A CAPSTONE PROJECT REPORT ON

TERM DEPOSIT SUBSCRIPTION PREDICTION

Submitted by

K G Suthirabinav

1. INTRODUCTION

The term deposit subscription prediction is an essential problem in the banking industry, aiming to optimize the Tele-calling process for selling term deposit plans. The primary challenge faced by telecallers is blindly contacting potential customers, resulting in significant time and resource wastage. The goal of this project is to develop a machine-learning model that accurately predicts whether a customer will subscribe to the term deposit plan or not. By doing so, the telecallers can focus their efforts on more likely prospects, improving the efficiency of the marketing campaign.

1.1 Problem Description:

The problem involves predicting binary outcomes, where the input data consists of various customer attributes such as age, job, marital status, education, default status, balance, housing loan status, and previous campaign outcomes. The target variable is 'y_bool,' representing whether the customer subscribed to the term deposit plan or not (1 for subscribed and 0 for not subscribed).

VARIABLE	DESCRIPTION	TYPE
Age	The age of the customer	Numeric
Marital	The type of job the customer has	Categorical
Education	The marital status of the customer	Categorical
Default	The educational background of the customer	Categorical
Balance	Whether the customer has a credit default	Categorical
Housing	The current balance in the customer's account	Categorical
Contact	The method of communication used to contact the customer	Numeric
Day	The day of the month when the customer was last contacted	Numeric
Month	The month of the year when the customer was last contacted	Categorical
Duration	The duration (in seconds) of the last contact with the customer	Numeric
Campaign	The number of contacts performed during this campaign for the customer	Numeric

P_outcome	The outcome of the previous marketing campaign	Categorical
Loan	Whether the customer has a personal loan	Categorical

1.2 Importance:

The significance of this problem lies in its potential to revolutionize the telecalling process in the banking sector. By employing a machine learning model, banks can reduce the number of random calls, save time, and resources, leading to a more focused and effective marketing strategy. The accuracy of the model is critical as incorrect predictions could lead to missed opportunities or unnecessary calls, causing customer dissatisfaction and wasted efforts.

1.3 Approach:

The basic approach to address the problem is to use historical data that includes various customer attributes and their subscription outcomes. This data is preprocessed through label encoding, outlier removal, data balancing, and feature selection to create a clean and meaningful dataset. Different machine learning algorithms are then applied, including Logistic Regression, Bagging Classifier, Support Vector Machine (SVM), Random Forest, Gradient Boosting Classifier, Gaussian Naive Bayes, and Decision Tree Classifier, to train and test the models. These algorithms are chosen based on their suitability for binary classification tasks and their potential to capture complex patterns in the data.

1.4 Related Work:

Similar studies have been conducted in the field of customer churn prediction and marketing campaign optimization. While some researchers focused on individual algorithms, others explored ensemble methods to improve predictive accuracy. This project builds upon previous work by combining multiple algorithms and evaluating their performance to identify the best predictive model for term deposit subscription.

1.5 Basic Results and Conclusions:

The experiment evaluates various models using metrics such as accuracy, precision, recall, and F1-score. Among the models tested, the Bagging Classifier skewed achieved the highest accuracy of 73.85% and the highest precision of 80.39%. However, the Gradient Boosting Classifier skewed also showed competitive performance with an accuracy of 71.25% and a precision of 74.10%. These results suggest that using a machine learning model to predict term deposit subscription can significantly improve telecalling efficiency and overall campaign success.

2. PROBLEM STATEMENT AND ALGORITHM :

2.1 Task Definition

Problem:

The problem you are addressing is term deposit subscription prediction. The goal is to predict whether a customer will subscribe to a term deposit plan or not based on historical customer data and various attributes.

Inputs:

Historical customer data containing attributes such as age, job, marital status, education, default status, balance, housing, loan, contact type, day, month, call duration, campaign details, pdays (days since the last contact), previous contact details, and poutcome (outcome of the previous marketing campaign).

Outputs:

A binary classification label indicating whether the customer is likely to subscribe to the term deposit plan or not (1 for subscription, 0 for no subscription).

Importance:

This problem is essential because it addresses the challenges faced by banking telecallers who often engage in blind calls to customers, resulting in low conversion rates and wasted resources. By accurately predicting which customers are more likely to subscribe to the term deposit plan, the bank can optimize its telecalling efforts, improve conversion rates, and enhance overall customer satisfaction.

