IE7300 Statistical Learning for Engineering

Client Subscription Classification

Group 5

Chaithanya Gudipati

Pooja Ramesh

Stuti Dhebar

Ramesh.po@northeastern.edu

Gudipati.c@northeastern.edu

Dhebar.s@northeastern.edu

Submission Date: Dec 8st, 2023

Abstract:

In the context of a Portuguese Banking institution's direct marketing campaign for term deposits, we address the challenges of predicting the client subscriptions. We navigate through imbalances, feature complexities and outliers to optimize predictive accuracy, providing insights into the effectiveness of marketing strategies. Our robust workflow aims to enhance subscription rates ad empower data driven decision making for the future campaigns.

Problem Definition:

A series of direct marketing campaigns using phone calls were conducted by the Portuguese banking institutions to promote a term deposit to its customers. Here, a term deposit is a financial product where a customer deposits an amount with the bank for a fixed period at a predetermined interest rate. One of the main objectives of these marketing campaigns is to encourage clients to subscribe to a term deposit and this can be a source of revenue for the bank.

The bank marketing dataset is a valuable resource for financial institutions to make data driven decisions in their marketing efforts. We intend to build predictive models (classification) to help the bank optimize its marketing campaigns and thereby improve subscription rates. This further extends to predicting whether a client will subscribe to a term deposit and gain insights into the efficacy of various marketing strategies.

Dataset Description:

The dataset, we have used for this project is from the UCI repository: Bank Marketing Data. In this dataset, we have ~41000 instances and 21 features which are categorical and numerical in nature. The target variable which is the subscription term deposit is 'Y' which is identified by whether a client has subscribed to a particular term deposit or not. This dataset gives us information about customer banking details.

Attributes	Description
Age	Age of customers
Job	Types of jobs
Marital	Marital status
Education	Levels of education

Default	Has credit in default?
Balance	Average yearly balance (in euros)
Housing	Has a housing loan?
Loan	Has a personal loan?
Contact	Contact types for communication
Day	Last contact day of the month
Month	Last contact month of the year
Duration	Last contact duration (in seconds)
Campaign	Number of contacts performed during this campaign and for this client
Pdays	Number of days that passed by since last contact
Previous	Number of contacts performed before this campaign
Poutcome	Outcomes of previous marketing campaign
Emp.var.rate	Employment variation rate
Cons.price.idx	Consumer price index
Cons.conf.idx	Consumer confidence index
Euribor3m.	Euribor 3-month rate
Nr.employed	Number of employees
y	Has client subscribed to term deposit? Y/N

There are few missing attribute values which are encoded with label “unknown”. We have assumed that such missing values are common in real-world scenarios, i.e there can be instances where we do have some customer data is missing and we have trained our model to handle/tackle such data.

Methods:

We have followed a systemic approach from preprocessing the data to facilitate training and making predictions. We implemented an end-to-end workflow starting with data preprocessing to model implementation and evaluation. Below are the steps we followed:

Imbalanced dataset – SMOTE:

SMOTE (Synthetic minority Over-sampling technique), is used to address the issue of class imbalance. Imbalance in classes usually happens when one class is underrepresented compared to the other class. Here, when we visualized the target 'Y' column, we noticed a major imbalance across the records in the two classes. We used SMOTE to balance the data before implementing the algorithms, we also did a comparison study between how the models learn differently by passing balanced and unbalanced data. This technique generates synthetic samples of the minority class 'Yes' by using the K- Nearest Neighbors approach. Here, we balance the class distribution to prevent the model from being biased toward the majority class 'No'.

Feature selection – Random Forest Classifier:

Feature selection is the process of choosing a subset of relevant features from a larger set to build a model. The main objective is to improve the performance of the model, reduce overfitting and enhance interpretability. To make the dataset feasible, we use random forest classifier as we have both categorical and continuous columns. We have selected the top 10 optimal features based on feature importance given by the algorithm. This ensemble method combines the predictions of multiple decision trees to improve the overall performance and generalization.

Models implemented:

Logistic regression: The main objective of logistic regression is to model the probability of a given input that belongs to a specific class. This is a straightforward algorithm and interpretable method used for binary classification problems. It aims to establish a linear relationship between the input features and the likelihood of belonging to a particular class (Yes, No). Here, it employs a sigmoid function for binary classification and the parameters are optimized using gradient descent.

Sigmoid Function:

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

Advantages:

1. Since the output can be expressed as a probability the results are easier to interpret

2. Simple and computationally efficient as it is suitable for large datasets.

Disadvantages:

1. Assumes there is a linear relationship between the input features and target.
2. Sensitive to outliers

Naïve bayes: Naïve Bayes Classifier is another classification technique based on the Bayes Theorem, assuming the independence among the predictors. In our dataset, we have both continuous and categorical features, and we make assumption about the distribution of the features which is continuous in nature after scaling and one hot encoding and use Gaussian Naïve Bayes.

Bayes theorem:

$$P(y|X) = \frac{P(X|y).P(X)}{P(y)}$$

GaussianDistribution:

$$P(X|Y = c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}}$$

Advantages:

1. Can handle large number of datapoints and predictors.
2. Converges faster than other models and not sensitive to irrelevant data.

Disadvantages:

1.Assumes independence between features (naïve assumption) which may not hold in some cases.

2. May or may not capture intricate relationships in the data

Support Vector Machine: The SVM is a powerful supervised learning model for classification. We use soft margin SVM due to presence of overlapping points where they might not be perfectly separable by a hyperplane and when there are outliers in the dataset. When there are outliers in the data, soft margin SVM introduces a penalty for misclassification allowing for a more robust model in the presence of such outliers. As SVM is less dependent on capturing every point on the training data set, the resulting model may be able to generalize the new data points that is unseen.

