Capstone Project - III

ML Supervised Classification

Bank Marketing Effectiveness Prediction

Presented by-

Prayag Raj Dubey

CONTENT

01 STEP Objective

02 STEP Data Summary

03 STEP Exploratory Data Analysis

04 STEP Feature Engineering

05 STEP Data Preparation

06 STEP Handling Imbalance

07 STEP Classification Models

08 STEP Conclusions

Data Description and Objective

The data is related with direct marketing campaigns (phone calls) 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 ('yes') or not ('no') subscribed.

The classification goal is to predict if the client will subscribe a term deposit (variable y).

Dataset Attributes and their Description

Bank Client data:

- **age** (numeric)
- **job** : type of job (categorical)
- **marital** : marital status (categorical)
- **education** (categorical)
- **default**: has credit in default? (categorical)
- **housing**: has housing loan? (categorical)
- **loan**: has personal loan? (categorical)

Related with the last contact of the current campaign:

- **contact**: contact communication type (categorical)
- **month**: last contact month of year (categorical)
- **Day of week**: last contact day of the week (categorical)
- **duration**: last contact duration, in seconds (numeric)

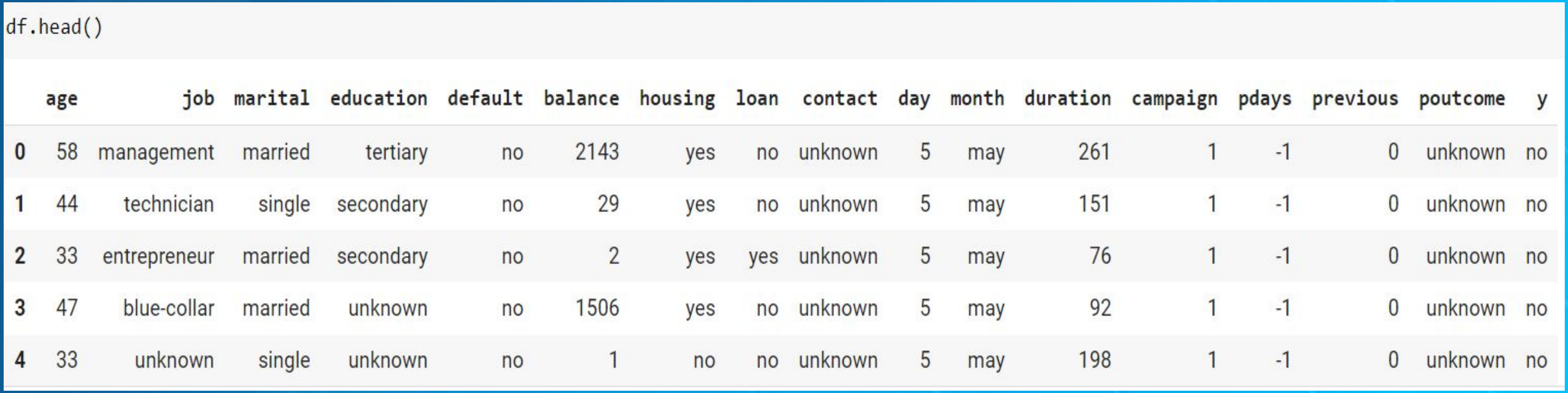
Other attributes:

- **campaign**: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- **pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric)
- **previous**: number of contacts performed before this campaign and for this client (numeric)
- **poutcome**: outcome of the previous marketing campaign (categorical)

Output variable (desired target):

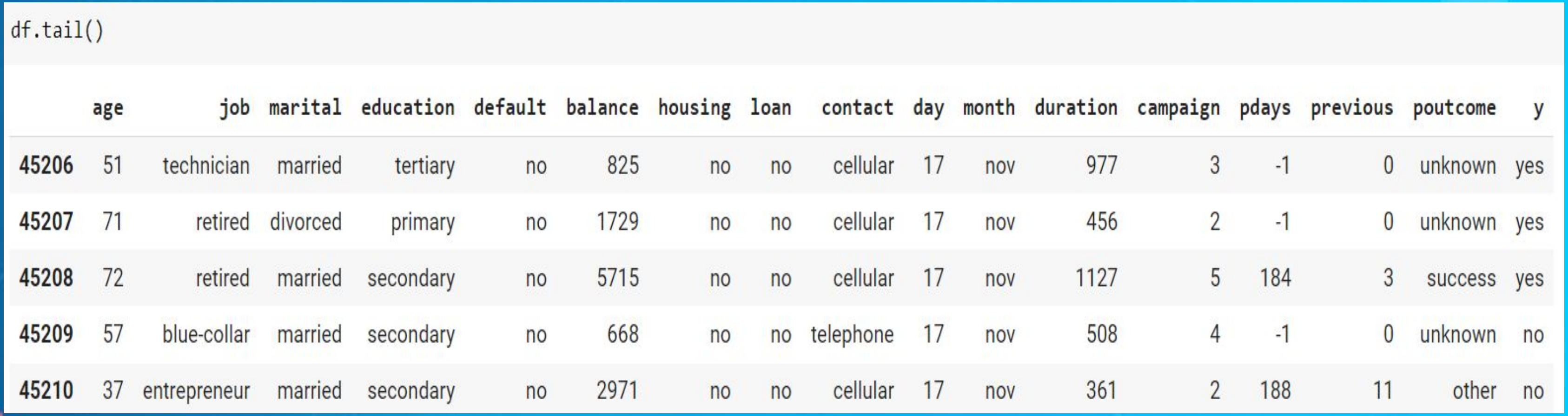
- **y** - has the client subscribed a term deposit? (binary: 'yes',’no)

Dataset Inspection



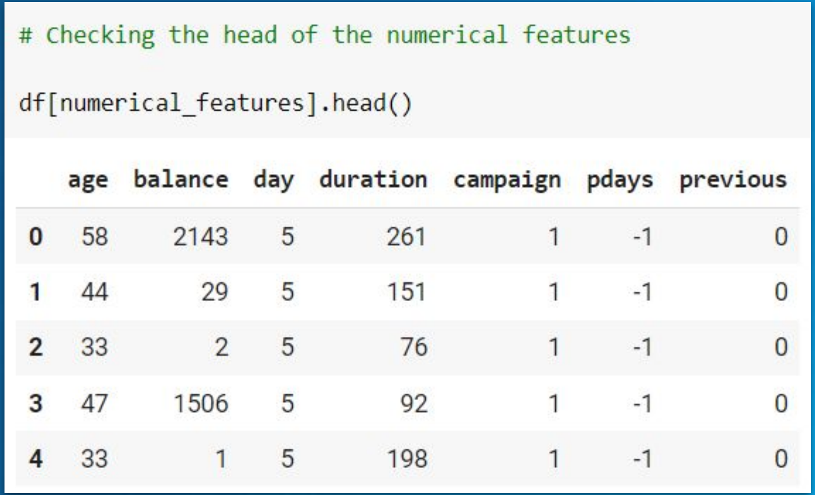
```
df.head()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
3	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no



```
df.tail()
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	unknown	yes
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	unknown	yes
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success	yes
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	unknown	no
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other	no

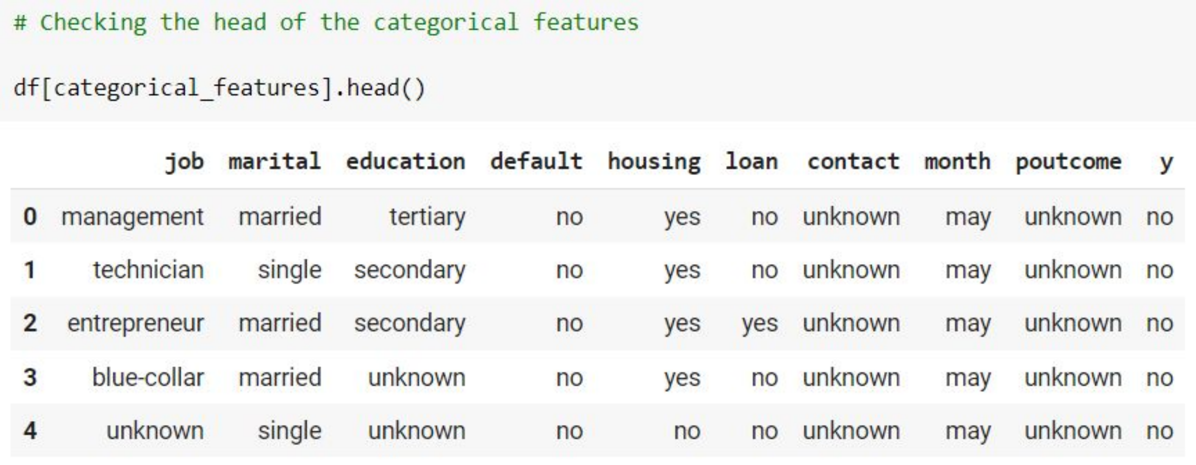


```
# Checking the head of the numerical features

df[numerical_features].head()
```

	age	balance	day	duration	campaign	pdays	previous
0	58	2143	5	261	1	-1	0
1	44	29	5	151	1	-1	0
2	33	2	5	76	1	-1	0
3	47	1506	5	92	1	-1	0
4	33	1	5	198	1	-1	0

After doing the basic dataset inspection we spilled the dataset in categorical and numerical variables separately.



```
# Checking the head of the categorical features

df[categorical_features].head()
```

	job	marital	education	default	housing	loan	contact	month	poutcome	y
0	management	married	tertiary	no	yes	no	unknown	may	unknown	no
1	technician	single	secondary	no	yes	no	unknown	may	unknown	no
2	entrepreneur	married	secondary	no	yes	yes	unknown	may	unknown	no
3	blue-collar	married	unknown	no	yes	no	unknown	may	unknown	no
4	unknown	single	unknown	no	no	no	unknown	may	unknown	no

Data Exploration

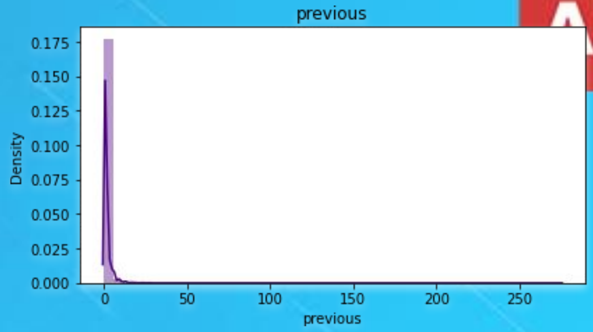
- The dataset has 45211 rows and 17 features (columns).
- 10 categorical features.
- 7 Numerical features.
- No null values.
- No Duplicate values.
- No Missing Values.