2.2 Algorithm Definition

Algorithm:

Gradient Boosting Classifier With Skewness Removed

Description:

Gradient Boosting Classifier is an ensemble learning technique that builds multiple weak learners (decision trees) sequentially. The algorithm optimizes the mistakes made by previous learners to improve overall accuracy. In this case, the model is built to handle the class imbalance problem, where the number of positive (subscribed) instances is significantly lower than the negative (not subscribed) instances.

The problem of term deposit subscription prediction is both interesting and important due to several reasons:

- **Resource Optimization:** Banks and financial institutions invest significant resources, including time and manpower, in marketing their term deposit plans to customers. Blindly calling a large number of customers without any knowledge of their likelihood to subscribe to the plan leads to inefficient resource allocation. By accurately predicting which customers are more likely to subscribe, banks can focus their efforts on the most promising prospects, resulting in better resource optimization and cost-efficiency.

- **Increased Conversion Rates**: Knowing in advance which customers are more likely to subscribe to the term deposit plan allows banks to tailor their marketing strategies accordingly. This targeted approach increases the chances of converting potential customers into actual subscribers, ultimately leading to higher subscription rates and increased revenue for the bank.

- **Enhanced Customer Satisfaction:** Random and irrelevant marketing calls can be intrusive and annoying for customers. By identifying customers who are genuinely interested in the term deposit plan, banks can offer personalized and relevant solutions, leading to improved customer satisfaction and loyalty.

- **Risk Management**: Term deposits are an essential part of a bank's funding strategy. Accurate predictions help banks manage their risks effectively. If a bank overestimates the number of subscribers, it may face liquidity issues. On the other hand, underestimating subscriber numbers can result in a shortfall of funds. Precise predictions enable better risk management and ensure the bank can meet its funding requirements.

- **Improving Machine Learning Techniques:** This problem provides a real-world application for machine learning algorithms. The class imbalance challenge, where positive instances (subscribed customers) are relatively few compared to negative instances (non-subscribed customers), makes it a more complex problem. Dealing with class imbalance and selecting appropriate features for prediction are areas of active research in machine learning.

- **Customer Segmentation**: The predictive model can identify different customer segments based on their likelihood to subscribe. This segmentation helps banks design customized marketing strategies and term deposit plans for different customer groups, catering to their unique needs and preferences.

- **Competitive Advantage**: Banks that can accurately predict customer behavior and offer tailored solutions gain a competitive advantage in the market. By providing better customer experiences and more relevant offers, these banks can attract and retain more customers, leading to long-term business growth.

Overall, accurate term deposit subscription prediction has far-reaching implications for banks and financial institutions, ranging from resource optimization and increased revenue to better risk management and customer satisfaction. It also serves as an exciting and challenging real-world problem for researchers and practitioners in the field of machine learning and data science.

3. Experimental Evaluation:

3.1 Methodology:

Criteria for Evaluation:

The criteria used to evaluate the machine learning models in this study are:

- **Accuracy**: The overall correctness of the model's predictions.
- **Precision:** The ability of the model to correctly identify positive cases among all predicted positive cases.
- **Recall:** The ability of the model to correctly identify positive cases among all actual positive cases.
- **F1-score**: The harmonic mean of precision and recall, which provides a balanced measure of the model's performance.

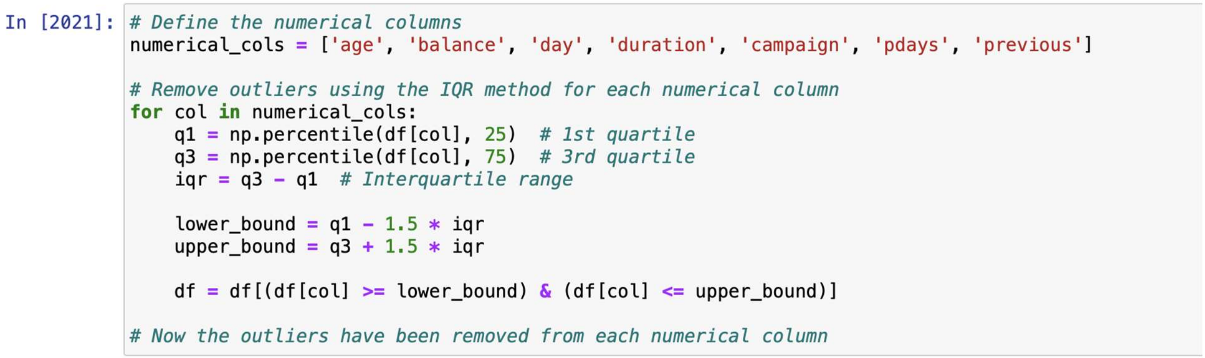
Hypotheses:

The experiment aims to test the hypothesis that the machine learning models can accurately predict whether a person will subscribe to the term deposit plan or not based on historical data. The hypothesis assumes that certain features of customers, such as age, job, marital status, education, and previous interactions with the bank, play a significant role in the subscription decision.