Function :

$$\begin{aligned} &\underset{w,\varepsilon_i}{\text{minimize}} && \frac{1}{2}||\boldsymbol{W}||_2^2 + C\sum_{i=1}^{n} \varepsilon_i \\ &\textit{Subject to} && y_i(W^T \emptyset(x_i)) \geq 1 - \varepsilon_i \qquad \forall_i = 1, \dots, n \\ & && \varepsilon_i \geq 0, \qquad \forall_i = 1, \dots, n \end{aligned}$$

Advantages:

1. Effective in high dimensional spaces
2. Memory efficient and can capture complex relationships in the data.

Disadvantages:

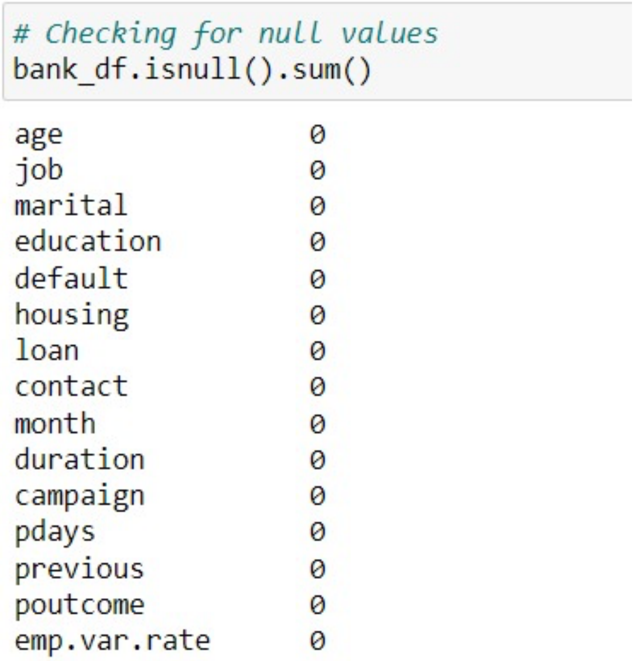
1. Computationally expensive with large dataset
2. Hard to interpret model in complex cases.

Data Exploration and Visualization:

Exploratory data analysis is performed to explore the various patterns and relationships in the features of the data. It helps us understand the data better by identifying the linear, non-linear relationships, checking for discrepancies in the data and familiarizing with the values in the input columns. Based on this exploration, we can understand our data better and learn what could be the next steps to prepare our data for model implementation.

Data Loading:

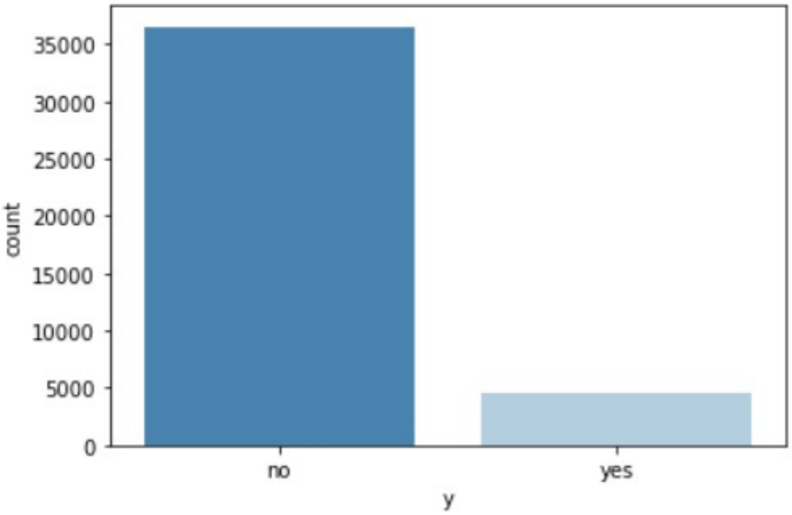
Firstly, we look for null values and unknown values in our dataset and found some columns with unknown values but as discussed earlier, we decided to retain them owing to real world scenarios.



```
# Checking for null values
bank_df.isnull().sum()
```

```
age             0
job             0
marital         0
education       0
default         0
housing         0
loan            0
contact         0
month           0
duration        0
campaign        0
pdays           0
previous        0
poutcome        0
emp.var.rate    0
```

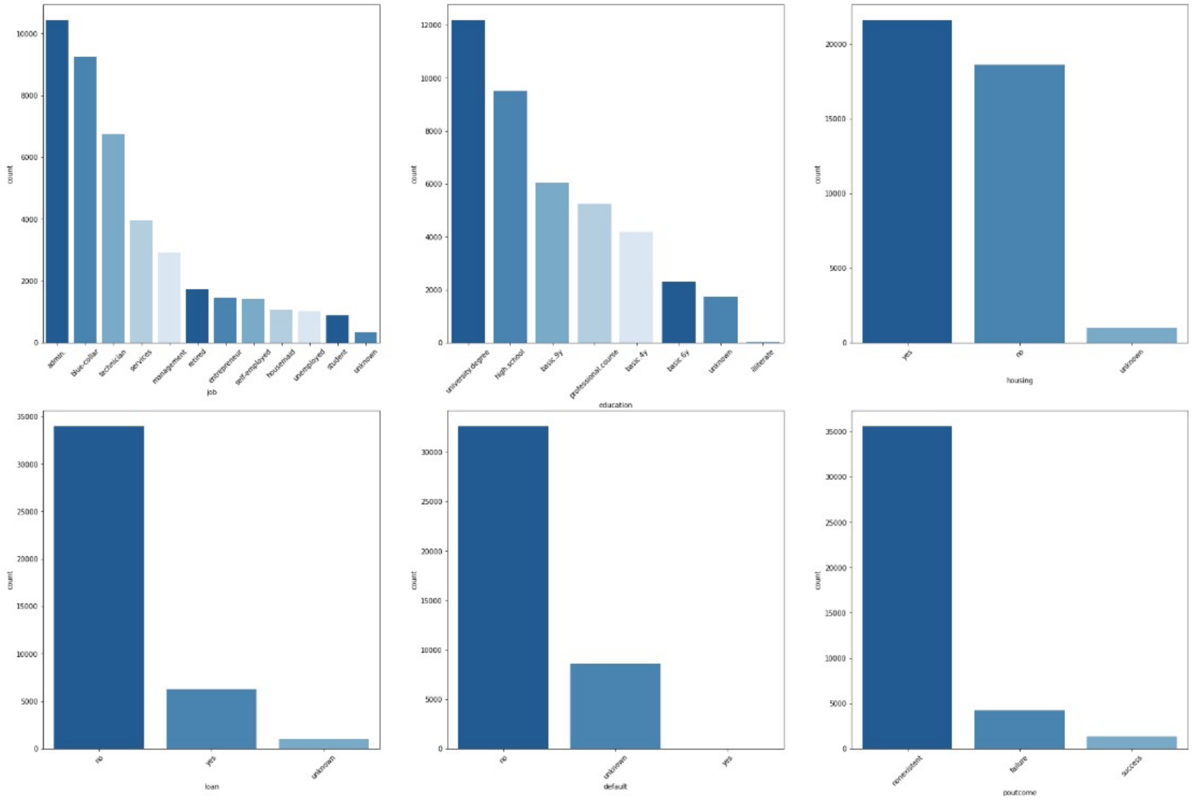
Next, we looked at the datatypes of all features and described statistics for numerical columns of the bank marketing dataset. Here, we observed that out dataset has a dimension of (41188, 11)
After this, we look at the distribution of classes in the ‘Y’ (Term deposit) and here, we observed that there is a major imbalance in our classes, that is the records of class ‘No’ are ~80x the records of class ‘Yes’



