PREDICTIVE MODELLING FOR CUSTOMER CHURN
USING MACHINE LEARNING APPROACHES

Abstract :----------

Customer churn analysis and prediction in industries is an issue now a days because it's very important for industries to analyze behavior of various customers to predict which customers are about to leave the subscription from the company. So machine learning techniques and algorithm plays an important role for companies in today's commercial conditions because gaining a new customer's cost is more than retaining the existing ones. This project focuses on various machine learning techniques for predicting customer churn through which we can build the classification models such as Logistic Regression, Decision Tree, Random Forest, K-Nearest-neighbors, Support Vector Machine and also compare the performance of these models.

Keywords – churn, machine learning, Logistic Regression, Decision Tree, Random Forest, K-Nearest-neighbors, Support Vector Machine

Introduction:----------

Churn prediction is predicting which customers are at high risk of leaving your company or cancelling a subscription to a service, based on their behavior with the product. To predict churn effectively, one has to synthesize and utilize key indicators defined by the team to signal when a customer has a probability of churning so that the company can take action. The goal of churn prediction is to be able to answer questions like "Will [X] customer leave the company in X months?" or "Will [X] customer renew their subscription?" and also to understand greater trends in churn.

According to a study done by McKinsey, technology and SaaS companies with the highest performance and revenue growth were also companies with high retention rates and low net revenue churn. The ability to predict churn before it happens allows business to take proactive actions to keep existing customers from churning. When predicting churn, one is not just identifying at-risk customers, but also identifying pain points leading up to churn and helping to increase overall customer retention and satisfaction. Preventing churn, in turn, serves as a great revenue source for business.

Data Set Description:----------

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, in order to access if the product (bank term deposit) would be (or not) subscribed.

The data set contains 4521 rows and 17 columns. Every row of the data set indicates a data point and every column indicates a different variable. There are 16 independent variables and 1 dependent variable which is our target variable. The target variable is dichotomous in nature.

The description of all the variables are given below:

Input variables:

1 - age (numeric)

2 - job : type of job (categorical "admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar", "self-employed", "retired", "technician", "services")

3 - marital : marital status (categorical: "married", "divorced", "single"; note: "divorced" means divorced or widowed)

4 - education (categorical: "unknown", "secondary", "primary", "tertiary")

5 - default: has credit in default? (binary: "yes", "no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes", "no")

8 - loan: has personal loan? (binary: "yes", "no")

related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: "unknown", "telephone", "cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")

12 - duration: last contact duration, in seconds (numeric)

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown", "other", "failure", "success")

Output variable (desired target):

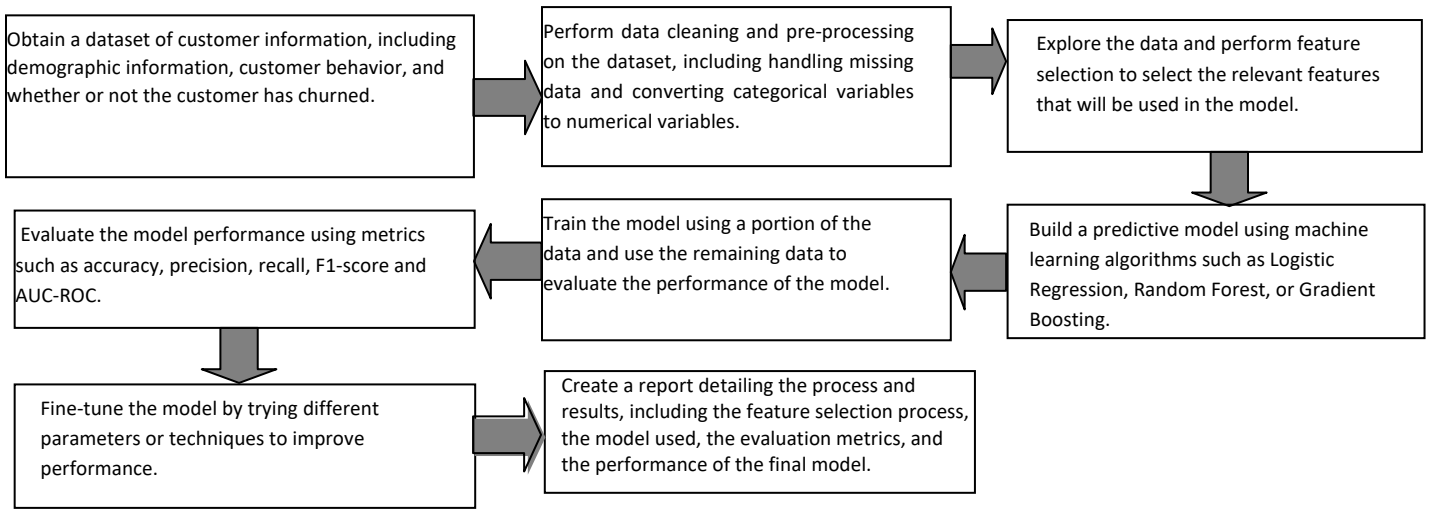
17 - y - has the client subscribed a term deposit? (binary: "yes", "no")