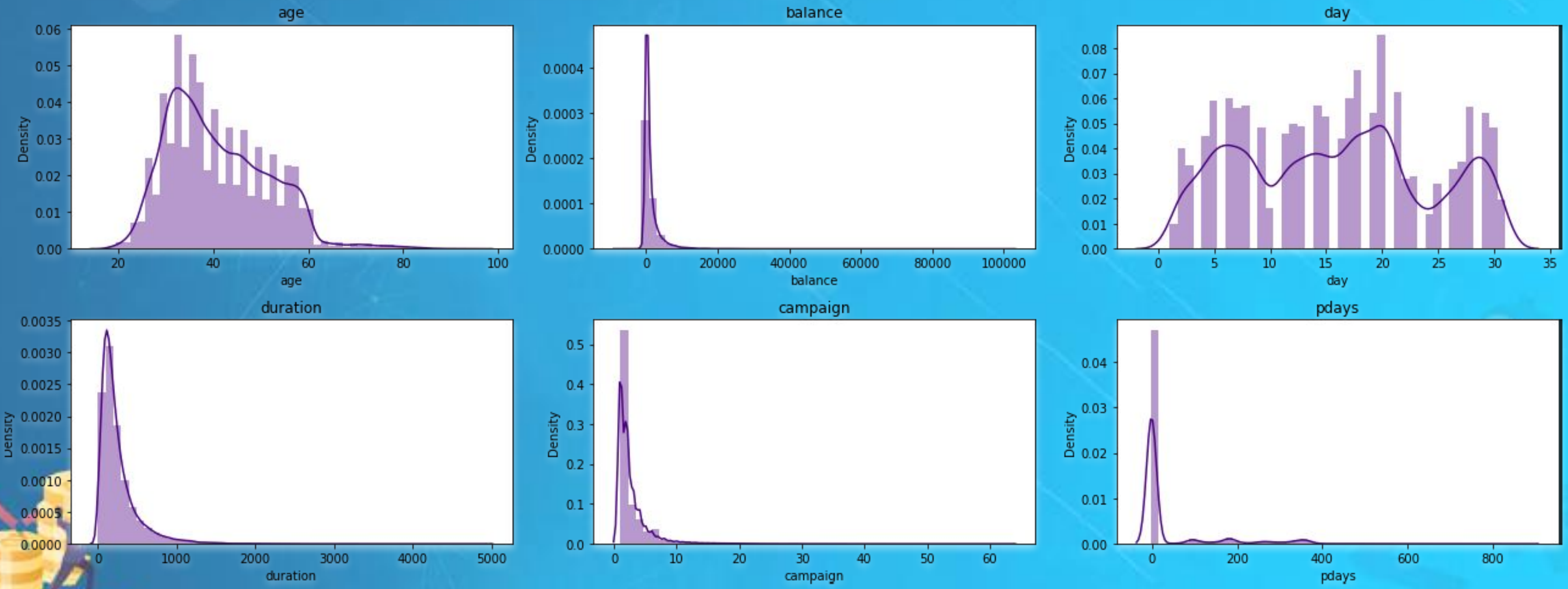
EDA

- Exploratory data analysis or commonly known as EDA helps to explore data, and possibly formulate hypotheses that might cause new data collection and experiments. EDA build a robust understanding of the data, issues associated with either the info or process. it’s a scientific approach to get the story of the data.

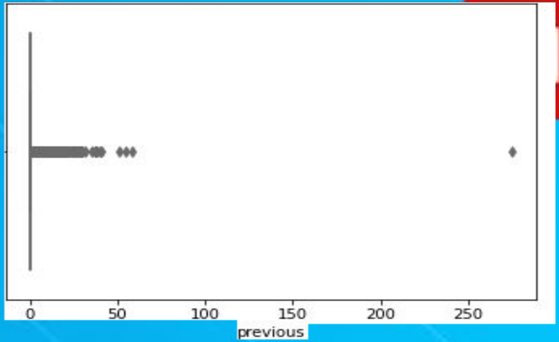
- It focuses more narrowly on checking assumptions required for model fitting and hypothesis testing. It also helps while handling missing values and making transformations of variables as needed.

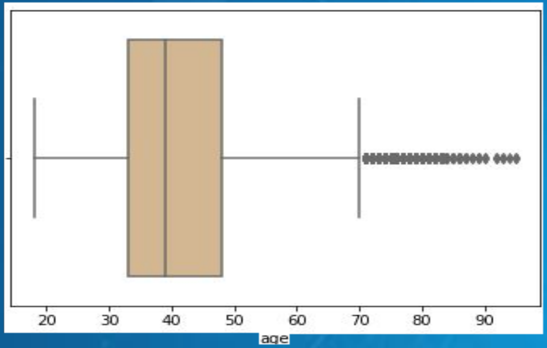
Univariate analysis of numerical features