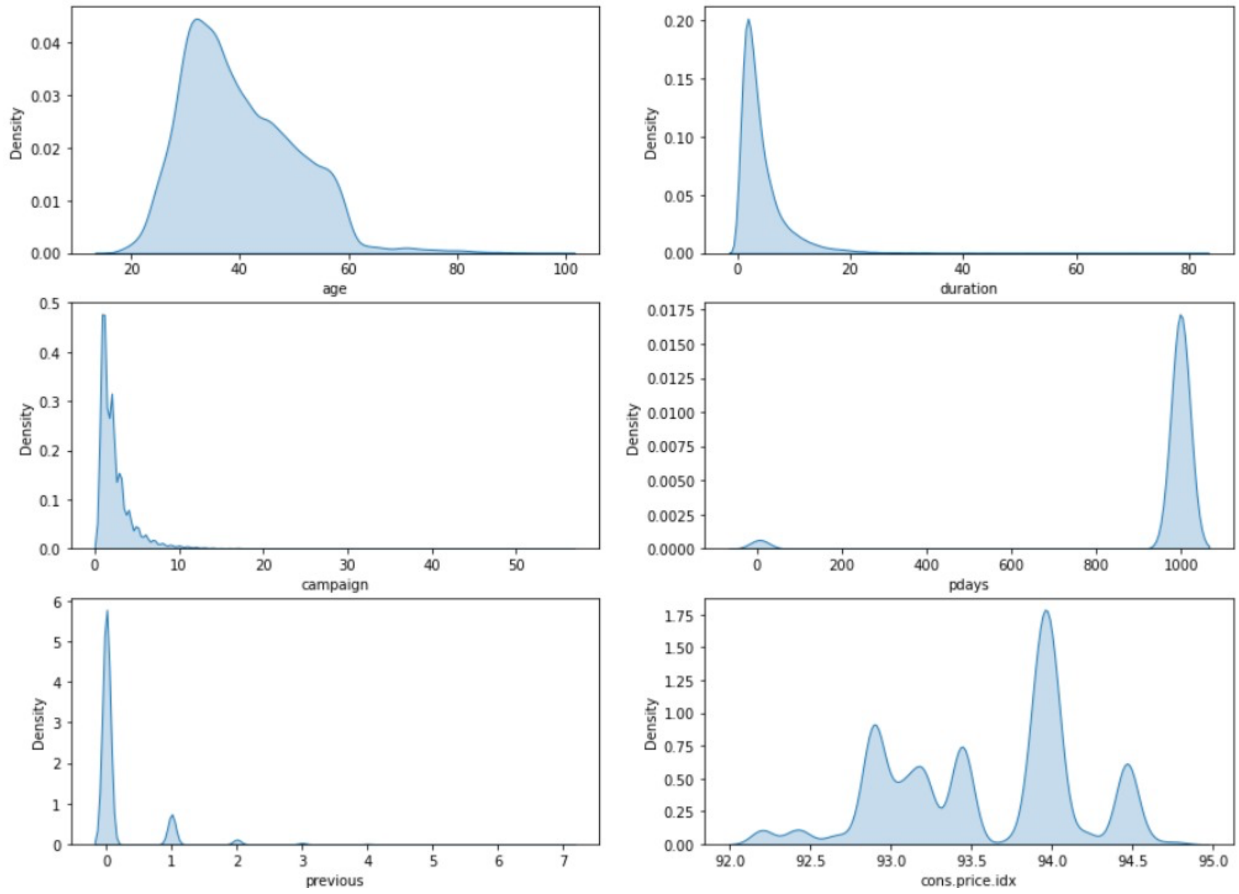
Univariate Analysis:

This is done to understand the distribution of continuous columns and obtain a count of categories in categorical columns. We looked at several features of both categorical and continuous nature. Below is a pictorial representation of the feature distribution looks like and we observed that our columns had a lot of categories, and we had a wide range of values. We noticed that for some categorical features, we had binary categories expect for job and education. For continuous, the range of the distribution is varied.

Categorical:

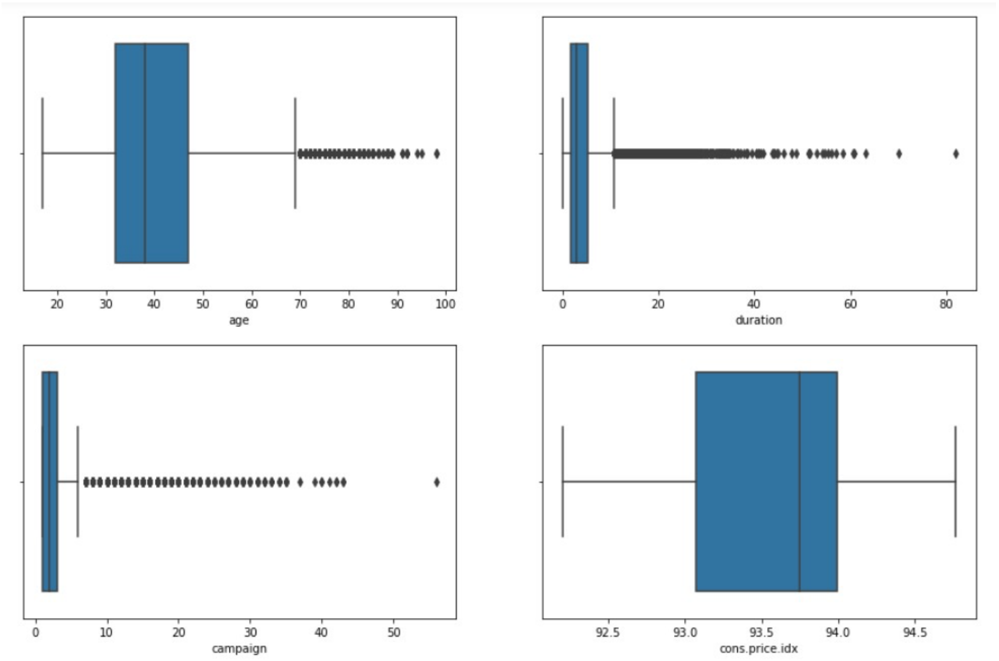


Cc



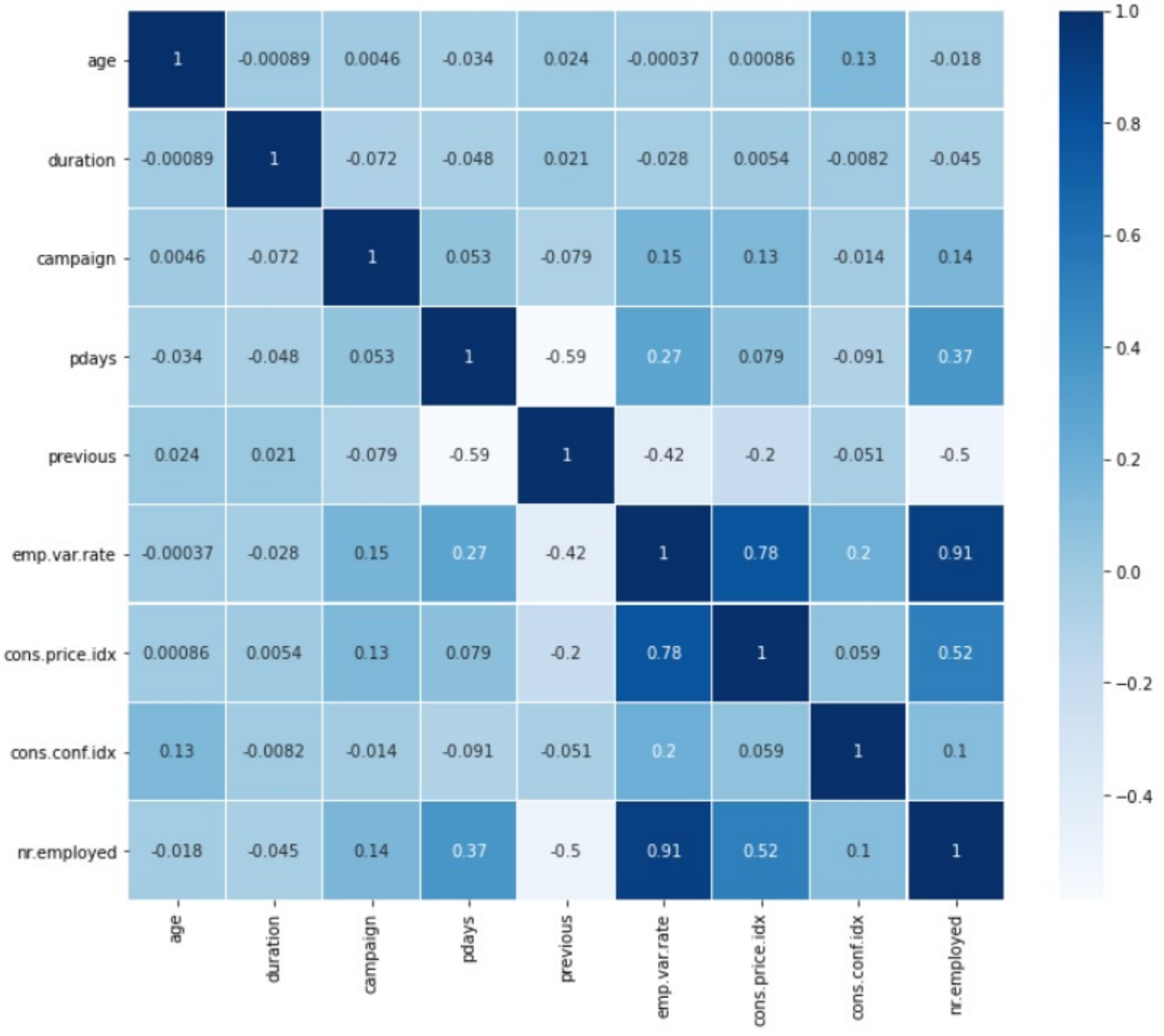
Checking for Outliers:

Here, we noticed that we have many outliers in columns like ‘age’, ‘duration’, ‘campaign’, etc. as shown in the image below, however we chose to retain the outliers as these datapoints can be important for our analysis and removing them could introduce bias.

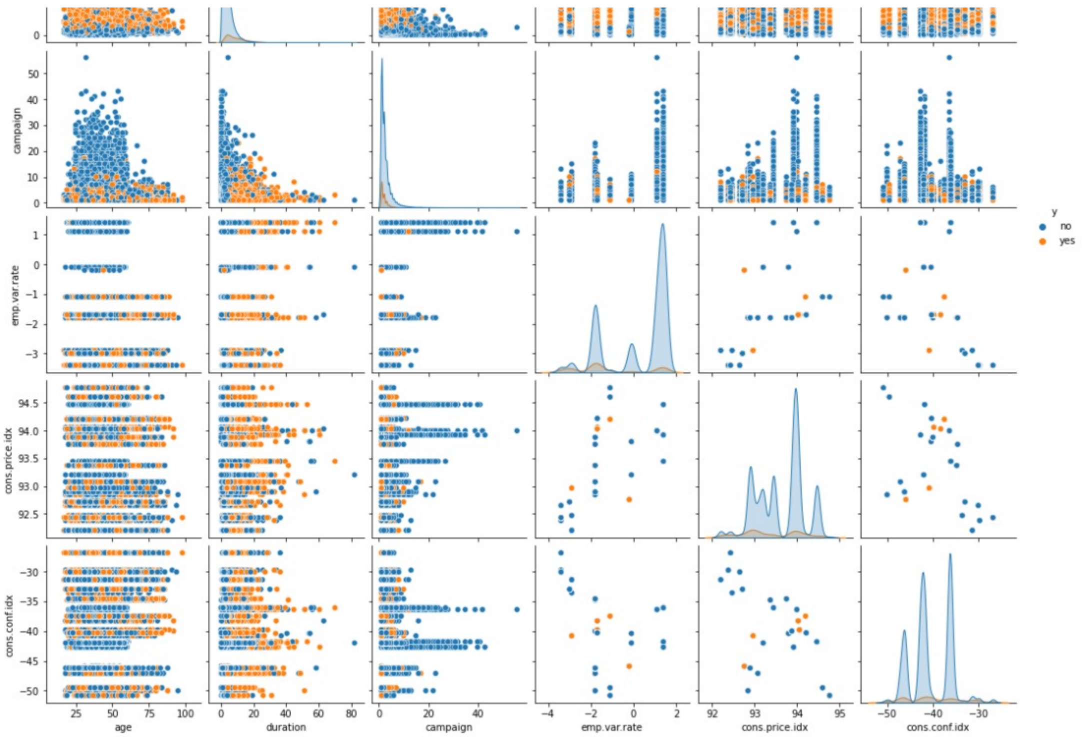


Bivariate/Multivariate Analysis:

Correlation plot was used to perform bivariate analysis between two sets of features to identify the numerical strength of association between them. From our plot, we observed that most of our columns have a weak correlation indicating that the set of features have an inverse relationship whereas columns like employee variation rate, consumer price index, consumer confidence index and number employed showed strong positive relations between each other. These features inform us on the confidence the clients have on the financial situation of the economy.