Problem Statement:----------

The objective of this assignment is to build a predictive model that can predict customer churn for a given company. I implemented several machine learning techniques to build the model and document the process, including feature selection, model evaluation, and performance metrics.

Methodology:----------

Instead of writing to much in the methodology part I will try to give you a very brief idea about the steps which I followed in a respective manner through a flow chart and after that I will explain all the steps thoroughly .
The flow chart is given below –



Machine Learning Methods:----------

Logistic Regression : This type of statistical model (also known as *logit model*) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

$$Logit\ (p_i) = \frac{1}{(1 + e^{-p_i})}$$

$$\ln\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \sum_{j=1}^{k} \beta_j * x_j$$

Where p_i is the probability of occurrence of an event, x_i denote the ith independent variable, β_0 is the intercept term of the linear path and β_i's are the coefficient corresponding to the ith independent variable. There are three types of logistic regression : 1) Binary logistic regression, 2) Multinomial logistic regression and 3) Ordinal logistic regression.

In python we can implement logistic regression from the sklearn library. From sklearn we need to extract linear_model and within this module we can get the LogisticRegression() function.

Min-Max Normalization : Before modelling, variables that are measured at different scales do not contribute equally to the model fitting and model learned function and might end up creating a bias. Thus to deal with this problem feature-wise or variable-wise normalization is usually used before the model fitting. Min-Max scaling is one of the technique used for normalization. By applying this Min-Max normalization technique, all features will be transformed into the range [0,1] meaning that the minimum and maximum value of a variable will be . and 1 respectively. The mathematical formulation of Min-Max scaling is given below:

$$Variable_{scaled} = \frac{x\ -\ \min\ (x)}{\max(x)\ -\ \min\ (x)}$$

In python we can implement this Min-Max normalization from sklearn library. From sklearn we need to extract preprocessing and within this module we can get the MinMaxScaler() function.

Decision Tree Classifier : In general, Decision tree analysis is a predictive modeling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decisions tress are the most powerful algorithms that falls under the category of supervised algorithms. They can be used for both classification and regression tasks. The two main

entities of a tree are decision nodes, where the data is split and leaves, where we got outcome. We have the following two types of decision trees −

1) Classification decision trees − In this kind of decision trees, the decision variable is categorical. Our project is an example of classification decision tree.

2) Regression decision trees − In this kind of decision trees, the decision variable is continuous.

In python we can implement this Decision Tree from sklearn library. From sklearn we need to extract tree and within this module we can get the DecisionTreeClassifier() function.

Random Forest Classifier : Random Forest is a Supervised Machine Learning algorithm that is used widely in classification and Regression problems. It builds decision trees on different samples and take their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest algorithm is that it can handle data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Random Forest is an ensemble bagging classifier, which means it creates a different training subset from sample training data with replacement and final output is based on the majority voting.