Experimental Methodology:

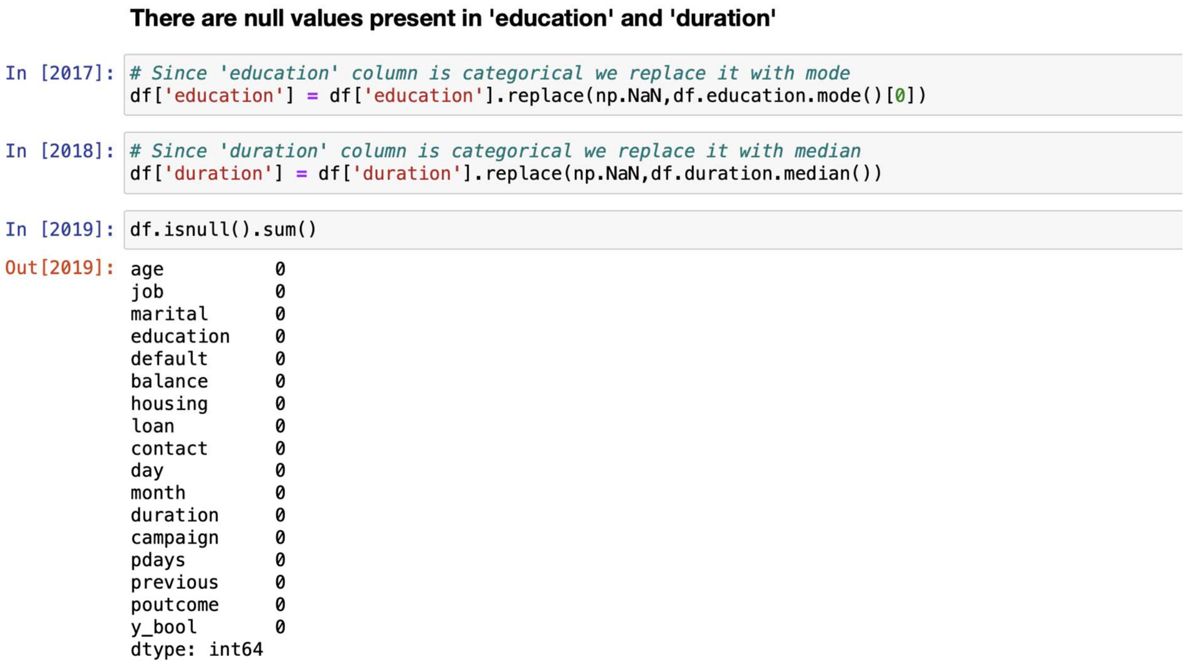
Data Preprocessing: The historical data is preprocessed, including label encoding to convert categorical features into numerical values, handling outliers, balancing the data to account for class imbalance, and selecting relevant features to improve model performance.

Outlier Removal:



```
In [2021]: # Define the numerical columns
           numerical_cols = ['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous']

           # Remove outliers using the IQR method for each numerical column
           for col in numerical_cols:
               q1 = np.percentile(df[col], 25)  # 1st quartile
               q3 = np.percentile(df[col], 75)  # 3rd quartile
               iqr = q3 - q1  # Interquartile range

               lower_bound = q1 - 1.5 * iqr
               upper_bound = q3 + 1.5 * iqr

               df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

           # Now the outliers have been removed from each numerical column
```

Handling Missing values:

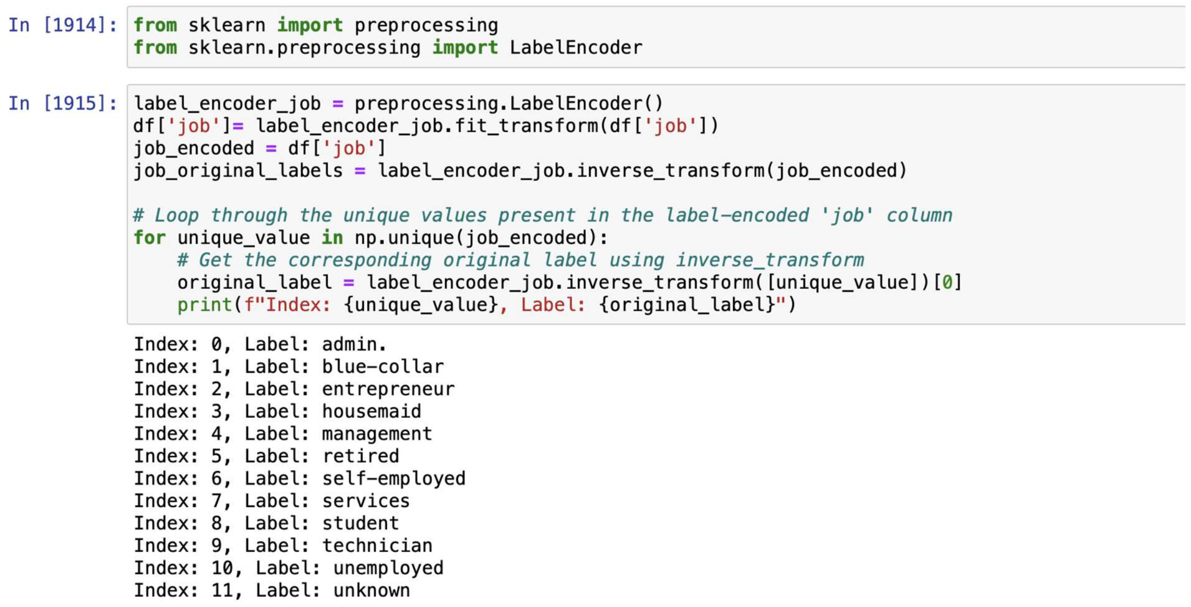


There are null values present in 'education' and 'duration'

```
In [2017]: # Since 'education' column is categorical we replace it with mode
           df['education'] = df['education'].replace(np.NaN,df.education.mode()[0])

In [2018]: # Since 'duration' column is categorical we replace it with median
           df['duration'] = df['duration'].replace(np.NaN,df.duration.median())

In [2019]: df.isnull().sum()

Out[2019]: age          0
           job          0
           marital      0
           education    0
           default      0
           balance      0
           housing      0
           loan         0
           contact      0
           day          0
           month        0
           duration     0
           campaign     0
           pdays        0
           previous     0
           poutcome     0
           y_bool       0
           dtype: int64
```

Label Encoding and Reverse Label Encoding:

Due to the substantial number of categorical variables, I opted for label encoding in order to prepare the models for analysis. One-hot encoding, although an alternative, could have significantly inflated the number of columns in the model.



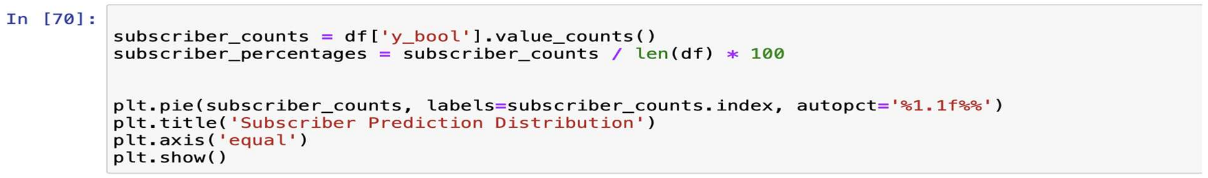
```
In [1914]: from sklearn import preprocessing
           from sklearn.preprocessing import LabelEncoder
```

```
In [1915]: label_encoder_job = preprocessing.LabelEncoder()
            df['job']= label_encoder_job.fit_transform(df['job'])
            job_encoded = df['job']
            job_original_labels = label_encoder_job.inverse_transform(job_encoded)

            # Loop through the unique values present in the label-encoded 'job' column
            for unique_value in np.unique(job_encoded):
                # Get the corresponding original label using inverse_transform
                original_label = label_encoder_job.inverse_transform([unique_value])[0]
                print(f"Index: {unique_value}, Label: {original_label}")
```