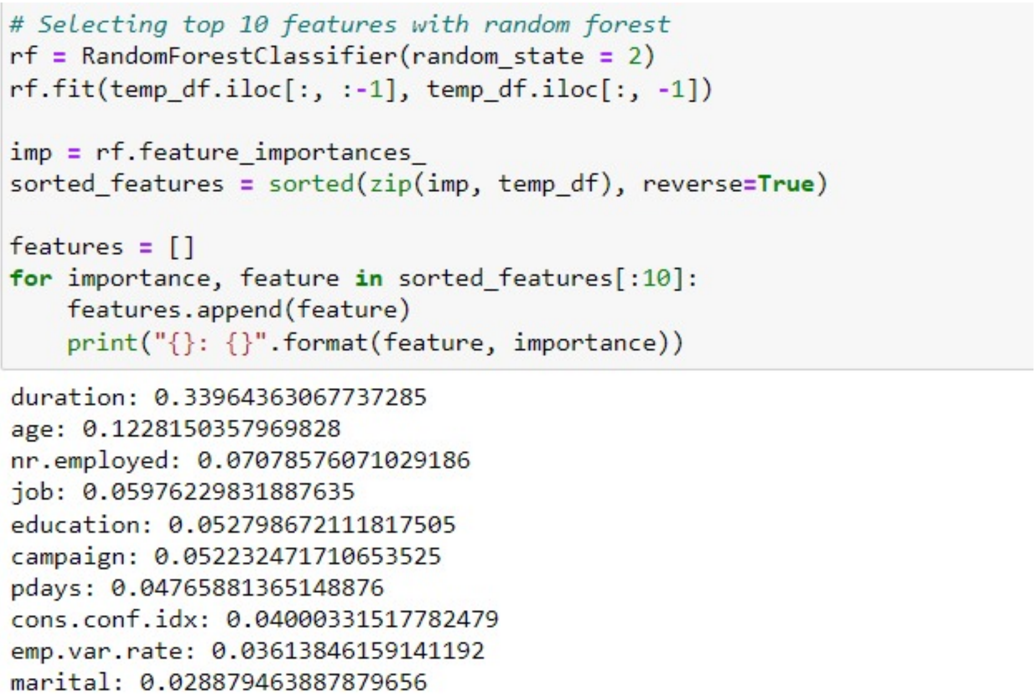


Pairplots are used for pairwise relationships to identify linear/ non-linear patterns between the features, in our case we have visualized the pair plot for continuous variables with respect to our target column. Based on this, we concluded that for most of the data, the classes were overlapping, indicating a non-linear relationship.



Data Processing and Feature Selection:

Before performing feature selection, we dropped columns ‘Day of the week’ and ‘euribor3m’ as they didn’t hold much significance. We converted the duration column, which was in seconds to minutes, for better interpretability. As we have categorical and continuous columns, we chose random forest classifier for feature selection. Based on the feature importance, we chose the top 10 optimal features. Below is the list of the features:



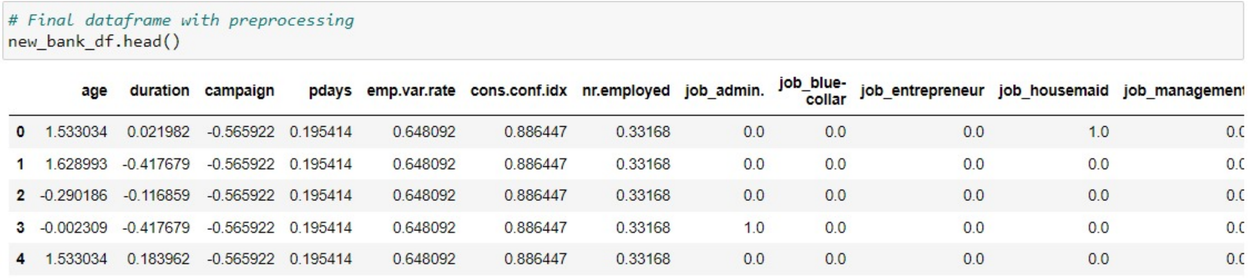
```
# Selecting top 10 features with random forest
rf = RandomForestClassifier(random_state = 2)
rf.fit(temp_df.iloc[:, :-1], temp_df.iloc[:, -1])

imp = rf.feature_importances_
sorted_features = sorted(zip(imp, temp_df), reverse=True)

features = []
for importance, feature in sorted_features[:10]:
    features.append(feature)
    print("{}: {}".format(feature, importance))
```

```
duration: 0.33964363067737285
age: 0.1228150357969828
nr.employed: 0.07078576071029186
job: 0.05976229831887635
education: 0.052798672111817505
campaign: 0.052232471710653525
pdays: 0.04765881365148876
cons.conf.idx: 0.04000331517782479
emp.var.rate: 0.03613846159141192
marital: 0.028879463887879656
```

We performed standardization and one hot encoding to preprocess the data and below is a snapshot of our preprocessed dataset.



```
# Final dataframe with preprocessing
new_bank_df.head()
```

	age	duration	campaign	pdays	emp.var.rate	cons.conf.idx	nr.employed	job_admin.	job_blue-collar	job_entrepreneur	job_housemaid	job_managemen
0	1.533034	0.021982	-0.565922	0.195414	0.648092	0.886447	0.33168	0.0	0.0	0.0	1.0	0.C
1	1.628993	-0.417679	-0.565922	0.195414	0.648092	0.886447	0.33168	0.0	0.0	0.0	0.0	0.C
2	-0.290186	-0.116859	-0.565922	0.195414	0.648092	0.886447	0.33168	0.0	0.0	0.0	0.0	0.C
3	-0.002309	-0.417679	-0.565922	0.195414	0.648092	0.886447	0.33168	1.0	0.0	0.0	0.0	0.C
4	1.533034	0.183962	-0.565922	0.195414	0.648092	0.886447	0.33168	0.0	0.0	0.0	0.0	0.C

To handle the imbalance in our dataset, we implement the SMOTE (synthetic minority oversampling technique) which generates instances of data to compensate for the low number of minority classes for which in addition to the standardized data; label encoded the target column and then, resampled the data.

