Final Project Report

Bank Marketing Campaign

- Data Science -

Table of Contents

Team member's details:

	Group Name: *Data Science Enthusiasts*				
	Name	**Email**	**Country**	**College/Company**	**Specialization**
1	Amira Asta	amira.asta02@gmail.com	Tunisia	Afrikanda	Data Science
2	Vatsal Vinesh Mandalia	vatsalvm10@outlook.com	Oman	Graduated	Data Science

Problem description:

ABC Bank wants to sell its term deposit product to customers and before launching the product they want to develop a model which helps them understand whether a particular customer will buy their product or not. In order to achieve this task, they approached an Analytics company to automate this process of classification. The Analytics company has given responsibility to the **Data Science Enthusiasts** Team and has asked to come up with a ML model to shortlist customers whose chances of buying the product is higher, so that ABC's marketing channel can focus only on those customers.

Business understanding:

There has been a revenue decline for an ABC bank and they would like to know what actions to take. After investigation, they found out that the root cause is that their clients are not depositing as frequently as before. Knowing that term deposits allow banks to hold onto a deposit for a specific amount of time, banks can invest in higher gain financial products to make a profit.

In addition, banks also hold better chances to persuade term deposit clients into buying other products such as funds or insurance to further increase their revenues. As a result, the ABC bank would like to identify existing clients that have higher chances to subscribe for a term deposit and focus marketing efforts on such clients. The classification goal is to predict if the client will subscribe to a term deposit or not.

Data Intake report:

Name: Bank Marketing Campaign - Data Science

Report date: August 7th, 2021

Internship Batch: LISUM01

Version: 1.0

Data intake by: Data Science Enthusiasts Team

Data intake reviewer: Vatsal Vinesh Mandalia

Data storage location:
https://github.com/AsAmira02/Bank-Marketing-Campaign-DSEnthusiasts2021

Tabular data details: 'bank.csv'

Total number of observations	4521
Total number of files	1
Total number of features	17
Base format of the file	.csv
Size of the data	461 KB

Tabular data details: 'bank-full.csv'

Total number of observations	45211
Total number of files	1
Total number of features	17
Base format of the file	.csv
Size of the data	4.61 MB

Tabular data details: ‘bank-additional.csv’

Total number of observations	4119
Total number of files	1
Total number of features	21
Base format of the file	.csv
Size of the data	584 KB

Tabular data details: ‘bank-additional-full.csv’

Total number of observations	41118
Total number of files	1
Total number of features	21
Base format of the file	.csv
Size of the data	4.61 MB

Proposed Approach:

- There are 12 rows of duplicated data in the ‘bank-additional-full’ dataset.
- There are no missing values in all datasets.

Data Understanding:

The data corresponds to 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 assess if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

There are four datasets provided for this classification problem. Among the four datasets, there are two pairs of train and test data available for analysis. The ‘bank-full.csv’ and ‘bank.csv’ are one of the pairs having less than 20 input features and are an older version of ‘bank-additional-full.csv’ and ‘bank-additional.csv’. The information of the datasets is given below.

	Dataset type	Description
bank-additional-full.csv	train	41118 observations and 20 inputs ordered by date (from May 2008 to November 2010)
bank-additional.csv	test	4118 observations (10% of train data) with 20 inputs
bank-full.csv	train	45211 observations and 17 inputs ordered by date (older version of bank-additional-full)
bank.csv	test	4521 observations (10% of train data) and 17 inputs

Data set features description:

Table 1: Dataset input features.

N°	Feature name	Description	Type
1	age	age	numeric
2	job	type of job	categorical
3	marital	marital status	categorical
4	education	level of education	categorical
5	default	has credit in default?	categorical
6	housing	has housing loan?	categorical
7	loan	has personal loan?	categorical
8	contact	contact communication type	categorical
9	month	last contact month of year	categorical
10	day of week	last contact day of the week	categorical
11	duration	last contact duration, in seconds	numeric
12	campaign	number of contacts performed in this campaign	numeric
13	pdays	number of days passed by after the last contact	numeric
14	previous	number of contacts performed for this client	numeric
15	poutcome	outcome of the previous marketing campaign	categorical
16	emp.var.rate	employment variation rate	numeric
17	cons.price.idx	consumer price index - monthly indicator	numeric
18	cons.conf.idx	consumer confidence index - monthly indicator	numeric
19	euribor3m	euribor 3 month rate - daily indicator	numeric
20	nr.employed	number of employees - quarterly indicator	numeric
21	y	has the client subscribed a term deposit?	binary

The first 7 features are Bank client data.

