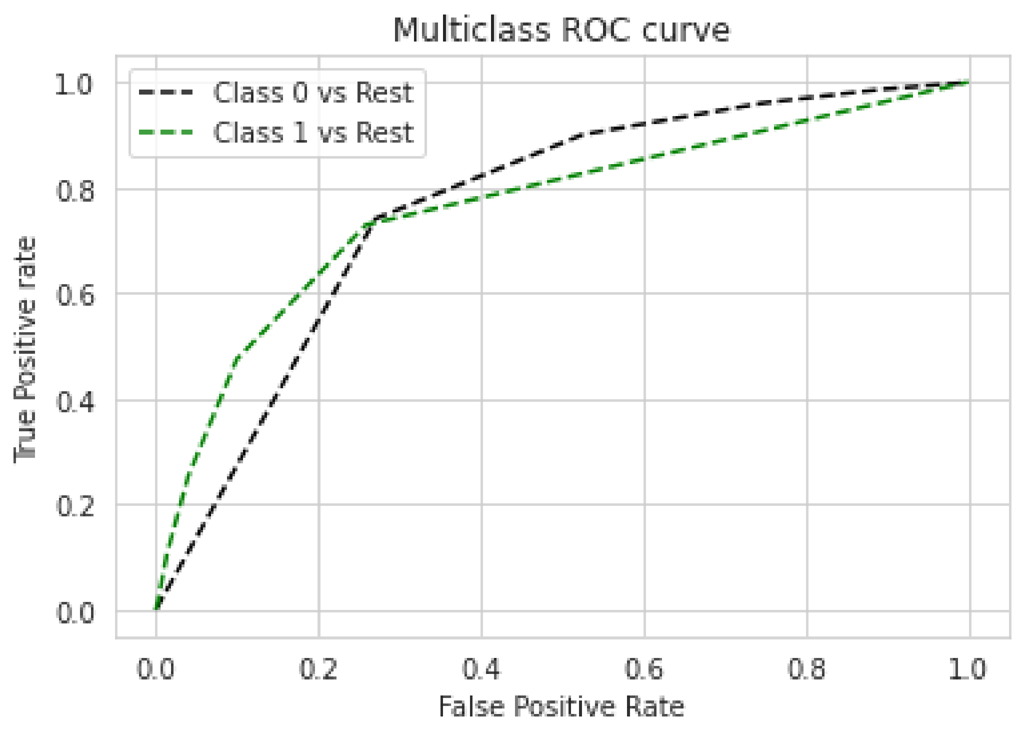
```
array([[7331, 649], [ 556, 507]])
```

ROC Curves:

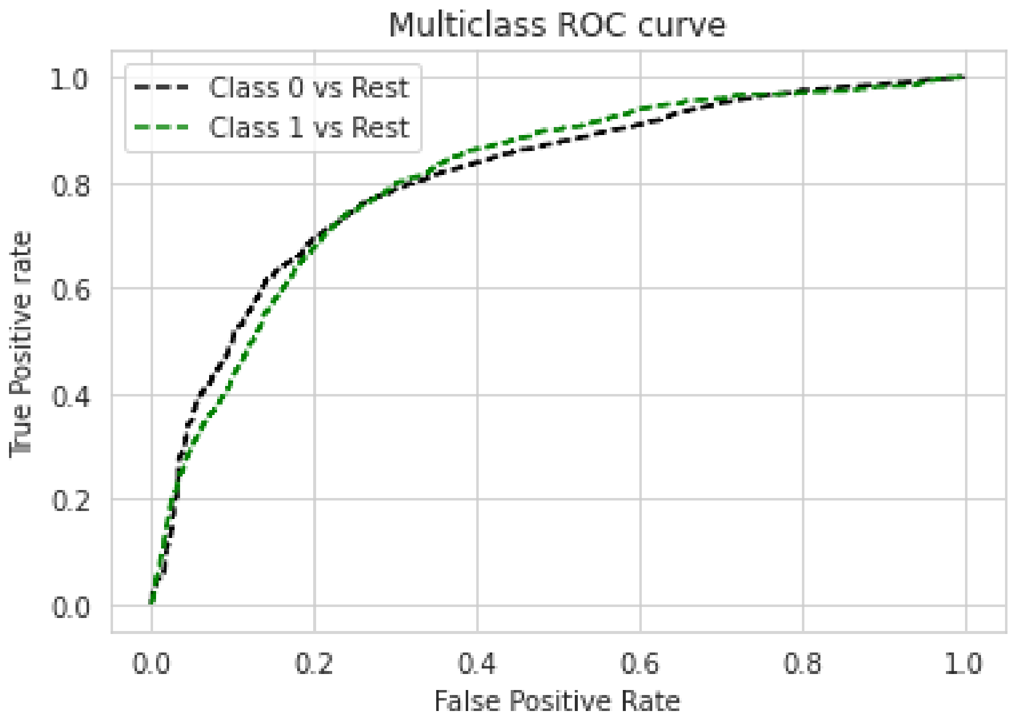
KNN:



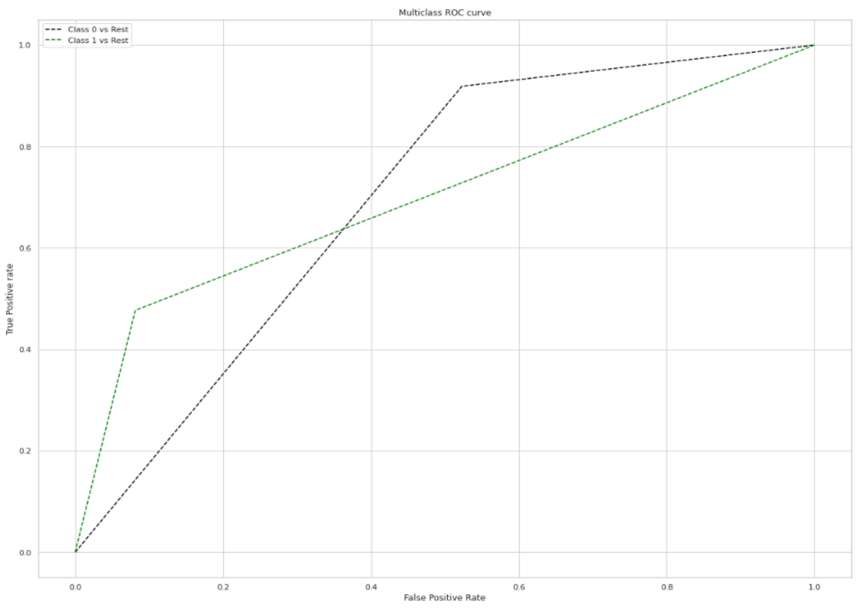
Random Forest:


Multiclass ROC curve
Class 0 vs Rest
Class 1 vs Rest
True Positive rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

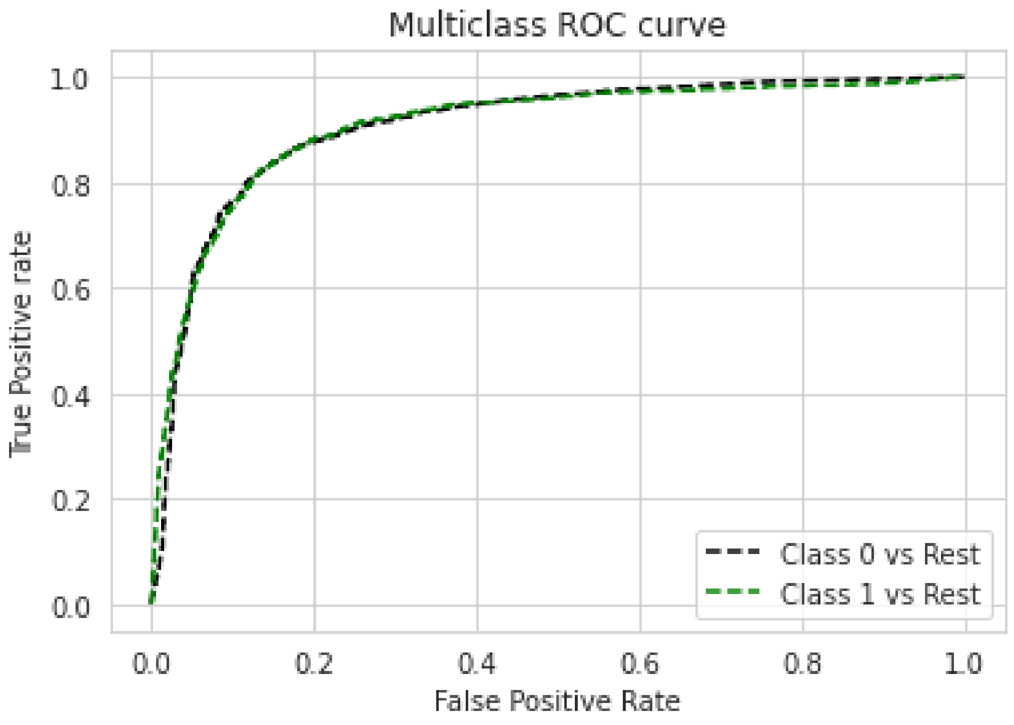
Naïve Bayes:



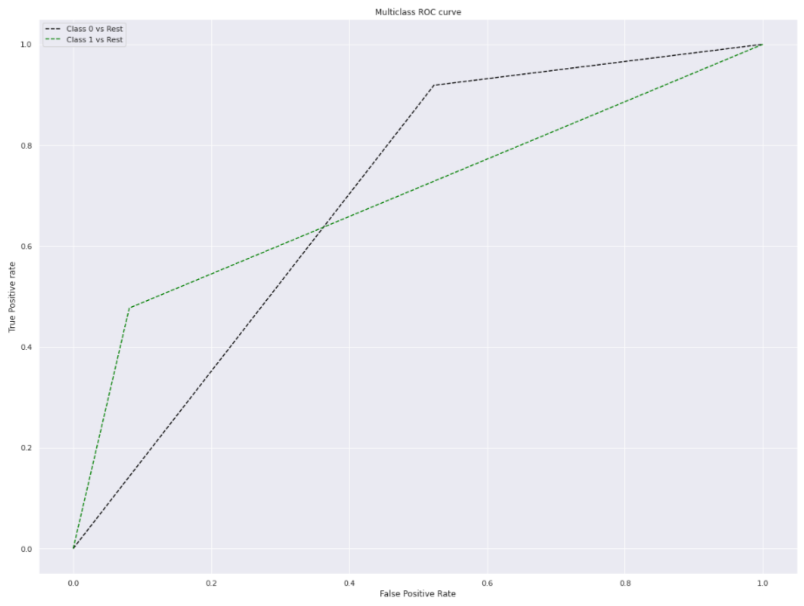
Bernoulli Naïve Bayes:



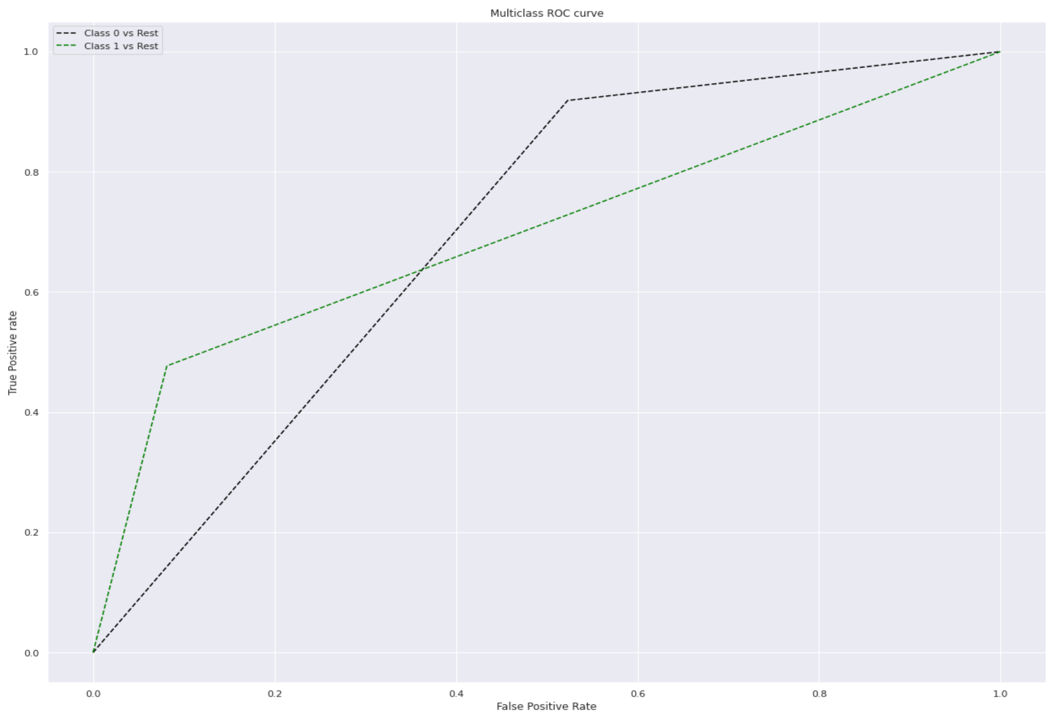
Logistic Regression with Regularisation:



Decision Tree using Gini Index:



SVM



CONCLUSION

The major purpose of this study was to determine whether the client will subscribe to bank or not. Many different algorithms were applied to predict the result, but the algorithm that most appropriately predicted the correct values is linear SVM. Unlike logistic regression and other algorithms, SVMs are designed to generate more complex decision boundaries. An LS-SVM with a simple linear kernel corresponds to a linear decision boundary. Instead of a linear kernel, more complex kernel functions, such as the commonly used RBF kernel, can be chosen. The reason why SVMs work well with high dimensional data is that they are automatically regularized, and regularization is a way to prevent overfitting with high dimensional data. The accuracy of the model using linear SVM is approximately 87%, which means that if a student answers all the questions about the instructor and courses then this model may predict the attendance of the student. However, a word of caution should be noted. The data in this study were gathered from just one course in one university. So, we are not generalizing the results to other courses, lecturers, or universities across the nation. Plus, the accuracy of the data provided by the respondents depends on their honesty as well as their understanding of the questions asked. The respondents' skills of evaluating based on the criteria are different which is due to their personal judgment of the scale used in the questions. Lastly, limited time and cost are the constraint faced when conducting this study where only two classes are selected and thus, the result may not as accurate and reliable.

References

UCI Machine Learning Repository: Bank Marketing Data Set

https://www.hindawi.com/journals/cin/2022/4151487/

https://link.springer.com/article/10.1007/BF00991617

https://scholar.google.com/scholar_lookup?title=Faculty%20ratings%20and%20student%20grades%3A%20A%20university-wide%20multiple%20regression%20analysis&journal=Journal%20of%20Educational%20Psychology&volume=68&pages=573-577&publication_year=1976&author=Brown%2CD.%20L.

https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

https://towardsdatascience.com/logistic-regression-using-python-sklearn-numpy-mnist-handwriting-recognition-matplotlib-a6b31e2b166a