Model Implementation and Evaluation:

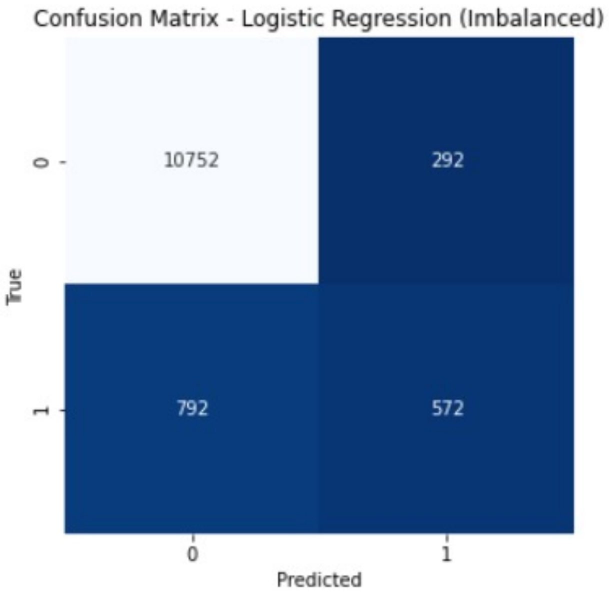
We chose to implement the below three models:

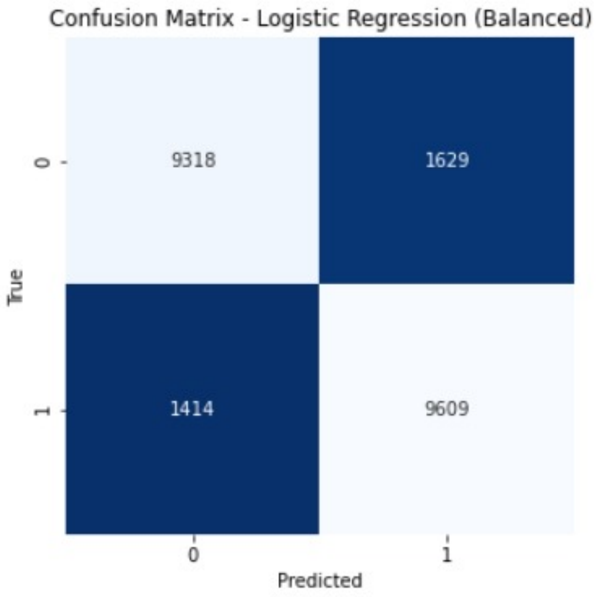
Logistic Regression:

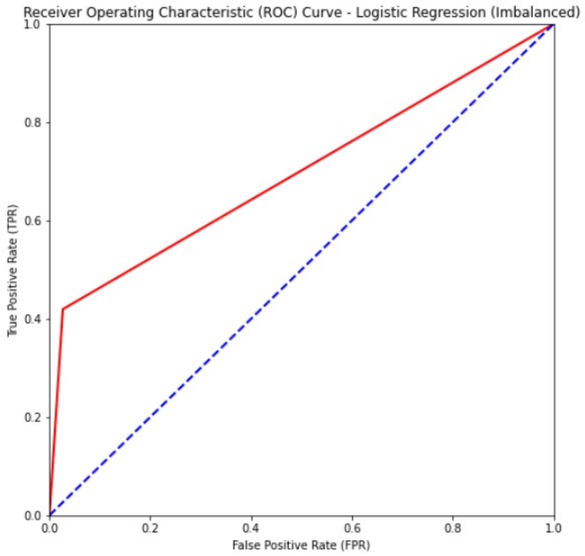
Logistic regression is a statistical method used for binary classification that models the probability of an event occurring as a function of one or more predictor variables. It uses a sigmoid function that transform a linear combination of input features into a range between [0,1]. Here the goal is to maximize the likelihood of observed outcomes and the likelihood function is derived from the probability distribution of sigmoid function.

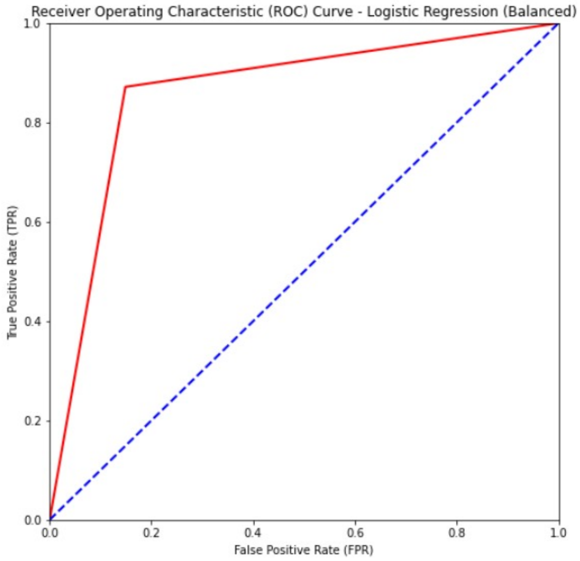
We calculate the cost function which gives the differences between the predicted probabilities and the actual outcomes. Now, we minimize the cost function by adjusting parameters like alpha (constant) using algorithms like gradient descent. Once these parameters are optimized, decision boundaries are established which divides the instances into class 0 and 1.

Here, we implemented logistic regression algorithm for both balanced and imbalance dataset. We observed that F1 score, which provides the balance precision and recall, as well as ROC_AUC score which is an overall measure of the model's ability to distinguish between classes, show higher values for balanced dataset in comparison to imbalanced dataset.