Features 8, 9, 10 and 11 are related to the last contact of the current campaign.

Features 12, 13, 14 and 15 are other attributes.

Features 16, 17, 18, 19 and 20 are social and economic context attributes.

The last feature is the output feature (desired target).

Data Types

In this dataset the features that we described above, are divided between "object" types i.e. categorical attributes and "int64 / float64" types i.e. numerical attributes.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41188 non-null  int64
 1   job             41188 non-null  object
 2   marital         41188 non-null  object
 3   education       41188 non-null  object
 4   default         41188 non-null  object
 5   housing         41188 non-null  object
 6   loan            41188 non-null  object
 7   contact         41188 non-null  object
 8   month           41188 non-null  object
 9   day_of_week     41188 non-null  object
 10  duration        41188 non-null  int64
 11  campaign        41188 non-null  int64
 12  pdays           41188 non-null  int64
 13  previous        41188 non-null  int64
 14  poutcome        41188 non-null  object
 15  emp.var.rate    41188 non-null  float64
 16  cons.price.idx  41188 non-null  float64
 17  cons.conf.idx   41188 non-null  float64
 18  euribor3m       41188 non-null  float64
 19  nr.employed     41188 non-null  float64
 20  y               41188 non-null  object
dtypes: float64(5), int64(5), object(11)
memory usage: 6.6+ MB
```

Data Problems:

Missing Attribute Values:

All 4 datasets have no missing values.

Note: There are a significant number of observations/rows with a value 'unknown' for the majority of the categorical features in the four datasets. So we assume the value 'unknown' as another category for these variables in our analysis.

```
admin.           10422
blue-collar       9254
technician        6743
services          3969
management        2924
retired           1720
entrepreneur      1456
self-employed     1421
housemaid         1060
unemployed        1014
student            875
unknown            330
Name: job, dtype: int64
```

```
married     24928
single      11568
divorced     4612
unknown        80
Name: marital, dtype: int64

university.degree      12168
high.school             9515
basic.9y                6045
professional.course     5243
basic.4y                4176
basic.6y                2292
unknown                 1731
illiterate                18
Name: education, dtype: int64
```