In python we can implement this Random Forest from sklearn library. From sklearn we need to extract ensemble and within this module we can get the RandomForestRegressor() function.

K-Nearest Neighbour : 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. K-NN 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. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

In python we can implement this KNN from sklearn library. From sklearn we need to extract neighbors and within this module we can get the KNeighborsClassifier() function.

Support Vector Machine (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.

In python we can implement this SVM from sklearn library. From sklearn we need to extract svm and within this module we can get the SVC() function.

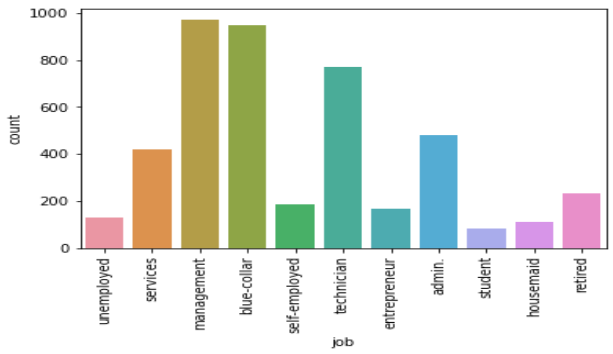
Methodology Explanation:----------

1. **Collection of data** : The data set was obtained from Kaggle and its link is provided here - https://www.kaggle.com/competitions/bank-marketing-uci/overview

2. **Data cleaning and pre-processing** : After reading the dataset in csv format, the categorical variables containing the category label as 'unknown' was replaced by 'NaN' value. After replacing the unknown values, it is observed that the variable 'poutcome' has 3705 NaN values out of 4521 values which is approximately 82% of the data. So, the column 'poutcome' has been dropped from our dataset.

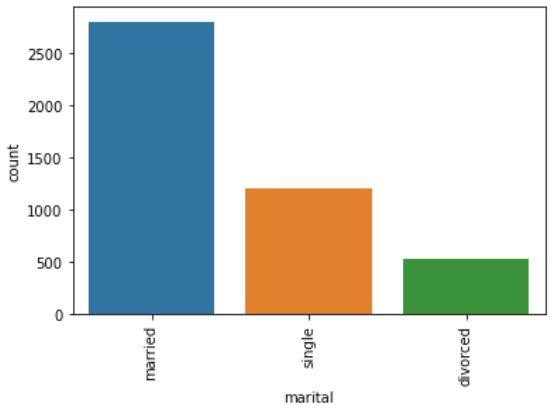
3. **Data Visualization** : To get an idea about the distribution of the categorical variables, bar plots were constructed and a brief interpretation of each plot is given below to understand the data:

a)



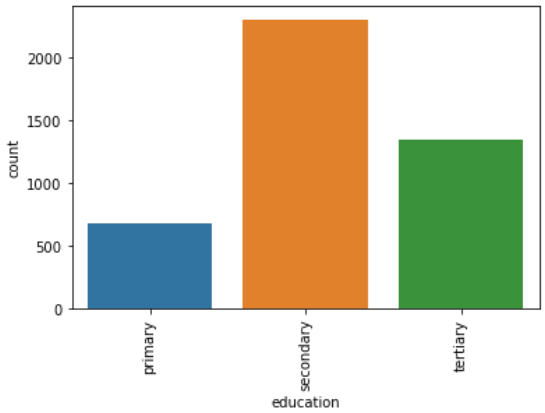
From the plot we can see that most of the targeted customers belong to the 'management' job category, i.e, among all the job categories, the job category 'management' has the highest count followed by 'blue-collar', 'technician' and many more.

b)



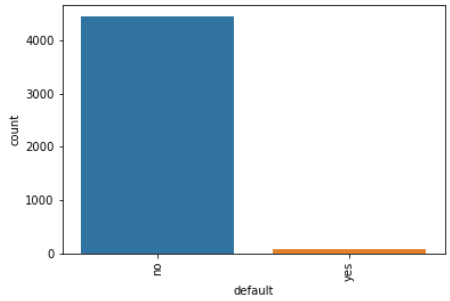
From the plot we can see that most of the targeted customers of this campaign are married.