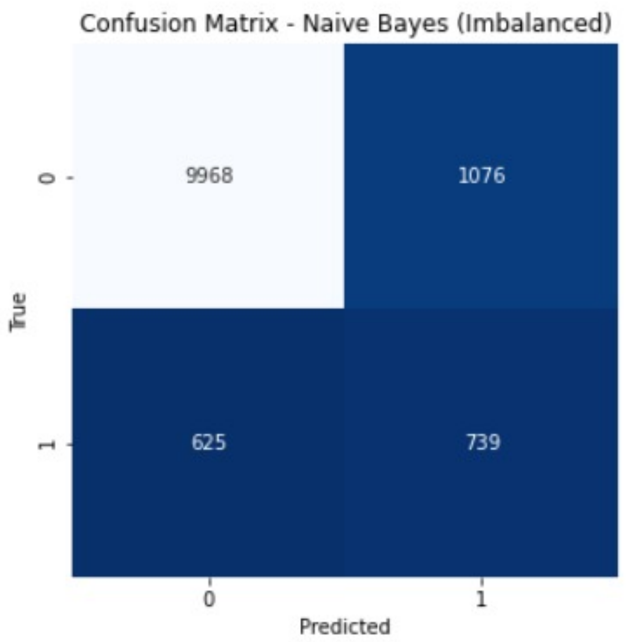


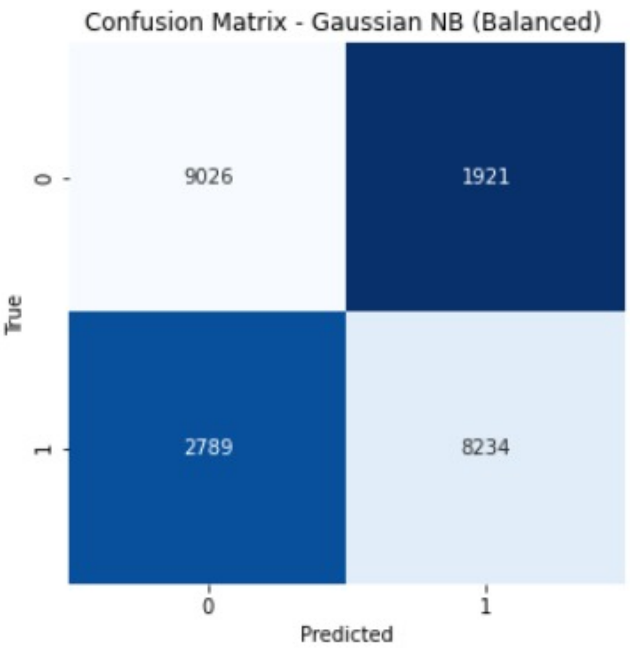


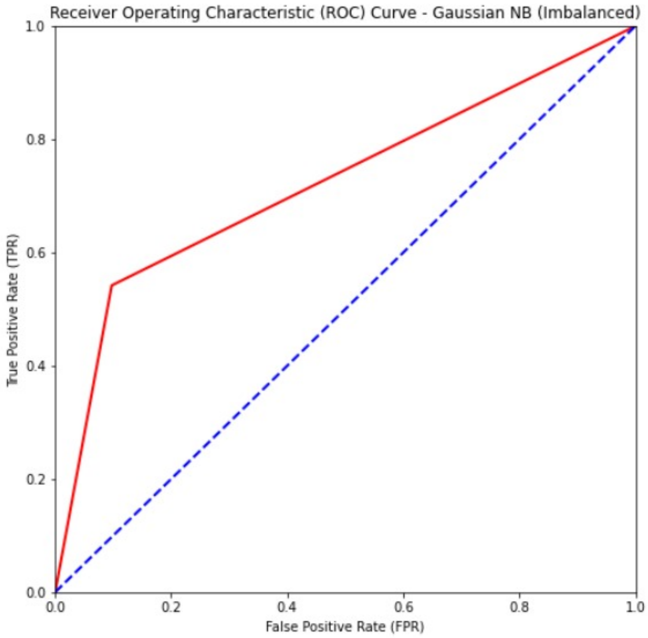


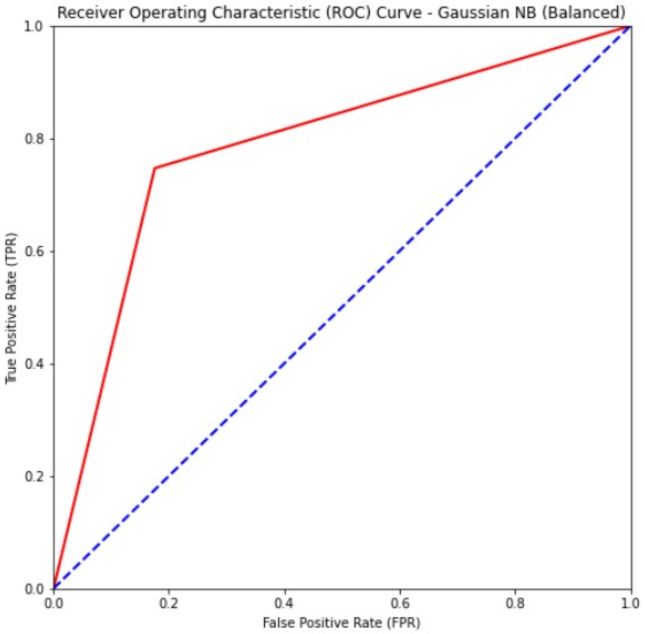
Gaussian Naïve Bayes:

Naïve Bayes is also a probabilistic classification algorithm based on Bayes theorem. It makes a naïve assumption that features are independent of each other. It also assumes that the continuous features of each class are nominally distributed. Here, the likelihood of observing a specific value for a class is calculated using probability density function of a Gaussian distribution. Next, we compute the prior probabilities and when we finally make the prediction, the likelihood of observed features for each class is calculated using Gaussian distribution parameters. Final class prediction is based on the class with the highest posterior probability.