```
no         32588
unknown     8597
yes            3
Name: default, dtype: int64
yes        21576
no         18622
unknown      990
Name: housing, dtype: int64
no         33950
yes         6248
unknown      990
Name: loan, dtype: int64
```

Duplicate rows:

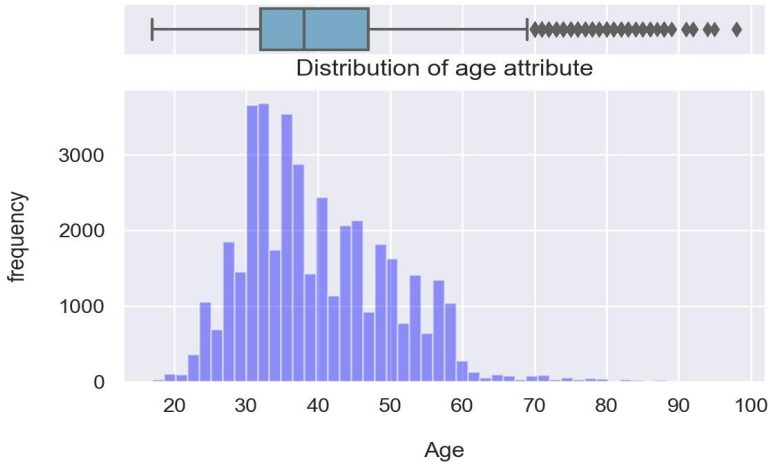
There are 12 rows whose duplicates are present in bank-additional-full data. The .drop_duplicates() method is used to drop the duplicate rows.

In bank-additional, bank-full and bank.csv datasets, there are no duplicate rows.

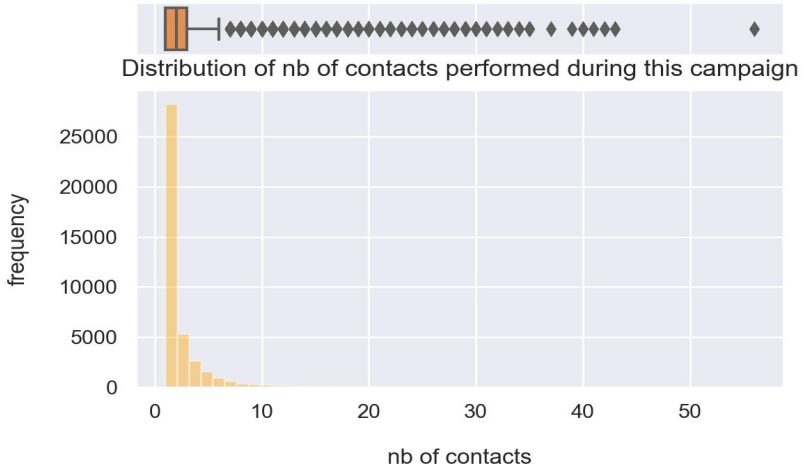
Outliers:

An outlier is a value/observation which lies at an abnormal distance from other values in the normal distribution. It can occur due to an error in measurement or data collection. The following features of bank-additional-full have significant outliers. Outliers can affect the mean of the distribution.

- **‘age’** attribute:



- **‘campaign’** attribute:



In our case, we don’t need to remove outliers from the data since the max(‘age’)=98 and max(‘campaign’)=56 are not unrealistic values. This will help with the generalization of the model later since it should reflect the real world.