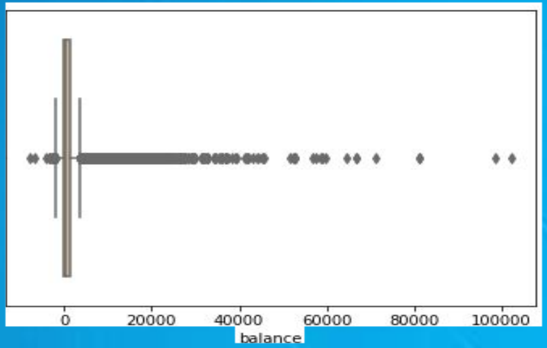


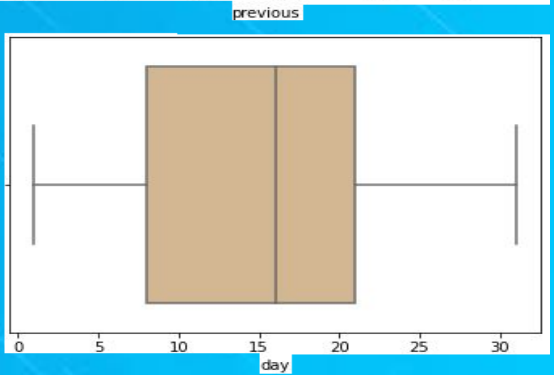


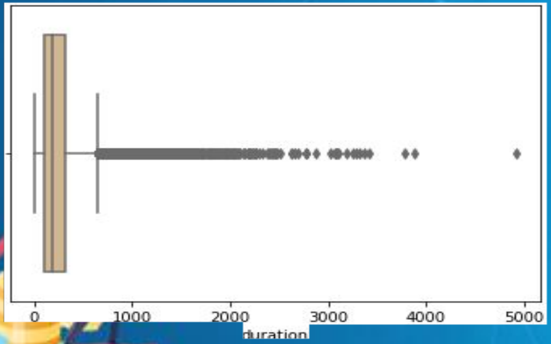
Outlier Detection of Numerical Features

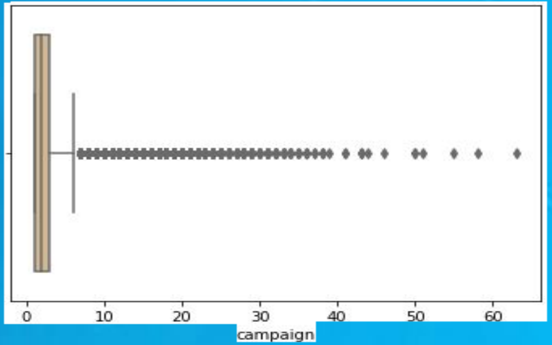


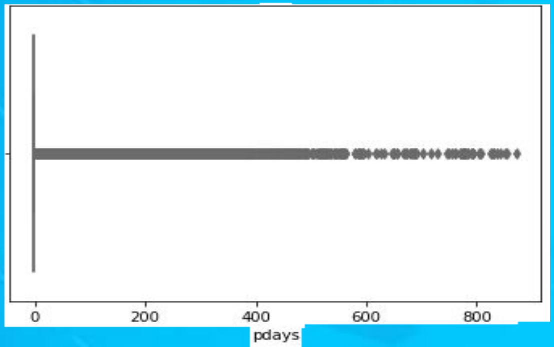




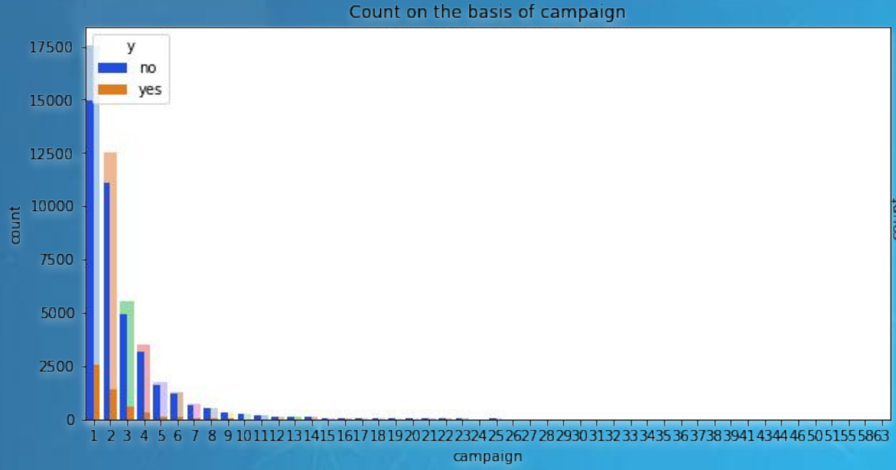


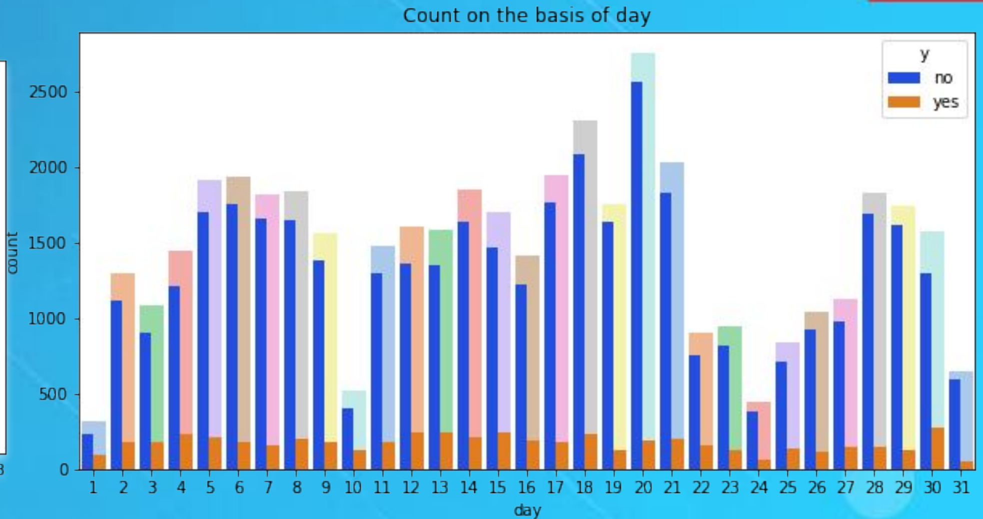


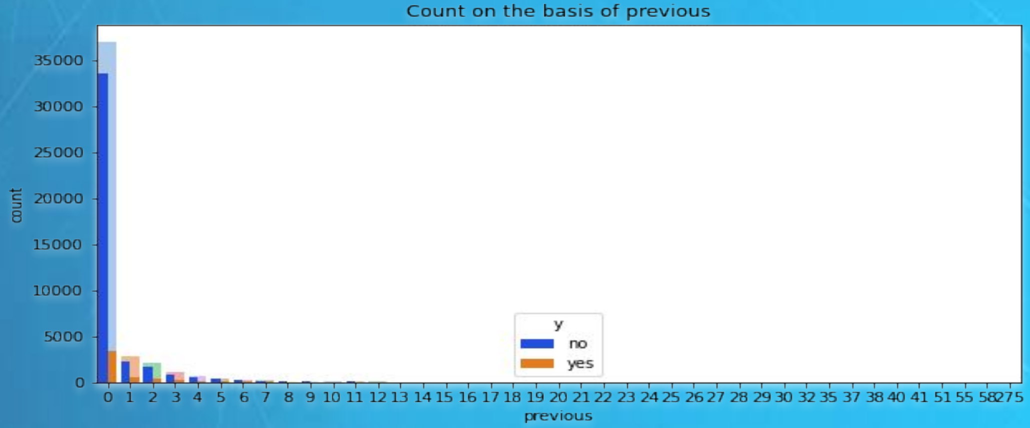




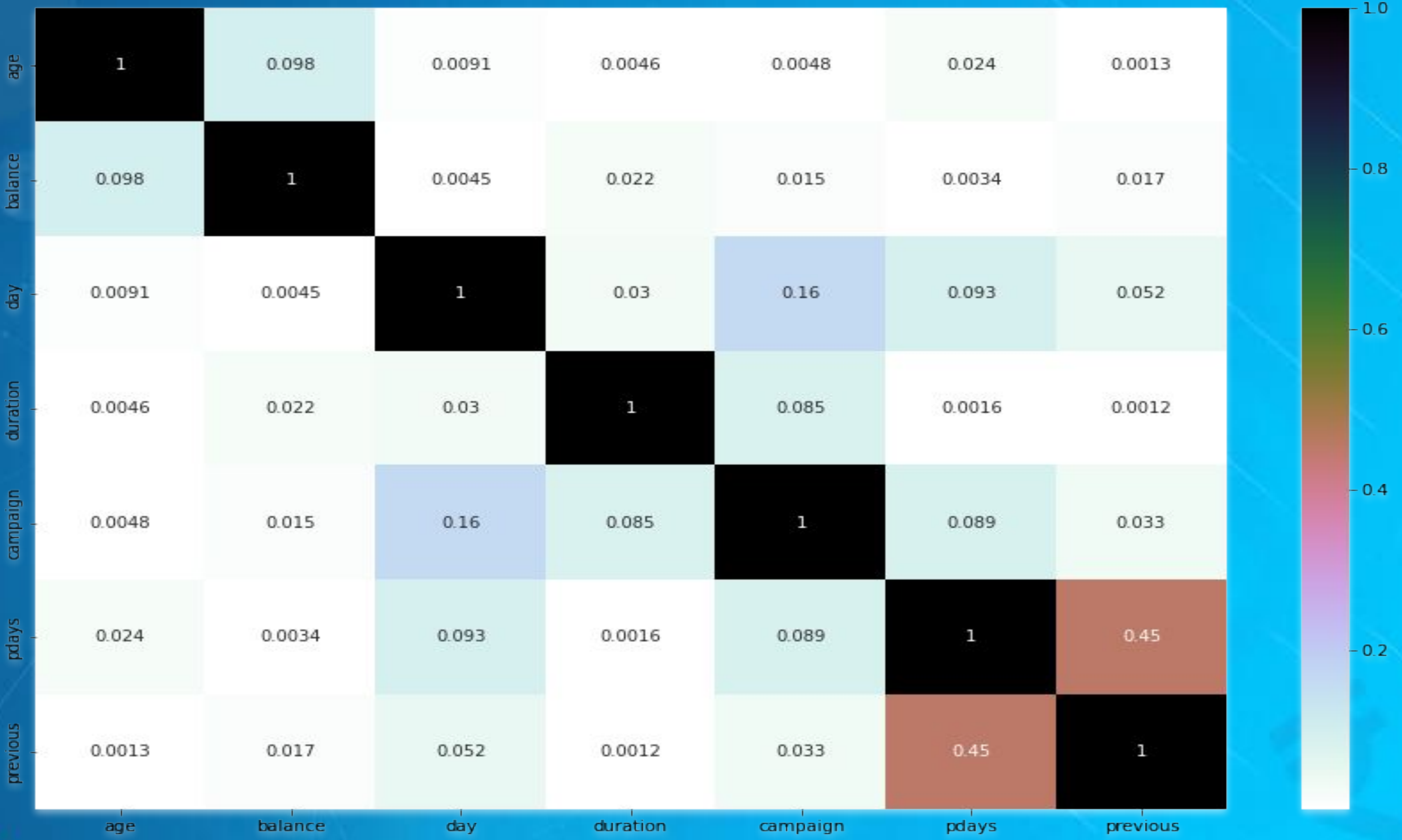
Bivariate Analysis of numerical feature with Target Variable


Count on the basis of campaign


Count on the basis of day

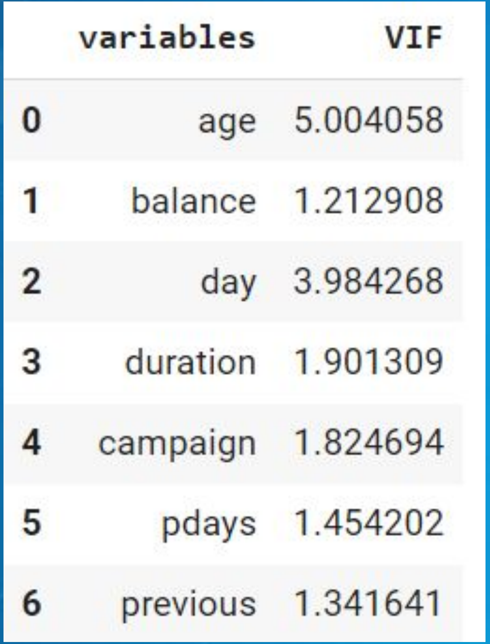

Count on the basis of previous

Checking Multicollinearity



	age	balance	day	duration	campaign	pdays	previous
age	1	0.098	0.0091	0.0046	0.0048	0.024	0.0013
balance	0.098	1	0.0045	0.022	0.015	0.0034	0.017
day	0.0091	0.0045	1	0.03	0.16	0.093	0.052
duration	0.0046	0.022	0.03	1	0.085	0.0016	0.0012
campaign	0.0048	0.015	0.16	0.085	1	0.089	0.033
pdays	0.024	0.0034	0.093	0.0016	0.089	1	0.45
previous	0.0013	0.017	0.052	0.0012	0.033	0.45	1