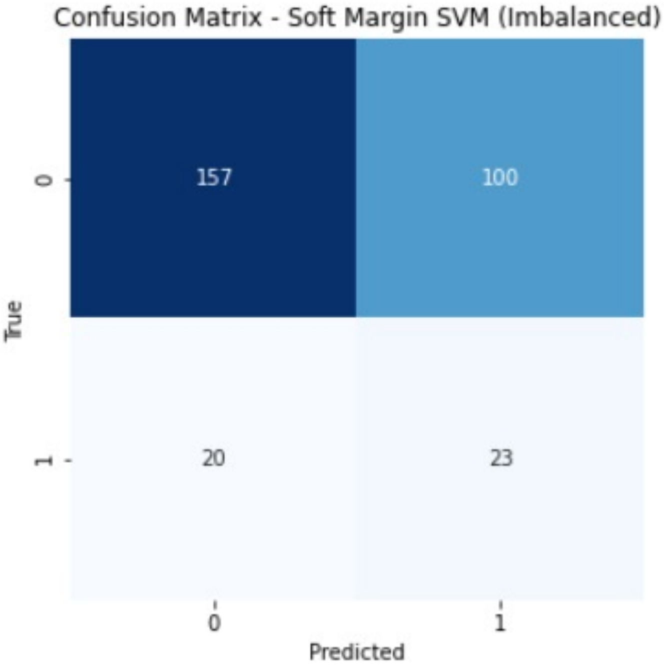


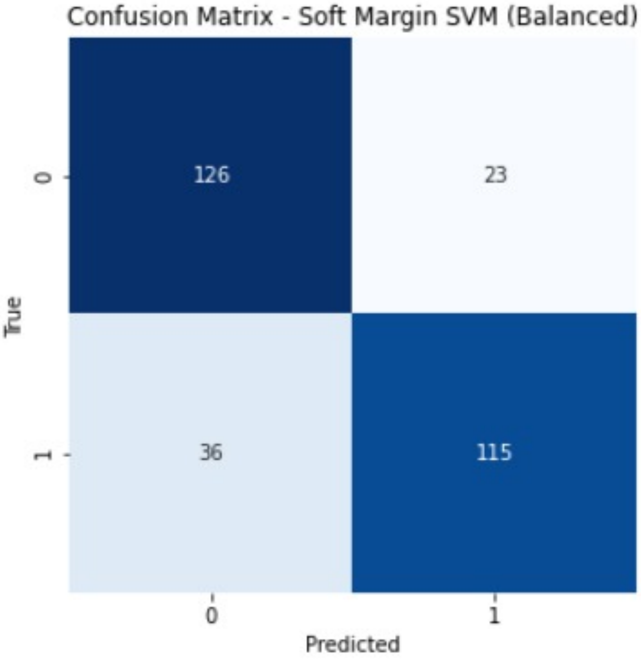


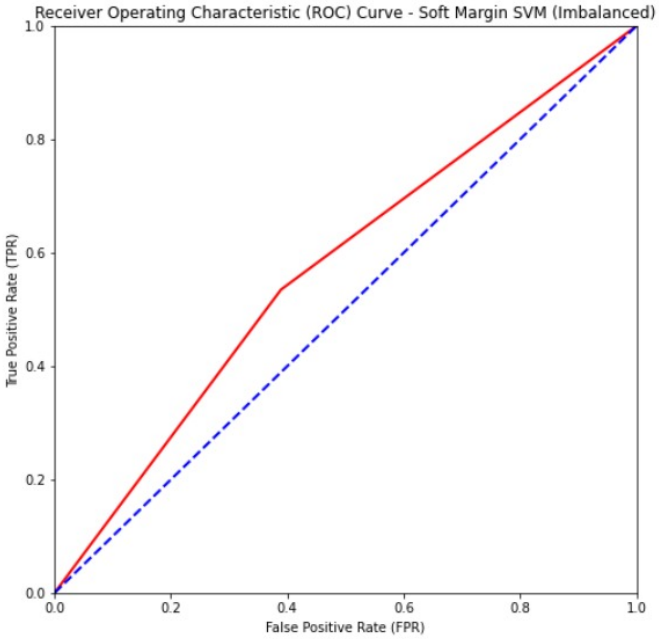
Support Vector Machine:

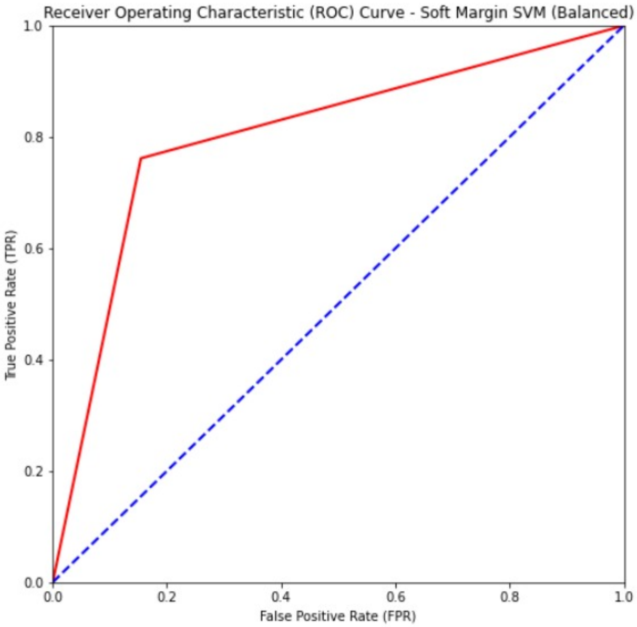
SVM's primary goal is to find a hyperplane that separates data points of various classes while maximizing the margin, which is the distance between the hyperplane and the nearest datapoints. Soft Margin support vector machine allows for some misclassification of datapoints. The main goal is to handle cases where the data might not be linearly separable. The original optimization problem is modified to include a cost parameter which takes care of the tradeoff between margin and misclassified files. This parameter includes a penalty for misclassification. To handle non-linear relationships, it can be extended using the kernel trick. Like Logistic Regression and

Gaussian Naïve Bayes, SVM also gives better results for balanced datapoints in comparison to imbalanced data.


Confusion Matrix - Soft Margin SVM (Imbalanced)


Confusion Matrix - Soft Margin SVM (Balanced)


Receiver Operating Characteristic (ROC) Curve - Soft Margin SVM (Imbalanced)


Receiver Operating Characteristic (ROC) Curve - Soft Margin SVM (Balanced)

Results:

Balanced Data (with SMOTE)

	Logistic Regression	Gaussian Naïve bayes	Support Vector Machine
Accuracy	0.86	0.79	0.80
F-1 score	0.86	0.78	0.80
Precision	0.86	0.81	0.83
Recall	0.87	0.75	0.76
ROC-AUC score	0.86	0.79	0.80

Imbalanced Data (without SMOTE)

	Logistic Regression	Gaussian Naïve bayes	Support Vector Machine
Accuracy	0.91	0.86	0.60
F-1 score	0.51	0.46	0.28
Precision	0.66	0.41	0.19
Recall	0.42	0.54	0.53
ROC-AUC score	0.70	0.72	0.57

From the above table, we can see that the performance of all algorithms with balanced dataset is much better in comparison to the imbalanced dataset. We can also see that F1 score and ROC_AUC indicate that the model is performing well, thereby accounting for improving the overall classification performance. Additionally, our baseline model is Gaussian Naïve Bayes as it provides the benchmark for gauging efficiency of other model due to its poor performance in comparison to the other models. Here, logistic regression is performing more stronger in comparison to others. This could be due to - high efficiency for binary classification, less prone to overfitting and easy to interpret.

Discussions:

In predicting term deposit subscriptions using the UCI bank marketing data, a comprehensive approach was adopted to address imbalances and optimize model performance. Synthetic Minority Over-sampling Technique (SMOTE) was employed to resolve class imbalance issue which generates synthetic instances for the minority class, enhancing model robustness. Random Forest classifier was leveraged for feature selection, utilizing its ability to discern feature importance potentially leading to dimensionality reduction. This ensured that the logistic regression model, which is chosen for its simplicity and interpretability, focused on the most influential attributes.
We took into consideration the bias-variance trade-off, by using SMOTE technique and the logistic regression model that aimed to capture the underlying patterns in the data without overfitting.

The decision to favor logistic regression, despite its simplicity is due to it’s interpretability and efficiency. By incorporating these techniques, the model aims to provide reliable predictions regarding term deposit subscriptions, navigating the complexities of real-world data while accounting for the nuanced trade-offs inherent in predictive modeling.