c)



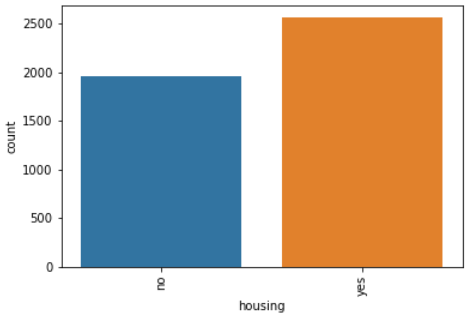
From the plot we can see that most of the targeted customers of this campaign has secondary education.

d)



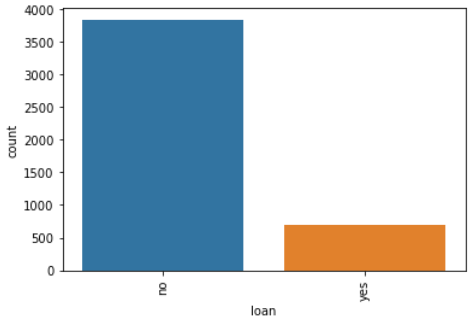
From the plot we can see that more than 95% of the targeted customers of this campaign are non-defaulter.

e)



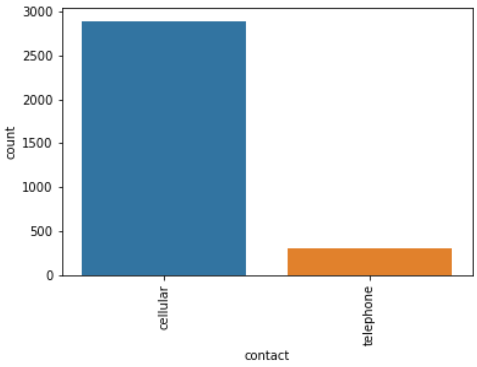
From the plot we can see that most of the targeted customers of this campaign have a housing loan.

f)



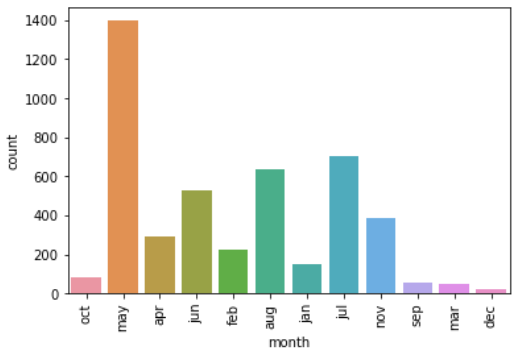
From the plot we can see that more than 85% of the targeted customers of this campaign have a personal loan.

g)



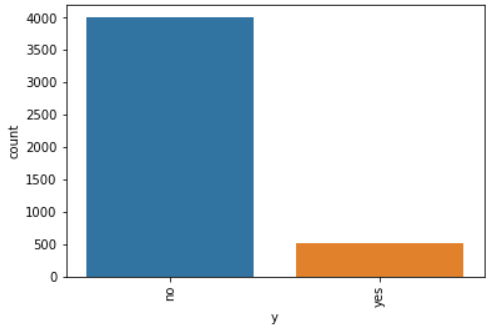
From the plot we can see that more than 90% of the targeted customers of this campaign have a cellular phone.

h)



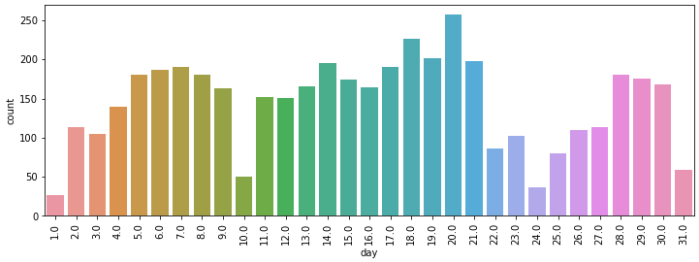
From the plot we can see that the last contacted month of the year for most of the targeted customers of the campaign is the month of May.