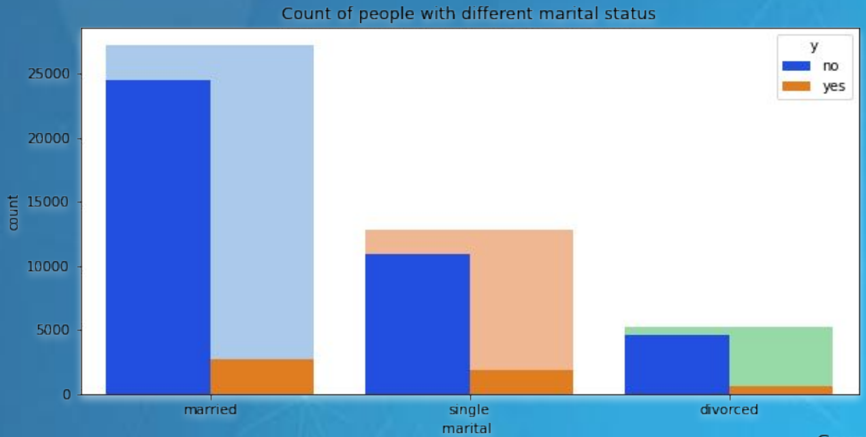
VIF

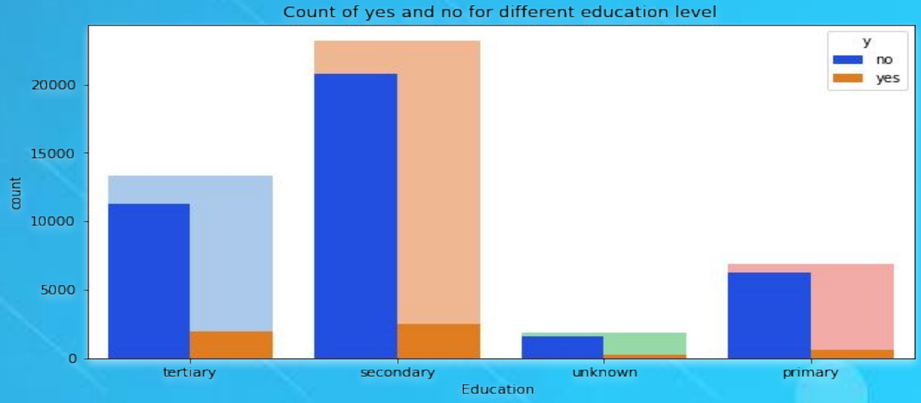


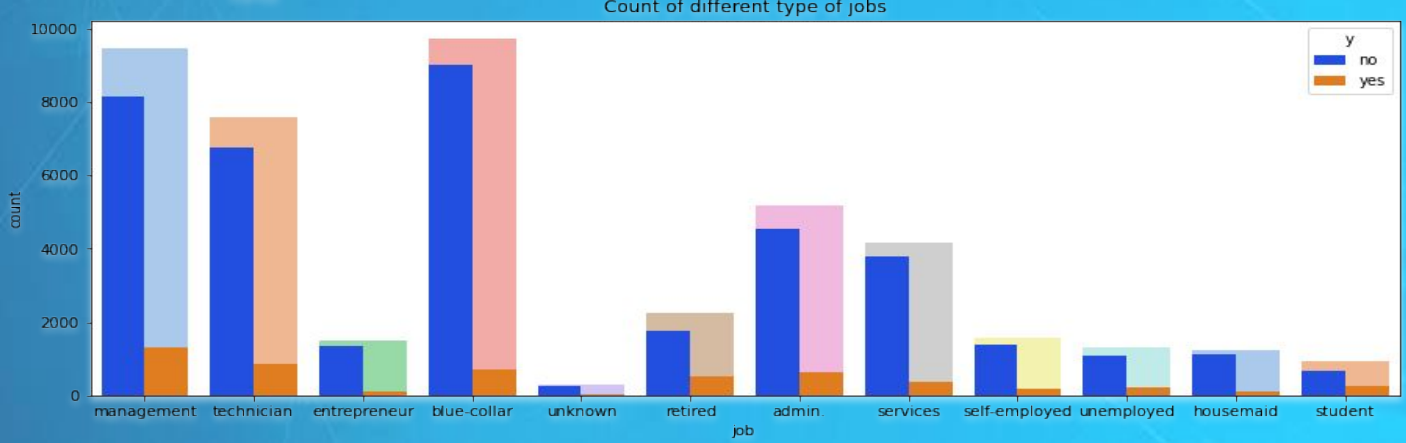
	variables	VIF
0	age	5.004058
1	balance	1.212908
2	day	3.984268
3	duration	1.901309
4	campaign	1.824694
5	pdays	1.454202
6	previous	1.341641

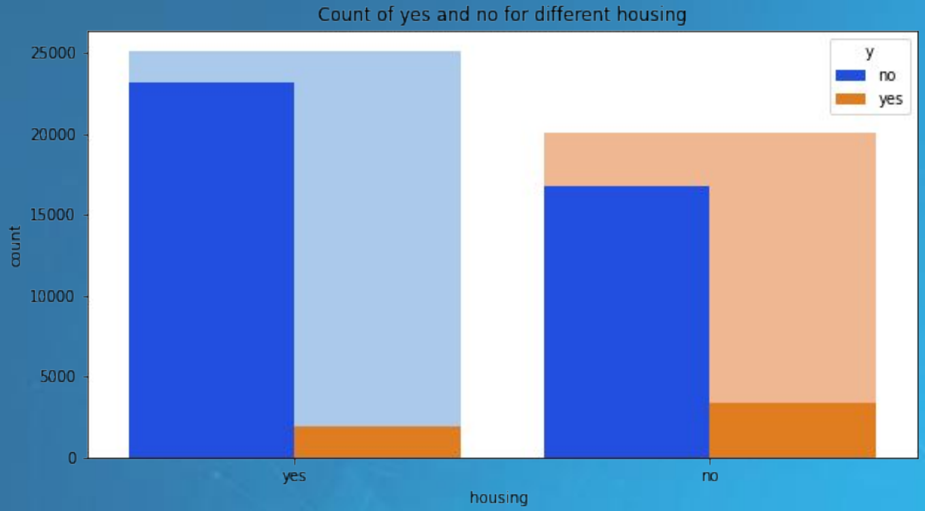
- VIF determines the strength of the correlation between the independent variables.
- VIF less than 5 will be included in the model. In some cases VIF of less than 10 is also acceptable.

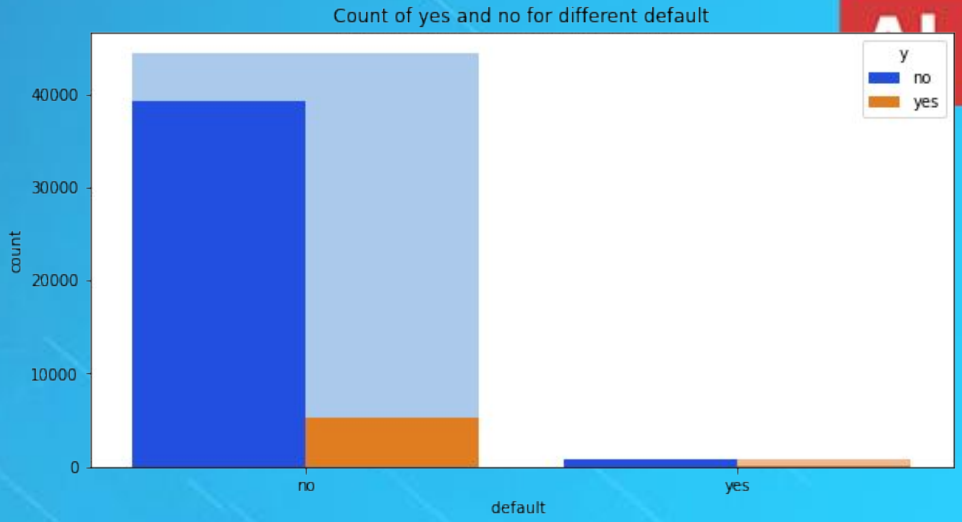
Univariate Analysis of Categorical Features & relation with Target Variable