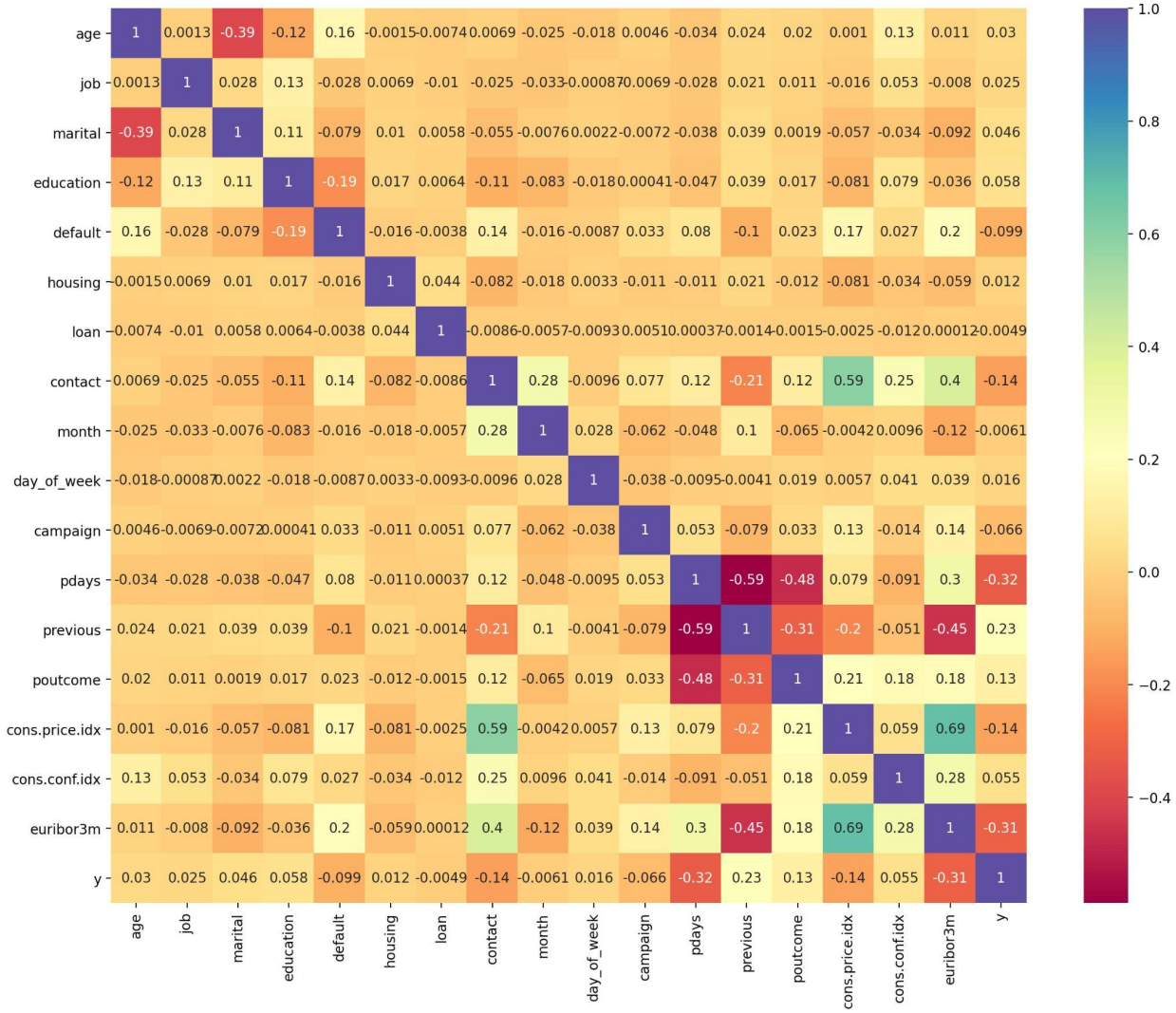
Data Transformation:

- Drop 'Duration' feature: This attribute highly affects the output target (e.g., if duration=0 then y="no"). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
- Remove duplicated rows.
- Handle highly correlated variables: Observations showed positive high correlation between '**emp.var.rate**', '**nr.employed'**, and '**euribor3m'**. So we drop 'emp.var.rate' and 'nr.employed' as 'euribor' and also give us the price of money in the current market.

- Encoding Categorical features using LabelEncoder since Machine learning algorithms can only read numerical values. It is therefore essential to do this step.

Data Dependency:

Now that our dataset contains all numeric variables, we could check correlation between all the features. As shown in the following figure, there are no features that are highly correlated and inversely correlated. If we had, we could have written the condition that if the correlation is higher than 0.8 (or can be any threshold value depending on the domain knowledge) and less than -0.8, we could have dropped those features.



Model Building:

In order to predict the client subscription for a deposit term, we will use a predictive ML model to help us identify potential customers.

As a start we begin by splitting the dataset, into training and testing sets in 80% and 20% respectively.

We choose to test out the following set of models since we don't know yet what algorithms will do well on this dataset.

The following algorithms selected for this classification problem include:

- **Linear Algorithms:**

  Logistic Regression (LR) ***(Base Model)***

  Linear Discriminant Analysis (LDA).

- **Nonlinear Algorithms:**

  Classification and Regression Trees (CART),

  Gaussian Naive Bayes (NB)

  k-Nearest Neighbors (KNN).

- **Ensemble Methods:**

  Boosting Methods: AdaBoost (AB) and Gradient Boosting (GBM).

  Bagging Methods: Random Forests (RF) and Extra Trees (ET).

Results from Modelling:

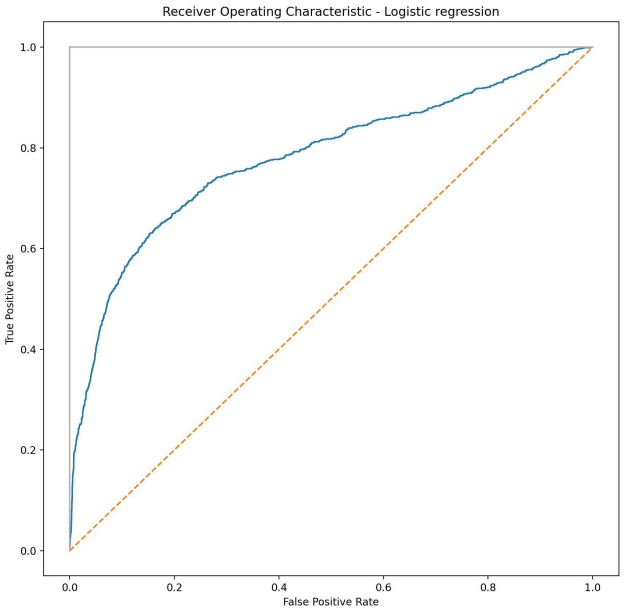
We suspected that the differing distributions of the raw data may be negatively impacting the skill of some of the algorithms. Therefore, we evaluated the same algorithms with a standardized copy of the dataset.

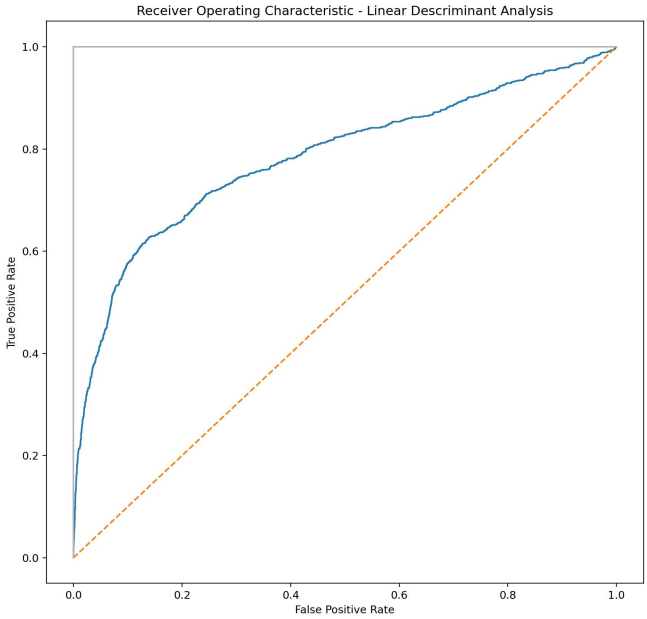