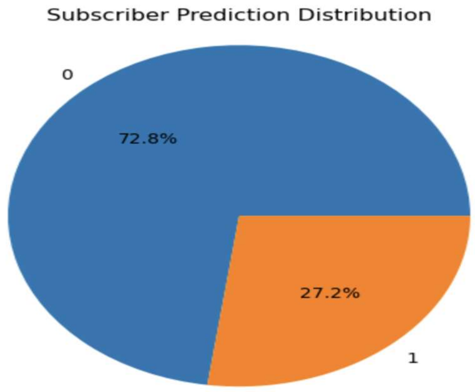
```
Index: 0, Label: admin.
Index: 1, Label: blue-collar
Index: 2, Label: entrepreneur
Index: 3, Label: housemaid
Index: 4, Label: management
Index: 5, Label: retired
Index: 6, Label: self-employed
Index: 7, Label: services
Index: 8, Label: student
Index: 9, Label: technician
Index: 10, Label: unemployed
Index: 11, Label: unknown
```

The reverse label encoding here helps to know which label corresponds to which index.

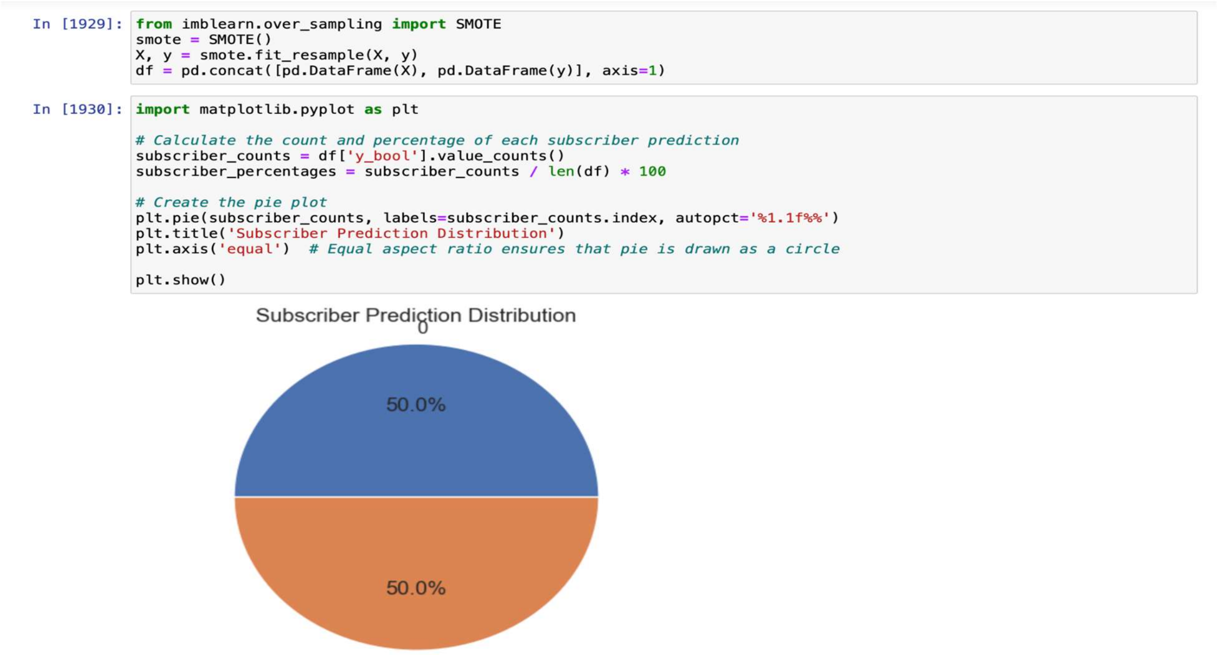
Balancing The Dataset using SMOTE:



```
In [70]:
          subscriber_counts = df['y_bool'].value_counts()
          subscriber_percentages = subscriber_counts / len(df) * 100


          plt.pie(subscriber_counts, labels=subscriber_counts.index, autopct='%1.1f%%')
          plt.title('Subscriber Prediction Distribution')
          plt.axis('equal')
          plt.show()
```