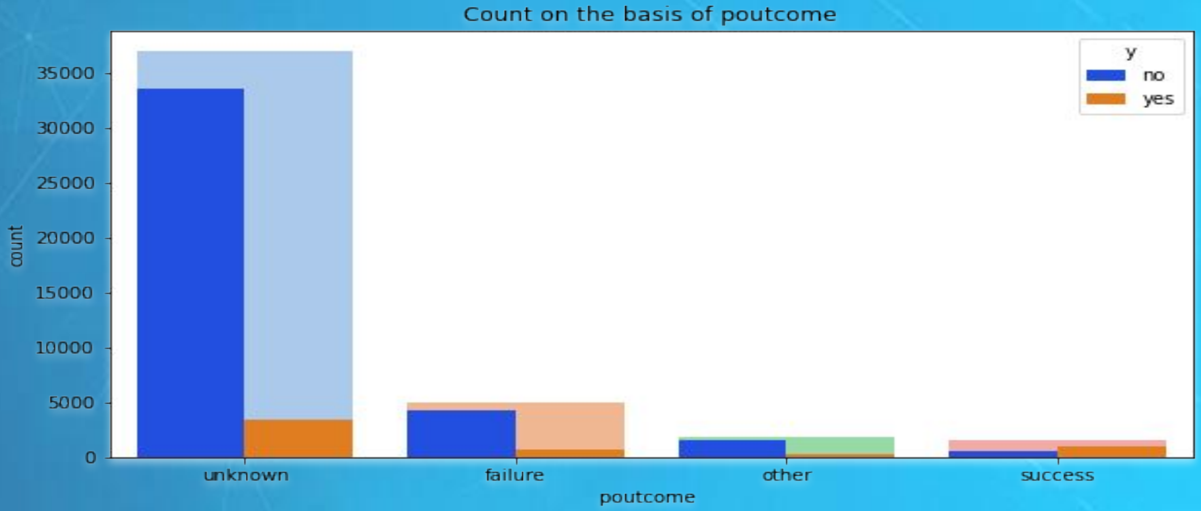

Count of people with different marital status


Count of yes and no for different education level

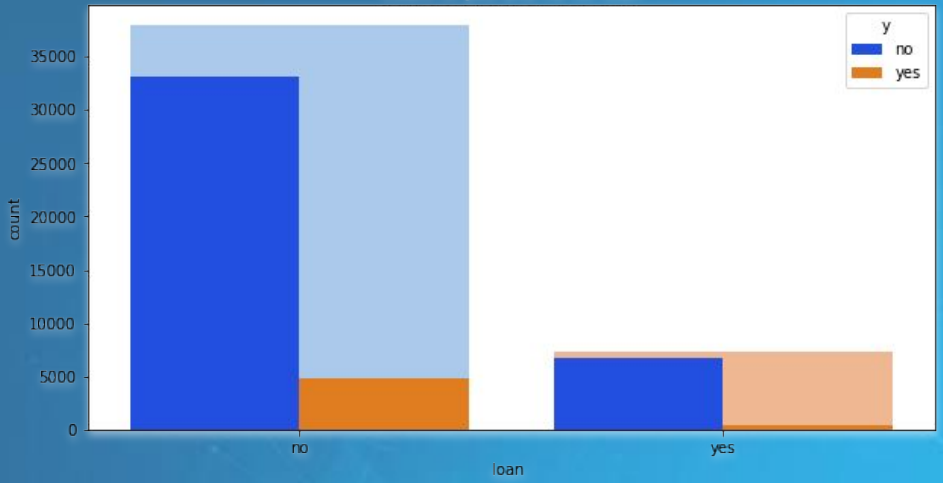

Count of different type of jobs


Count of yes and no for different housing

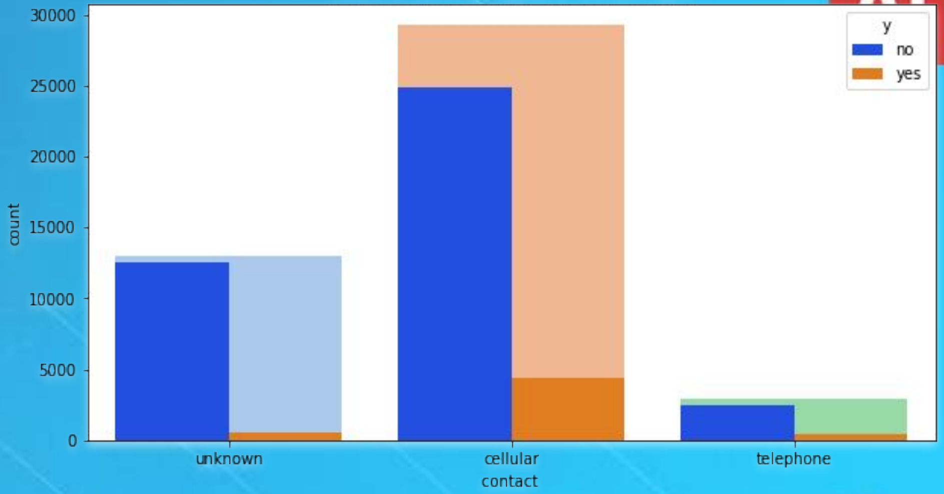

Count of yes and no for different default


Count on the basis of poutcome

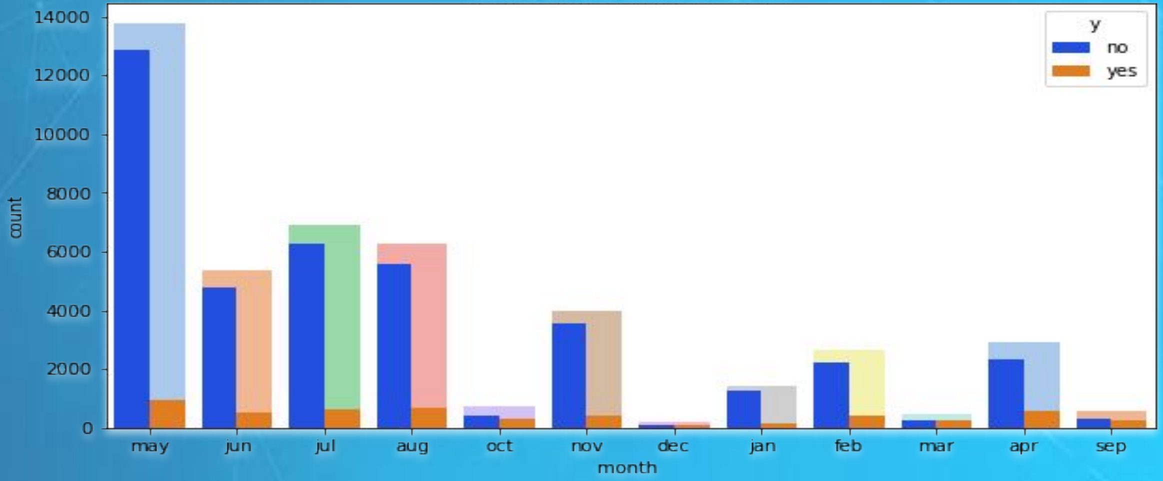
Count of yes and no for loan



Count of different types of contact type



Count on the basis of month



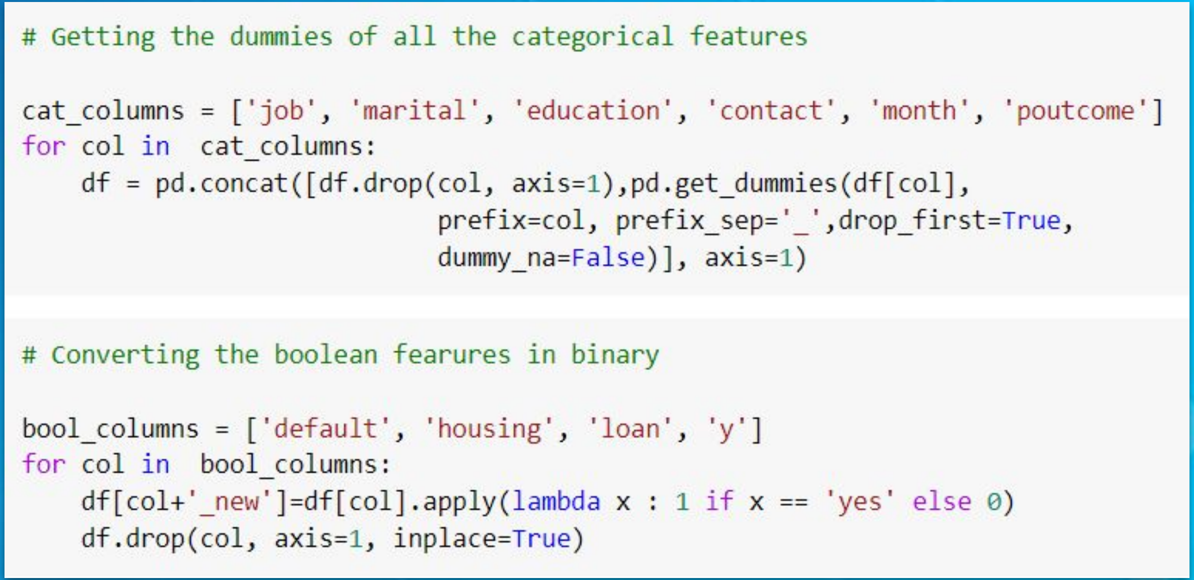
Feature Engineering

All machine learning algorithms use some input data to create outputs. Algorithms require features with some specific characteristics to work properly. Here, the need for feature engineering arises. Feature engineering mainly have two goals:

- Preparing the proper input dataset, compatible with the machine learning algorithm requirements.
- Improving the performance of machine learning models.

We'll try adding and removing some features in this section in order to make a perfect data matrix we can pass to a machine learning model. We will try to interpret categorical features as numeric to be passed to the ML models.

So, This is the code which we have applied for feature engineering

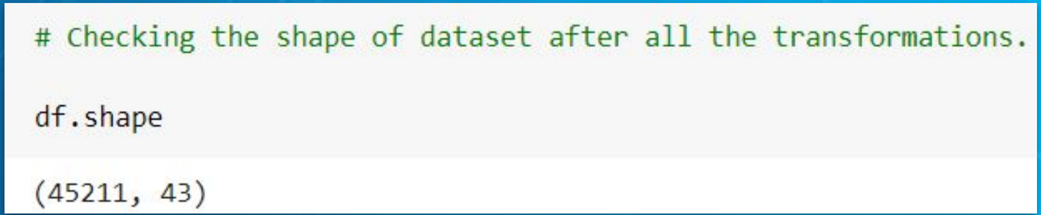


```
# Getting the dummies of all the categorical features

cat_columns = ['job', 'marital', 'education', 'contact', 'month', 'poutcome']
for col in  cat_columns:
    df = pd.concat([df.drop(col, axis=1),pd.get_dummies(df[col],
                            prefix=col, prefix_sep='_',drop_first=True,
                            dummy_na=False)], axis=1)
```

```
# Converting the boolean fearures in binary

bool_columns = ['default', 'housing', 'loan', 'y']
for col in  bool_columns:
    df[col+'_new']=df[col].apply(lambda x : 1 if x == 'yes' else 0)
    df.drop(col, axis=1, inplace=True)
```

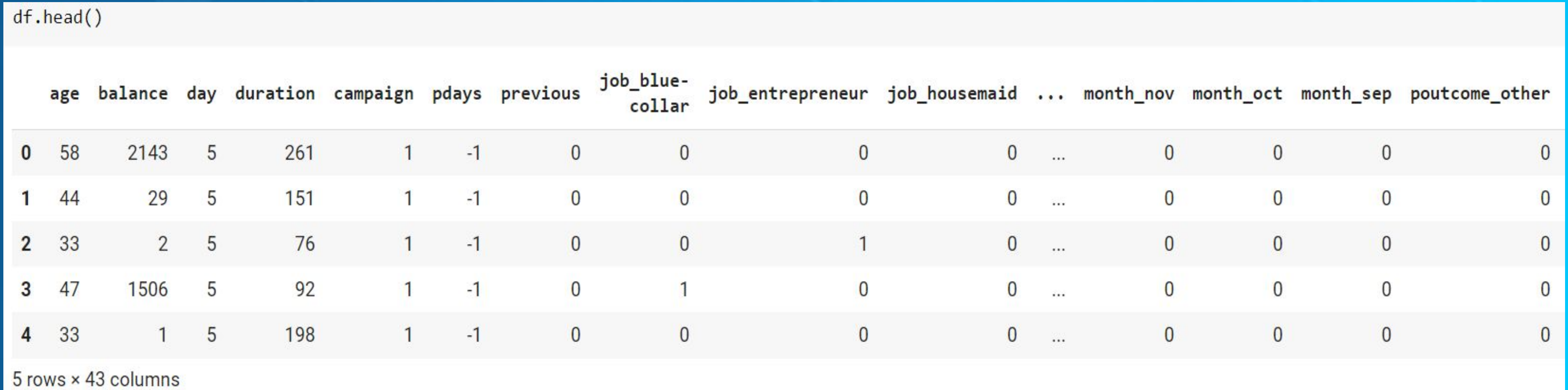


```
# Checking the shape of dataset after all the transformations.

df.shape
```

```
(45211, 43)
```

Now, Here is what our dataset looks like after all the transformations.