i)



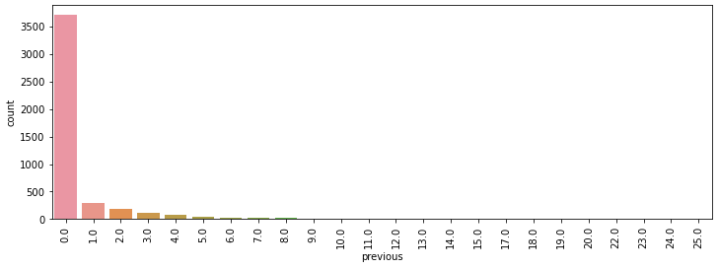
From the plot we can see that more than 85% of the targeted customers of this campaign have not subscribed to a term deposit.

j)



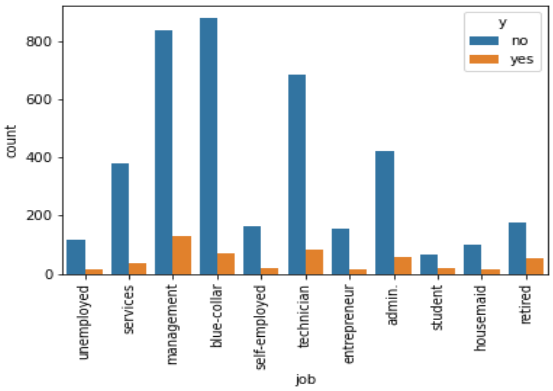
From the plot we can see that the last contact day of a certain month for most of the customers is on 20th followed by 18th and so on.

k)



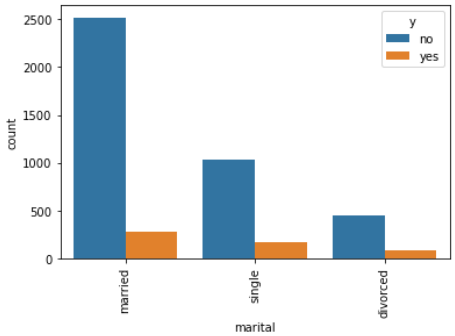
From the plot we can see that the number of contacts performed before this campaign for more than 90% of the targeted customers is 0.

l)

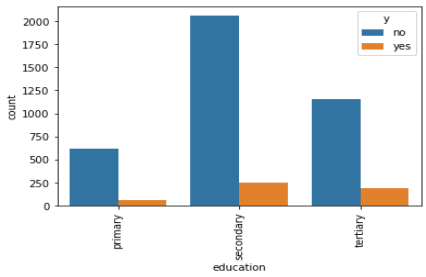


The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘job’. From the plot we can see that majority of the targeted customers of this campaign with the job category ‘management’ have subscribed to the term deposit whereas the job category ‘blue-collar’ have not subscribed to the term deposit.

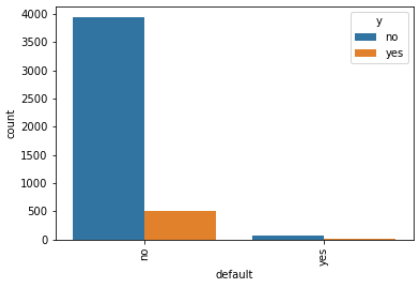
m)



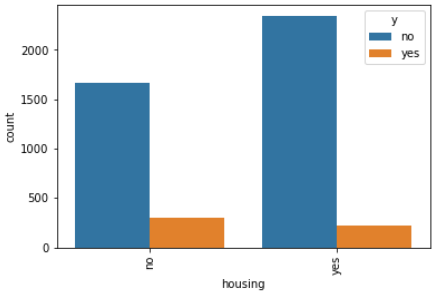
The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘marital’. From the plot we can see that most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit is married.