This is where the data is transformed such that each attribute has a mean value of zero and a standard deviation of one. We also need to avoid data leakage when we transform the data. A good way to avoid leakage is to use pipelines that standardize the data and build the model for each fold in the cross validation test harness. That way we can get a fair estimation of how each model with standardized data might perform on unseen data.

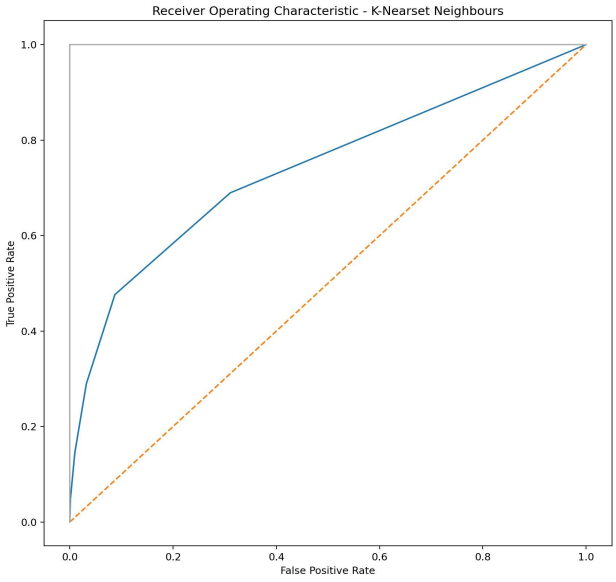
Linear Algorithms & Nonlinear Algorithms results:

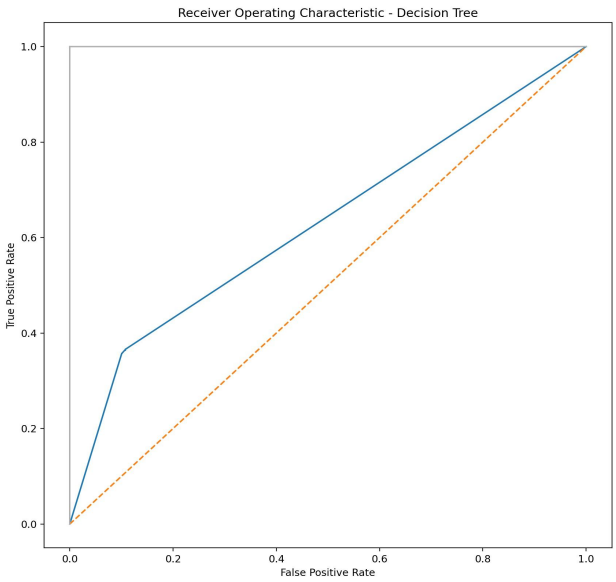
	Model Name	ROC AUC Score
0	LR	0.780634
1	LDA	0.783685
2	KNN	0.739323
3	CART	0.629956
4	NB	0.769487

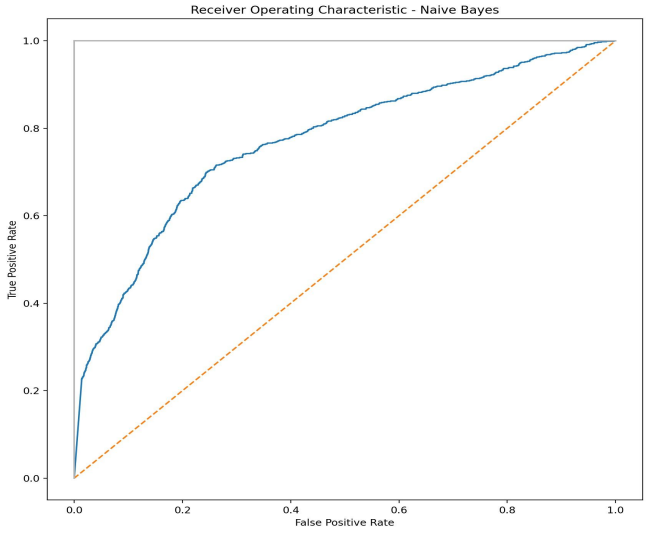
Logistic Regression is used as a base model in our case. As shown in the table, LDA gave better results than LR. Here are the ROC AUC Curves plotted separately for each model.


Receiver Operating Characteristic - Logistic regression
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - Linear Descriminant Analysis
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - K-Nearset Neighbours
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic - Decision Tree