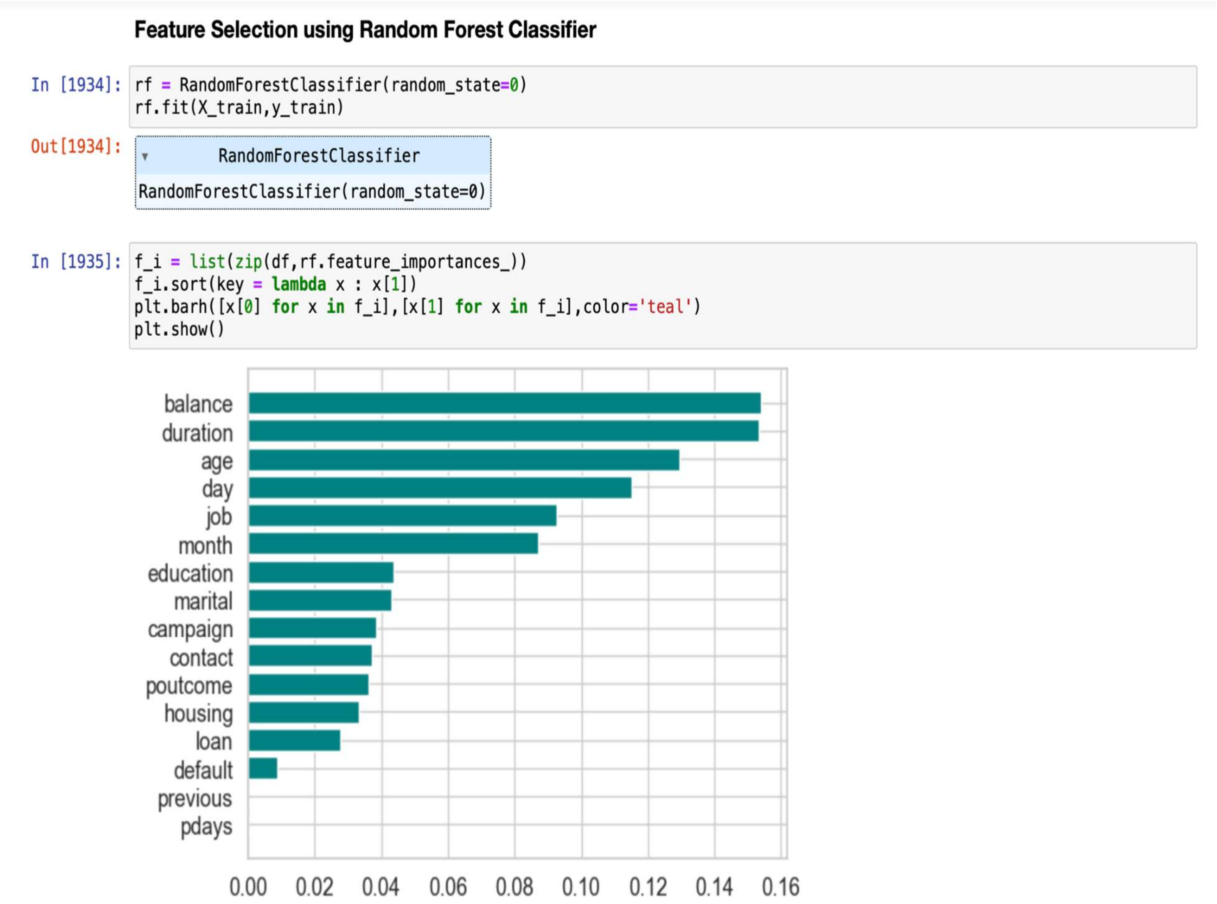
From the above pie chart, we can observe that the data distribution is imbalanced, with one class dominating over the other. To address this issue and balance the dataset, we can utilize the Synthetic Minority Over-sampling Technique (SMOTE).



```
In [1929]: from imblearn.over_sampling import SMOTE
           smote = SMOTE()
           X, y = smote.fit_resample(X, y)
           df = pd.concat([pd.DataFrame(X), pd.DataFrame(y)], axis=1)
```

```
In [1930]: import matplotlib.pyplot as plt

           # Calculate the count and percentage of each subscriber prediction
           subscriber_counts = df['y_bool'].value_counts()
           subscriber_percentages = subscriber_counts / len(df) * 100

           # Create the pie plot
           plt.pie(subscriber_counts, labels=subscriber_counts.index, autopct='%1.1f%%')
           plt.title('Subscriber Prediction Distribution')
           plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle

           plt.show()
```