n) 

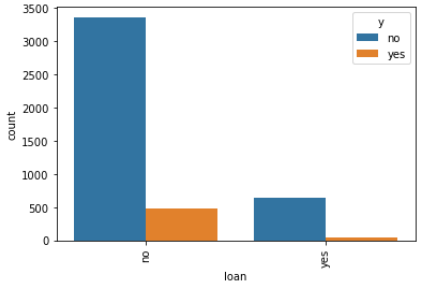
The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘education’. From the plot we can see that most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit has secondary education.

o) 

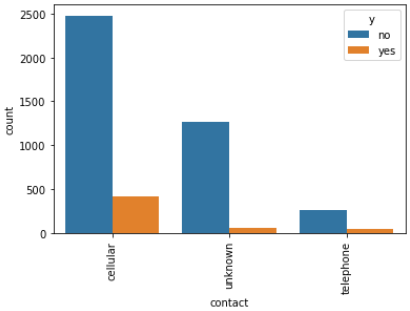
The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘default’. From the plot we can see that most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit are non-defaulter.

p) 

The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘housing’. From the plot we can see that most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit do not have a housing loan.

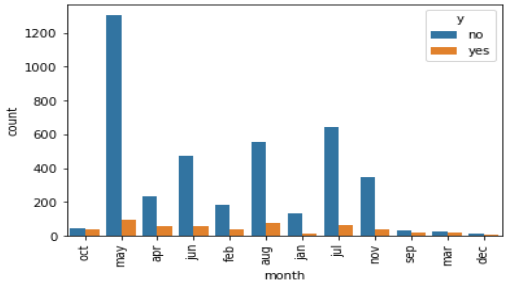
q) 

The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘loan’. From the plot we can see that most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit do not have a personal loan.

r) 

The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘contact’. From the plot we can see that most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit use cellular phone for communication.

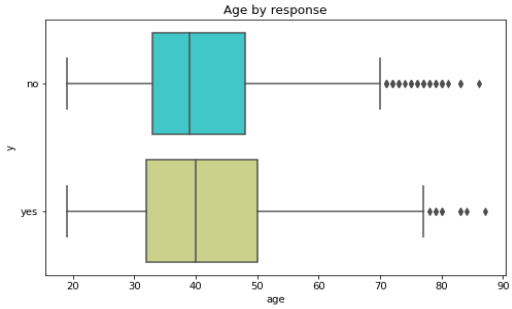
s)



The plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent categorical variable ‘month’. From the plot we can see that the last contact month of the year for most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit is the month of May.

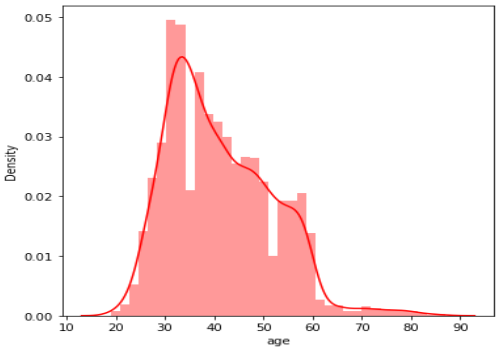
To get an idea about the distribution of the continuous variable ‘age’ with respect to the response variable, a box plot was constructed as shown below:

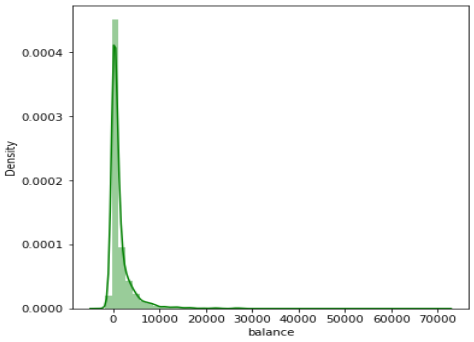
t)



The box plot shows the relation between the dependent variable ‘y’(has the client subscribed a term deposit? (binary: "yes","no")) and the independent continuous variable ‘age’. From the plot we can see that the median age for most of the targeted customers of this campaign who subscribed as well as who did not subscribe to the term deposit is more or less the same however the spread of age is different for both the categories of the response variable.