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0



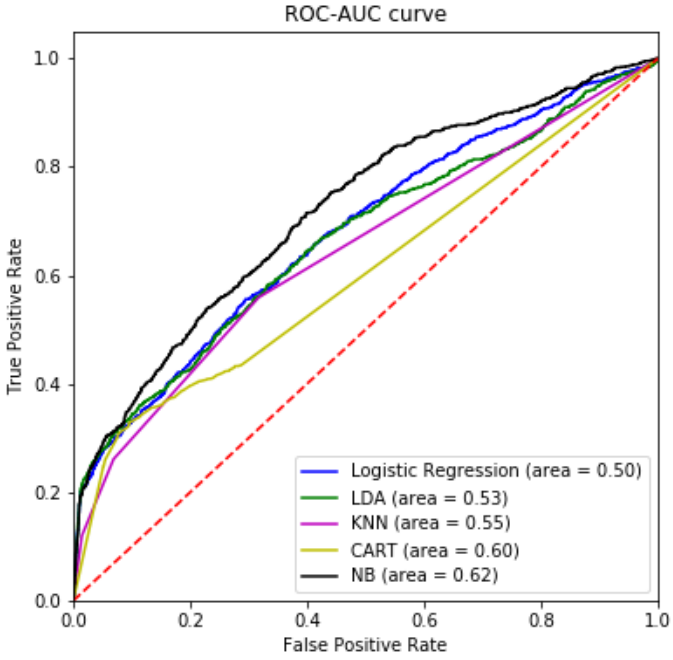
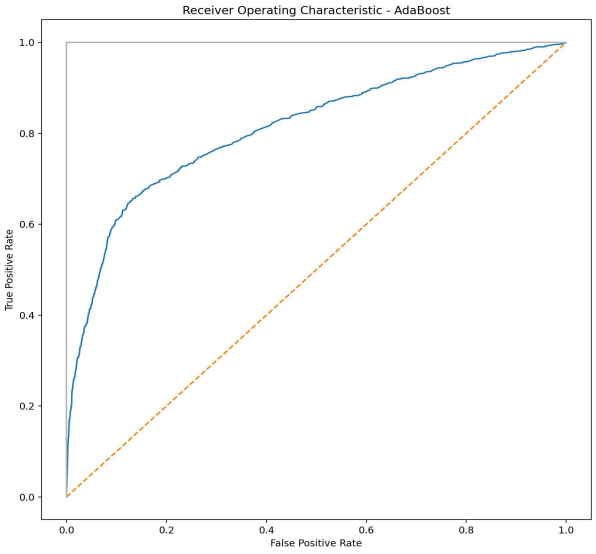
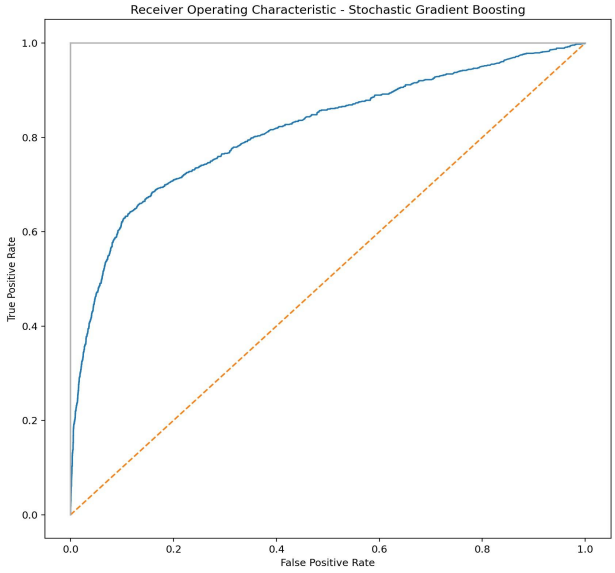


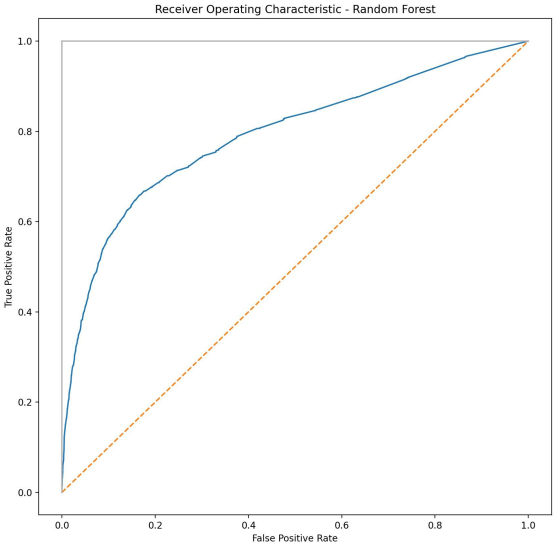
Figure: ROC_AUC curves for the linear and non-linear algorithms. For the plot, Standard Scaler and Cross Validation is not used to get a basic result.

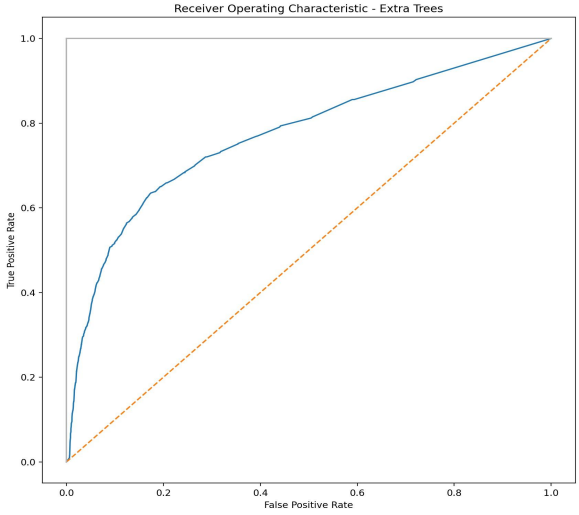
Ensemble Methods results:

	Model Name	ROC AUC Score
0	AB	0.812690
1	GBM	0.814578
2	RF	0.791371
3	ET	0.771145









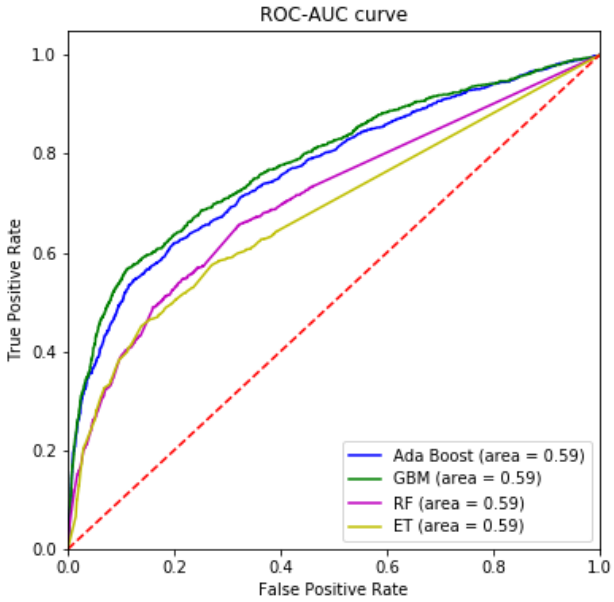


Figure: ROC_AUC curves for Ensemble Methods Algorithms. Curves produced from data without using Standard Scaler and Cross Validation to get a basic result.

Final recommendation:

We can see that both boosting techniques provide strong accuracy scores in the high 70s (%) and even reached 80% with default configurations. The GBM model is the best model compared to the other ones. Therefore we will consider that model for production.

Github Repo link:

https://github.com/AsAmira02/Bank-Marketing-Campaign-DSEnthusiasts2021

This repository includes the four datasets, model code and necessary files used in this project.