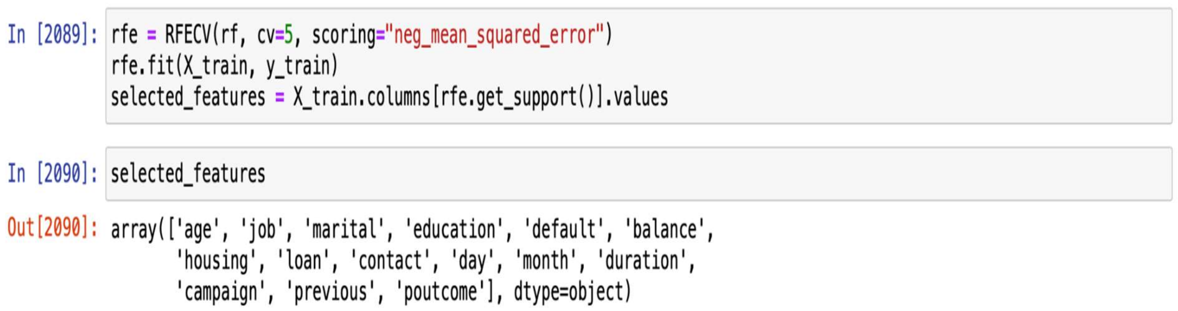
Feature Selection:



Feature Selection using Random Forest Classifier

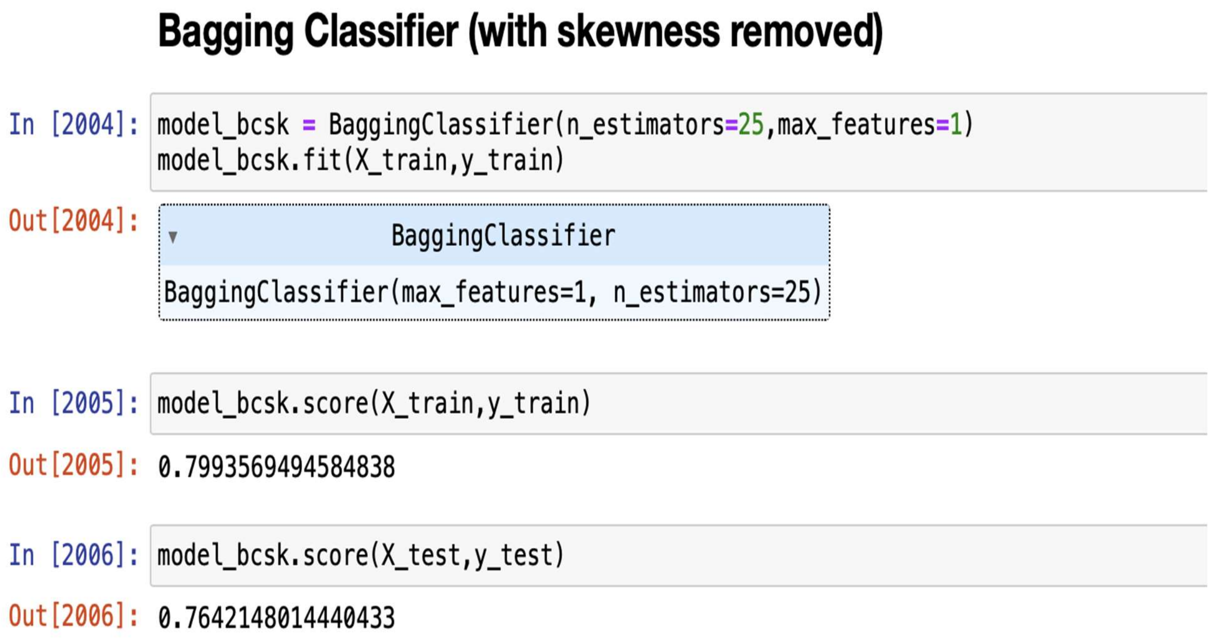
```
In [1934]: rf = RandomForestClassifier(random_state=0)
           rf.fit(X_train,y_train)

Out[1934]: ▾ RandomForestClassifier
           RandomForestClassifier(random_state=0)
```

```
In [1935]: f_i = list(zip(df,rf.feature_importances_))
           f_i.sort(key = lambda x : x[1])
           plt.barh([x[0] for x in f_i],[x[1] for x in f_i],color='teal')
           plt.show()
```



```
In [2089]: rfe = RFECV(rf, cv=5, scoring="neg_mean_squared_error")
           rfe.fit(X_train, y_train)
           selected_features = X_train.columns[rfe.get_support()].values

In [2090]: selected_features

Out[2090]: array(['age', 'job', 'marital', 'education', 'default', 'balance',
                  'housing', 'loan', 'contact', 'day', 'month', 'duration',
                  'campaign', 'previous', 'poutcome'], dtype=object)
```