To get an idea about the distribution of the continuous variable ‘age’ and ‘balance’, density plots were constructed as shown below:



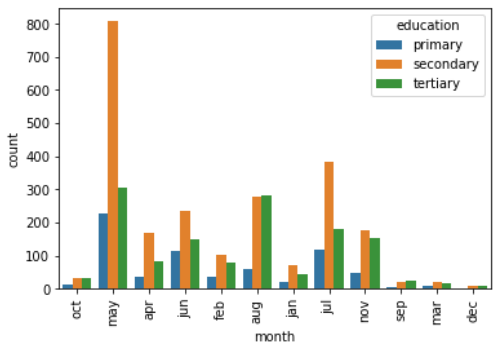


4) **Missing Value Imputation** : Three variables in our dataset had missing values namely 'job' (38), 'education' (187) and 'contact' (1324). To impute these missing values we first studied the association between all the categorical variables and the associations at 1% level of significance are listed below:

```
job and marital are dependent with association : 0.2023430333164971
job and education are dependent with association : 0.5532227925939571
job and housing are dependent with association : 0.2735307445109201
job and loan are dependent with association : 0.0970347953658868
job and contact are dependent with association : 0.14945441093110198
job and month are dependent with association : 0.12781373752240344
job and y are dependent with association : 0.12270504870947899
marital and education are dependent with association : 0.12632456095197642
marital and default are dependent with association : 0.05226049242263998
marital and loan are dependent with association : 0.049057347601134546
marital and contact are dependent with association : 0.057975681531135005
marital and month are dependent with association : 0.07649734883367068
marital and y are dependent with association : 0.06487879626537206
education and housing are dependent with association : 0.11568258170028019
education and loan are dependent with association : 0.06659761433431256
education and contact are dependent with association : 0.12850419986777173
education and month are dependent with association : 0.14080851438563646
education and y are dependent with association : 0.05846663956958666
default and loan are dependent with association : 0.06399394536288765
housing and month are dependent with association : 0.49018537762827885
housing and y are dependent with association : 0.10468340035106324
loan and month are dependent with association : 0.1857524933051144
loan and y are dependent with association : 0.07051703515462056
contact and month are dependent with association : 0.14868816694481723
month and y are dependent with association : 0.23538927243938482
```

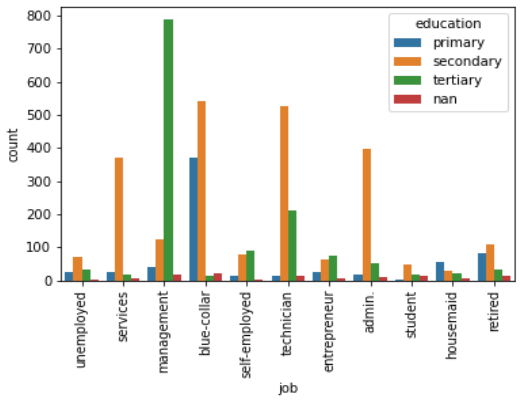
a) **Imputation of the missing values in the 'contact' variable :** Based on the count plot of the 'contact' variable as shown above in 3(g), we can see that more than 90% of the targeted customers of this campaign have a cellular phone. So, the missing values were imputed with the mode value, which is 'cellular' according to our data.

b) **Imputation of the missing values in the 'education' variable :** As it is seen that an association exists between the 'month' and 'education' variable, a multiple bar chart was plotted based on these two variables which is shown below –



From the plot it is observed that most of the targeted customers who have been last contacted in the months of May, April, June, February, July, November, March and January have 'secondary' education degree. So, we imputed the 'education' variable by the category label 'secondary' based on the above mentioned months and the remaining months by the category label 'tertiary'.

c) **Imputation of the missing values in the 'job' variable :** As it is seen that an association exists between the 'job' and 'education' variable, a multiple bar chart was plotted based on these two variables which is shown below –



From the plot it is observed that most of the targeted customers who have 'tertiary' education degree belong to the job category 'management'. So, we imputed the 'job' variable with the category label 'management' for the customers having 'tertiary' education degree and for the remaining customers we imputed the 'job' variable with the category label 'blue-collar'.