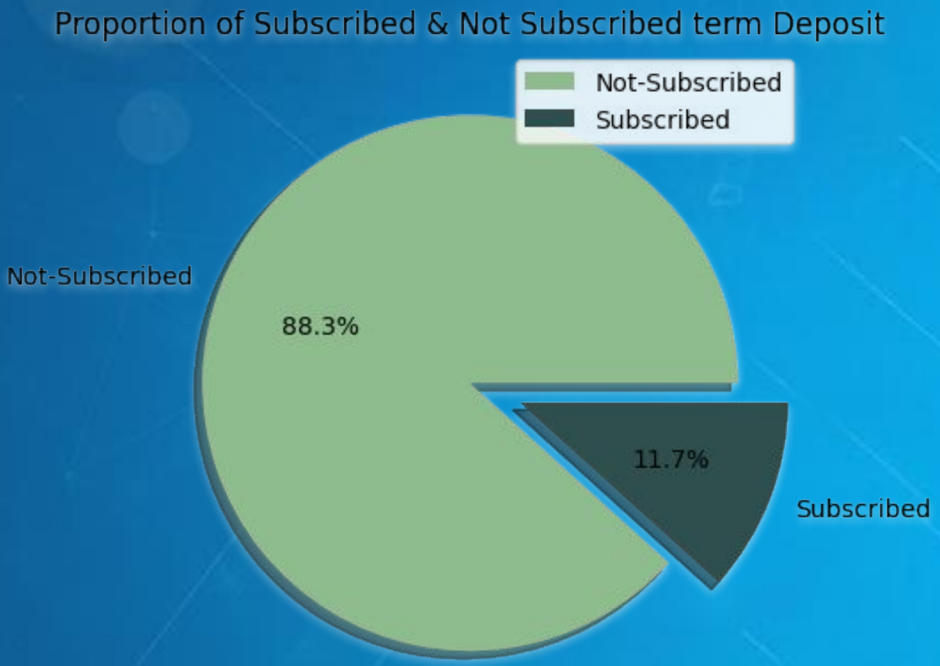


```
df.head()
```

	age	balance	day	duration	campaign	pdays	previous	job_blue-collar	job_entrepreneur	job_housemaid	...	month_nov	month_oct	month_sep	poutcome_other
0	58	2143	5	261	1	-1	0	0	0	0	...	0	0	0	0
1	44	29	5	151	1	-1	0	0	0	0	...	0	0	0	0
2	33	2	5	76	1	-1	0	0	1	0	...	0	0	0	0
3	47	1506	5	92	1	-1	0	1	0	0	...	0	0	0	0
4	33	1	5	198	1	-1	0	0	0	0	...	0	0	0	0

5 rows × 43 columns

Balance of our Target Variable

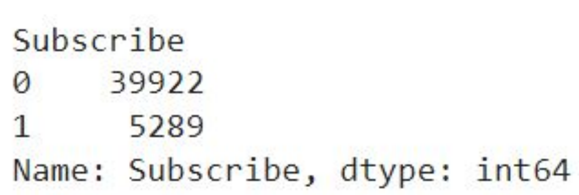


Proportion of Subscribed & Not Subscribed term Deposit

- As we can see that data is highly imbalanced.

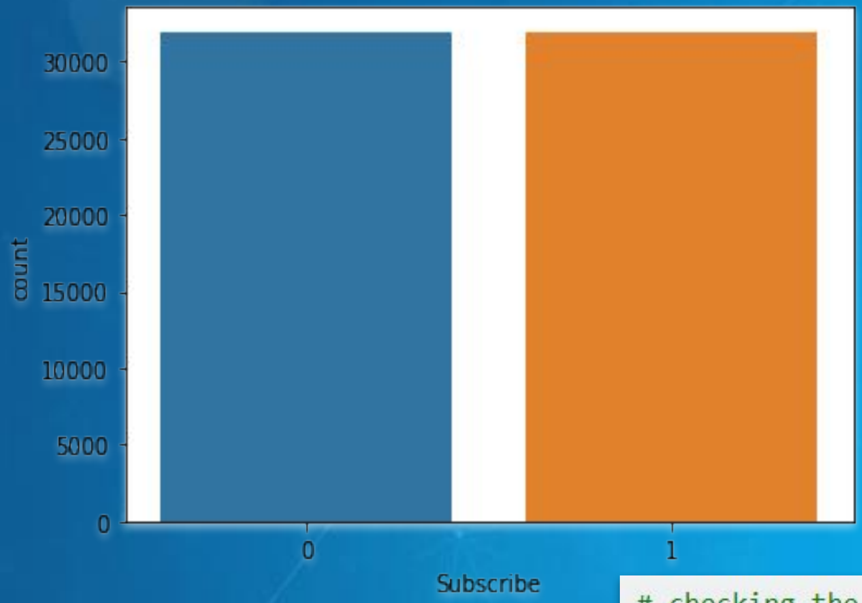
- Majority of the data points belong to “Not - Subscribed” class.

  Ratio of “Not - Subscribed” class to “Subscribed” class is 8:1

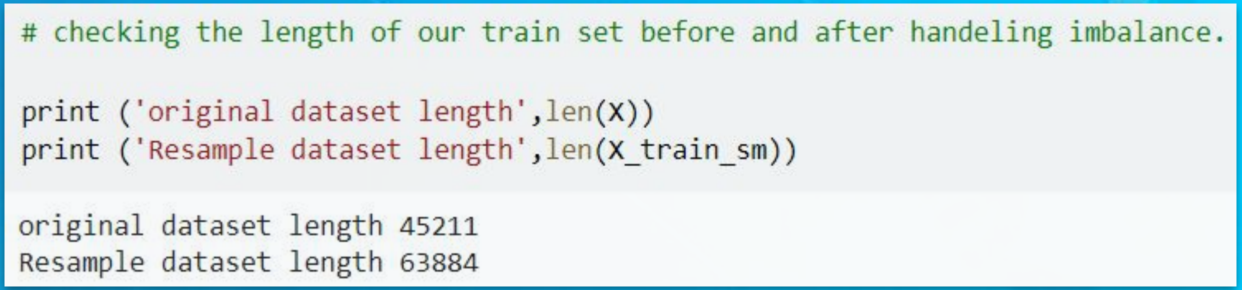


```
Subscribe
0    39922
1     5289
Name: Subscribe, dtype: int64
```

Handling the imbalance in the dataset using SMOTE



SMOTE (Synthetic Minority Oversampling Technique) works by randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.



```
# checking the length of our train set before and after handeling imbalance.

print ('original dataset length',len(X))
print ('Resample dataset length',len(X_train_sm))
```

```
original dataset length 45211
Resample dataset length 63884
```

Data Preparation

❖ Now that the Dataset is cleaned and we have added all the necessary features along with some conversions of categorical features via.,

– Label Encoding

– One Hot Encoding (Dummy Encoding)

❖ Then, We used MinMaxscaler for transforming data

❖ So, now we have split the data into training and testing sets.

– Train Test Split (Test size = “0.2”   Random state = “0”)

Performance Metrics

- **ROC** also known as Receiver Operating Characteristics, shows the performance of binary class classifiers across the range of all possible thresholds plotting between true positive rate and 1-false positive rate.

- **AUC** measures the likelihood of two given random points, one from positive and one from negative, the classifier will rank the positive points above negative points. AUC-ROC is popular classification metric that presents the advantage of being independent of false positive or negative points.

- **F1 SCORE** is the harmonic mean between Precision and Recall. Macro F1 score is used to know how our model works in overall dataset.

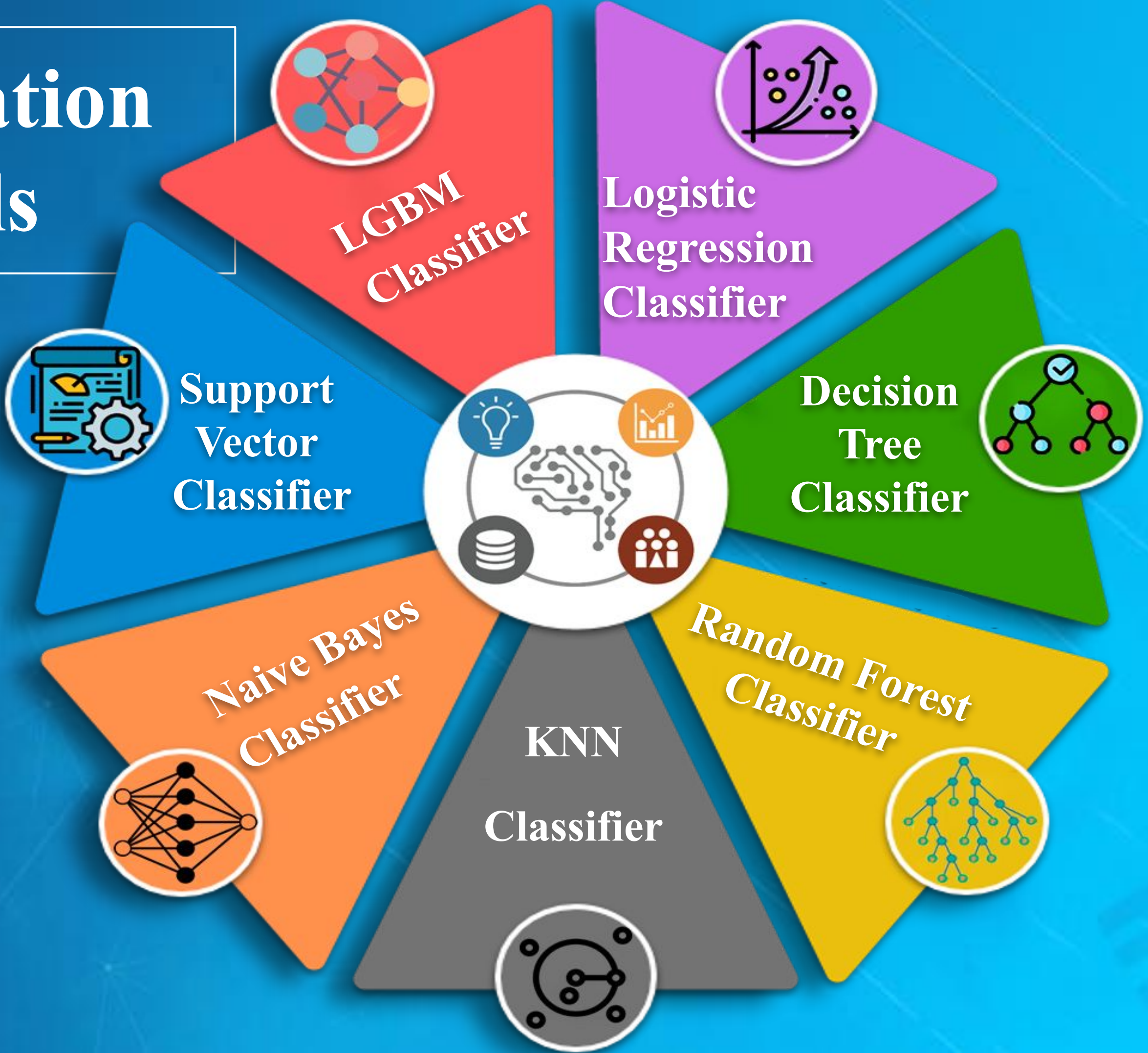
- **Confusion Matrix** gives the count of true negative, true positive, false positive and false negative data points.

Optimization

- Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm while applying this optimized algorithm to any data set.

- GridSearchCV is a technique to search through the best parameter values from the given set of the grid of parameters. It is basically a cross-validation method.

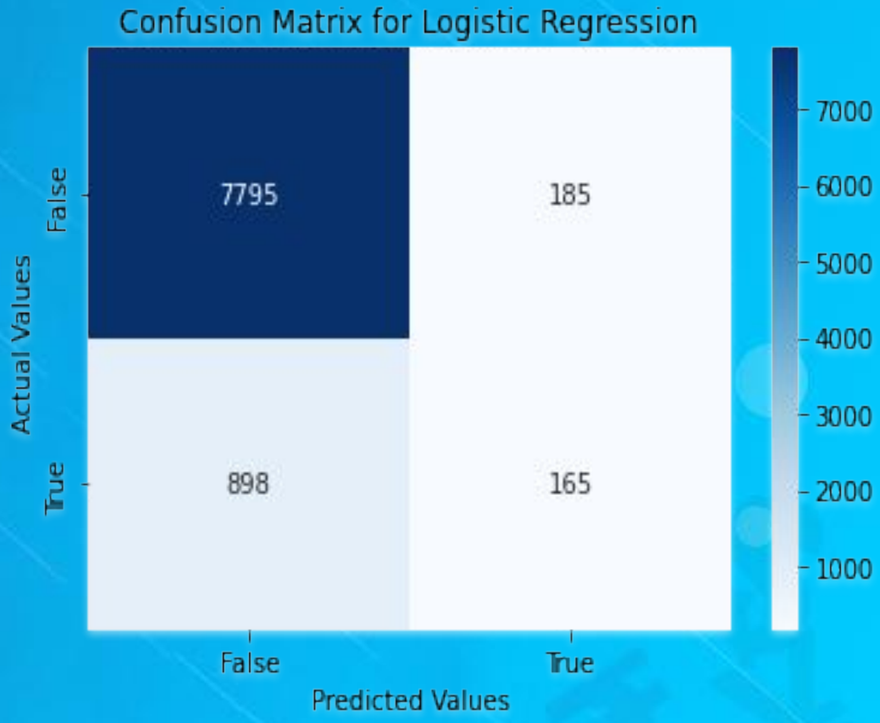
Classification Models



Logistic Regression Classifier

Precision	Recall	F1 Score	Support
0.8802	0.8802	0.8802	0.8802

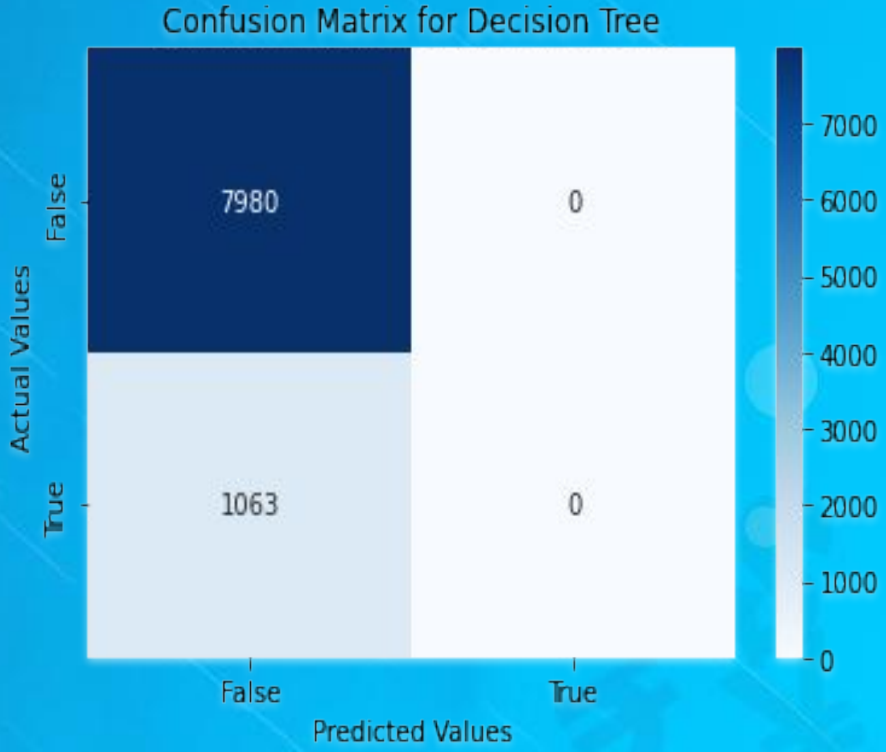
Accuracy on Train data	Accuracy on Test data
0.9299	0.8802



Decision Tree Classifier

Precision	Recall	F1 Score	Support
0.8824	0.8824	0.8824	0.8824

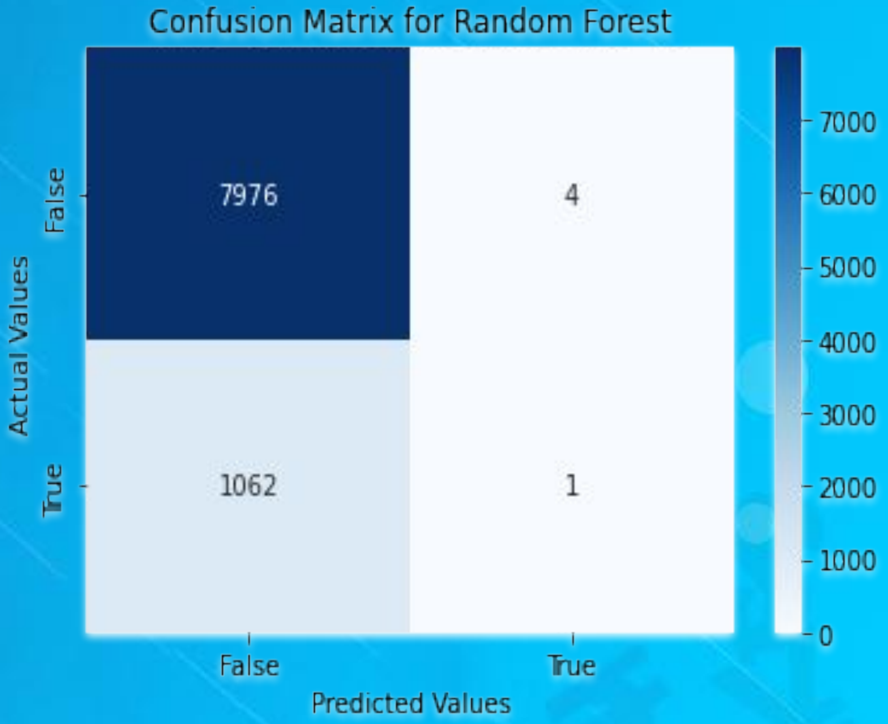
Accuracy on Train data	Accuracy on Test data
0.9214	0.8824



Random Forest Classifier

Precision	Recall	F1 Score	Support
0.8821	0.8821	0.8821	0.8821

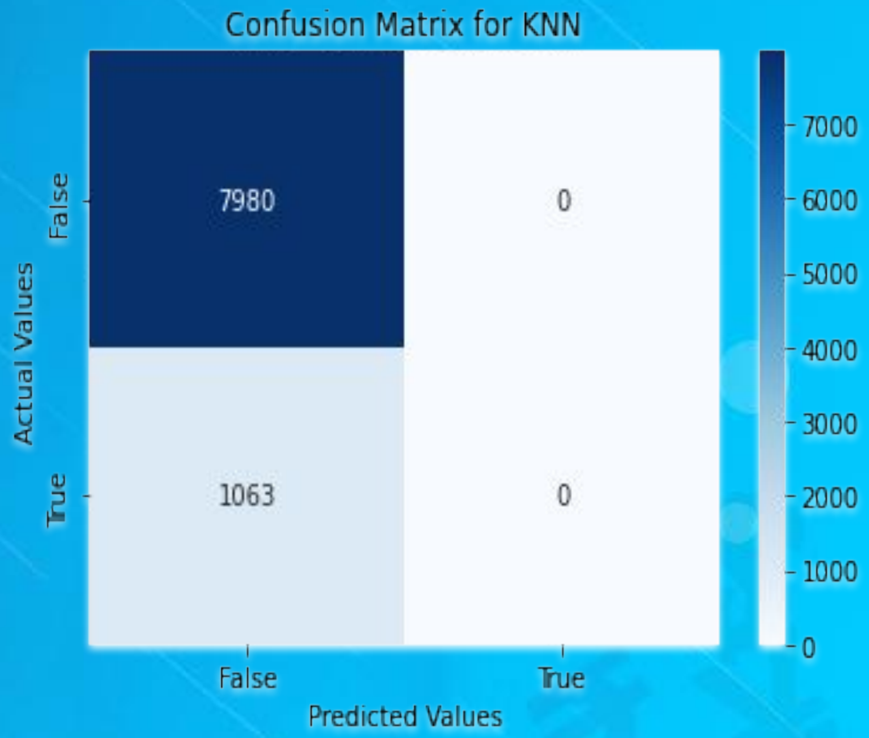
Accuracy on Train data	Accuracy on Test data
0.9290	0.8821



K- Nearest Neighbors Classifier

Precision	Recall	F1 Score	Support
0.8824	0.8824	0.8824	0.8824

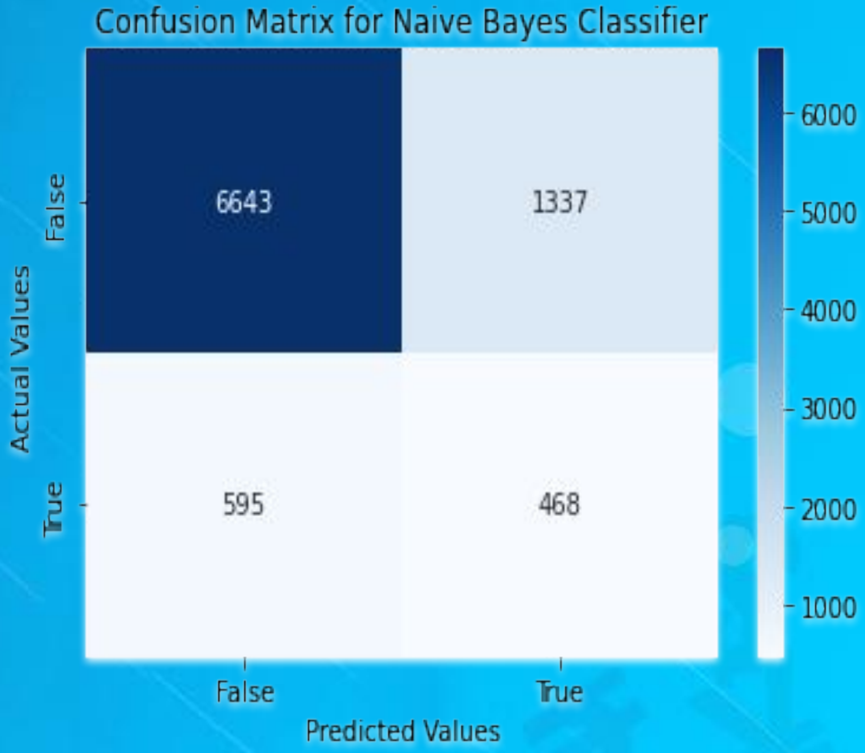
Accuracy on Train data	Accuracy on Test data
1.0	0.8824