5) **Multicollinearity checking and variable selection :** To check the presence of multicollinearity, Variance Inflation Factor (VIF) was used. We know that if for any variable VIF < 1, then there is no multicollinearity present, if 1<=VIF< 5, then there is moderate multicollinearity and if VIF > =5, then there is high multicollinearity. Based on this, we found that 'age', 'day', 'marital_married' and 'month_may' had high multicollinearity with the response variable 'y'. So we dropped these variables from our data set.

To choose the most important variables that will describe our target variable in a better way, we used ExtraTreesClassifier from sklearn.ensemble library. Based on the outcome, we haven chosen top 8 variables to build our model and they are : 'duration', 'balance', 'campaign', 'pdays', 'previous', 'marital_single', 'month_oct', 'housing_yes'.

6) **Creating dummy variable to convert categorical data to numerical data :** In our dataset there are 9 columns (independent variables) which are categorical. To build any machine learning model we need to convert these categorical columns to numerical columns. So, we need to create dummy variables which are linearly independent. For example, there are two categories in the 'housing' variable that are 'yes' and 'no'. Then the dummy variable considers 'yes' as 1 and 'no' as 0. In the same way the remaining 8 categorical columns are converted to numerical columns.

7) **Split the training data into two parts :** Now after performing all that has been stated in the previous steps, our data set is ready to fit machine learning model. But before building a machine learning model using this data we need to split the data set into two mutually exclusive and exhaustive sets, first part is for training the model and the second part is for validating the trained model. However this data set cannot be split based on our preference instead it should be done randomly. Python environment helps us to solve this problem. In python we need to import train_test_split from the sklearn.model_selection library. In train_test_split there are two very important arguments which are random_state and test_size. In test_size argument we need to pass the percentage of split for the training and validation data set, for example in our project we pass test_size as 0.30, which means that 70% of our data is for building the machine learning model and the remaining 30% data is for validating the built model. Random_state helps us to give a random combination of our data set in order to avoid any sort of over representation or under representation of the class labels of the dependent variable.

8) **Training the different machine learning models and validating them :** In this project, the various machine learning models that have been used are Logistic Regression, Decision Tree, Random Forest, K-Nearest-neighbors and Support Vector Machine. These models have been used to train our data set and then

calculated the evaluation metrices like accuracy score, precision, f1-score, recall, AUC and ROC to evaluate the performances of all the models. All the evaluation metrics result for various models that have been used are depicted in the following table after tuning the hyperparameters of each model:

MODELS	Accuracy Score	Precision	Recall	F1-score
Logistic Regression	0.890198	0.589743	0.1474358	0.235897
Decision Tree	0.883566	0.4875	0.25	0.330508
Random Forest	0.866617	0.406015	0.346153	0.373024
Support Vector Machine	0.8887251	0.571428	0.128205	0.209424
K-Nearest Neighbors	0.8747236	0.428571	0.269230	0.330708

Based on the above table we can see that the accuracy score for the logistic regression is the highest among the other machine learning models.

Conclusion :----------

We can safely conclude that among various machine learning models, Logistic Regression can be applied on the test data set to predict the customer churn, that is, to predict whether a particular customer will subscribe to a term deposit or not.

Future Work :----------

1) In future, if we get any bank related dataset having similar variables as compared to this project, and if we have to predict the customer churn, then we can apply this logistic regression model that has already been built on our data set
2) Here we have used 5 machine learning models but many more advanced machine learning models can be applied by tuning the hyperparameters to get a better accuracy.
3) In our project we used Min-Max normalization on the variables before model building, but we can also use several other normalization methods to build the model and get better result.