Naive Bayes Classifier

Precision	Recall	F1 Score	Support
0.7863	0.7863	0.7863	0.7863

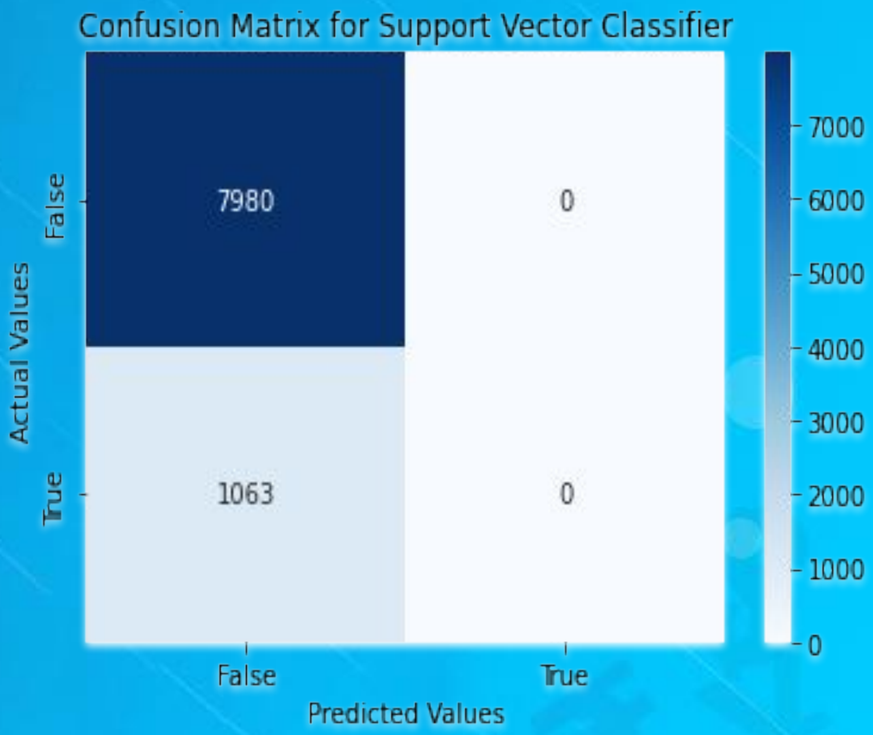
Accuracy on Train data	Accuracy on Test data
0.8862	0.7863



Support Vector Machine Classifier

Precision	Recall	F1 Score	Support
0.8824	0.8824	0.8824	0.8824

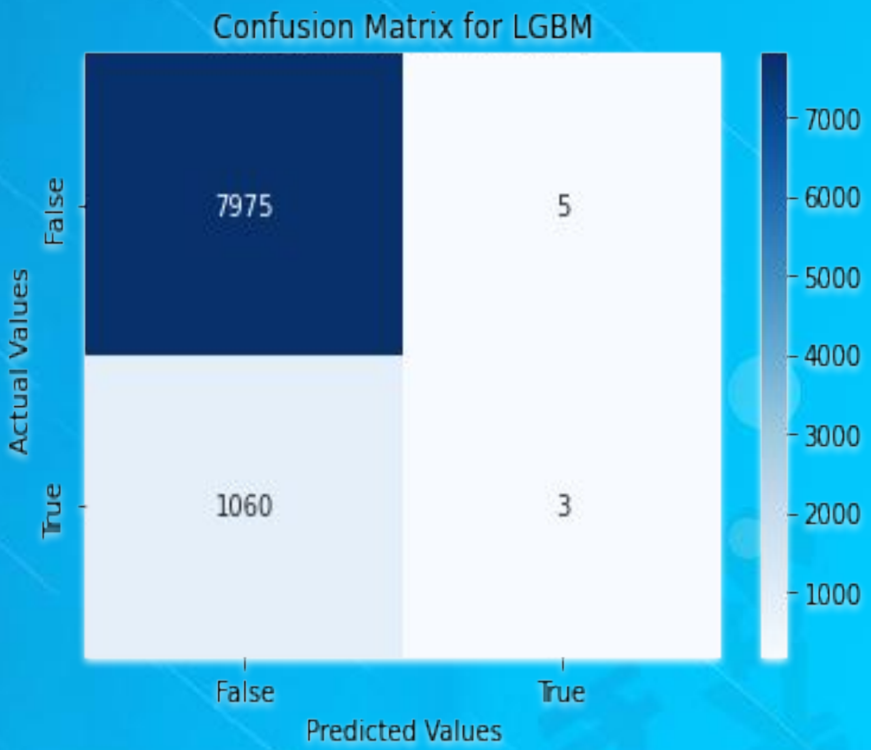
Accuracy on Train data	Accuracy on Test data
0.6200	0.8824



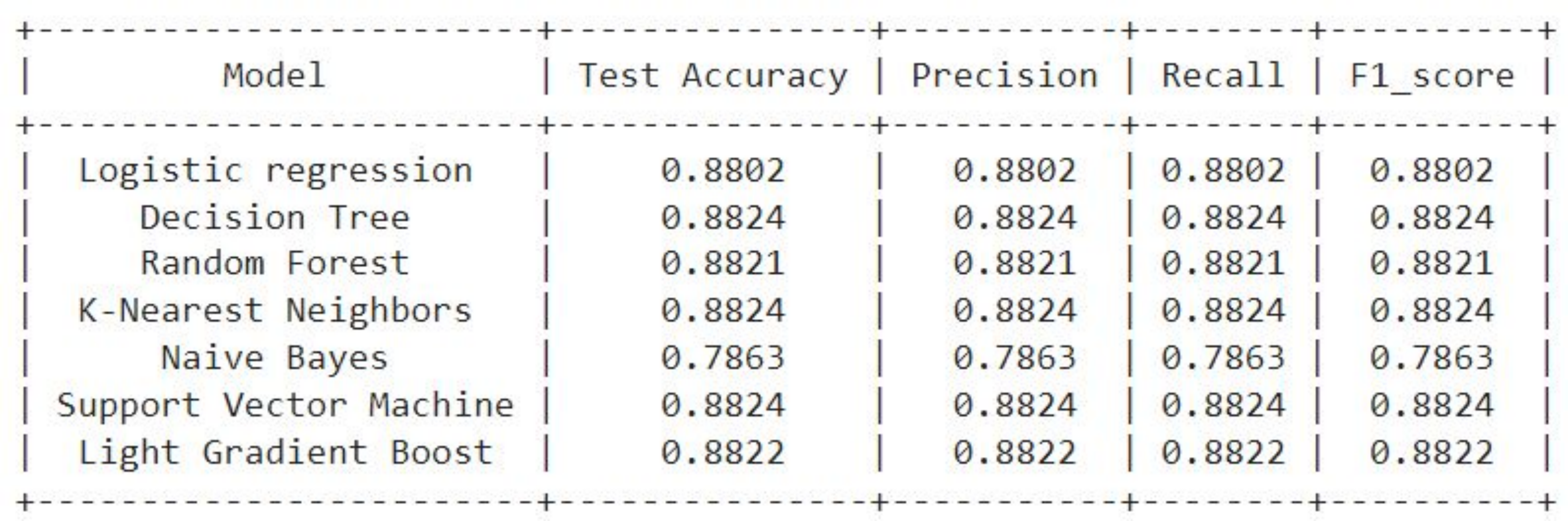
Light Gradient Boost Machine

Precision	Recall	F1 Score	Support
0.8822	0.8822	0.8822	0.8822

Accuracy on Train data	Accuracy on Test data
0.9621	0.8822



Evaluation Metrics For All Models



```
+------------------------+---------------+-----------+--------+----------+
|         Model          | Test Accuracy | Precision | Recall | F1_score |
+------------------------+---------------+-----------+--------+----------+
|  Logistic regression   |     0.8802    |   0.8802  | 0.8802 |  0.8802  |
|     Decision Tree      |     0.8824    |   0.8824  | 0.8824 |  0.8824  |
|     Random Forest      |     0.8821    |   0.8821  | 0.8821 |  0.8821  |
|  K-Nearest Neighbors   |     0.8824    |   0.8824  | 0.8824 |  0.8824  |
|      Naive Bayes       |     0.7863    |   0.7863  | 0.7863 |  0.7863  |
| Support Vector Machine |     0.8824    |   0.8824  | 0.8824 |  0.8824  |
|  Light Gradient Boost  |     0.8822    |   0.8822  | 0.8822 |  0.8822  |
+------------------------+---------------+-----------+--------+----------+
```

Conclusions

- 2nd quarter of the year has the highest number of subscription & Month of May is having the maximum subscriptions.
- Blue-collar, management and technician showed maximum interest in subscription.
- Compared to married and single, Divorced people have less interest in term deposit
- People with secondary education followed by tertiary education were subscribed to term deposit.
- Generally people who don't have credit in default are interested in deposit. Majority of the people have home loan but only few of them opted for term deposit.
- Cellular communication is seen more effective in comparison to other communication types
- The calls with large duration has more tendency for conversion.
- Majority of people were not contacted previously before this campaign.
- We can choose **KNN** or **Decision Tree** to predict Effectiveness as both of them are showing same accuracy of 88% & F1- Score of (**0.8824**).

Thank you