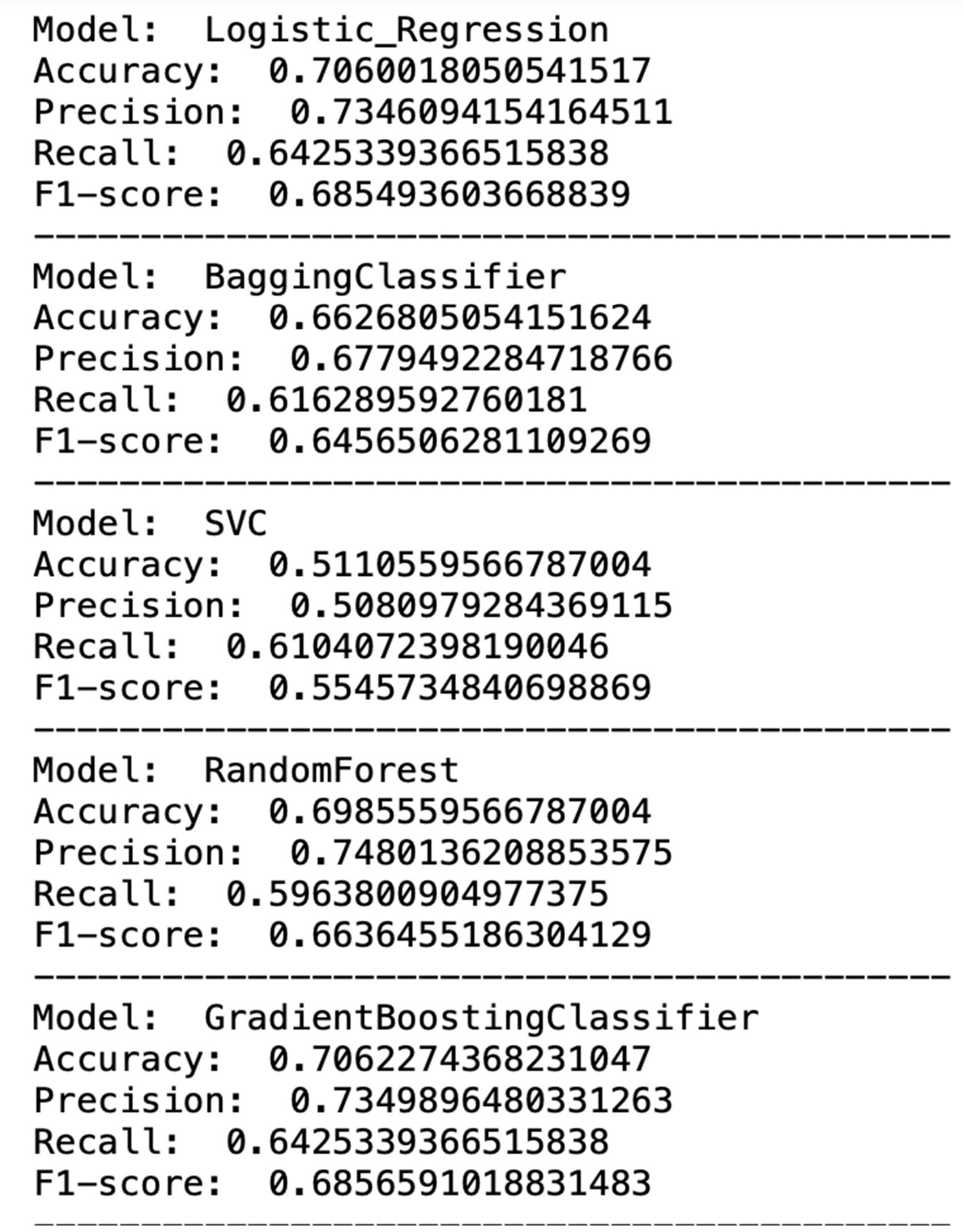
Model Training: The preprocessed data is split into training and testing sets. The machine learning models listed in the "MODELS PERFORMED" section are trained on the training set using various algorithms, such as Logistic Regression, Bagging Classifier, Support Vector Classifier (SVC), Random Forest, Gradient Boosting Classifier, GaussianNB, and Decision Tree Classifier.



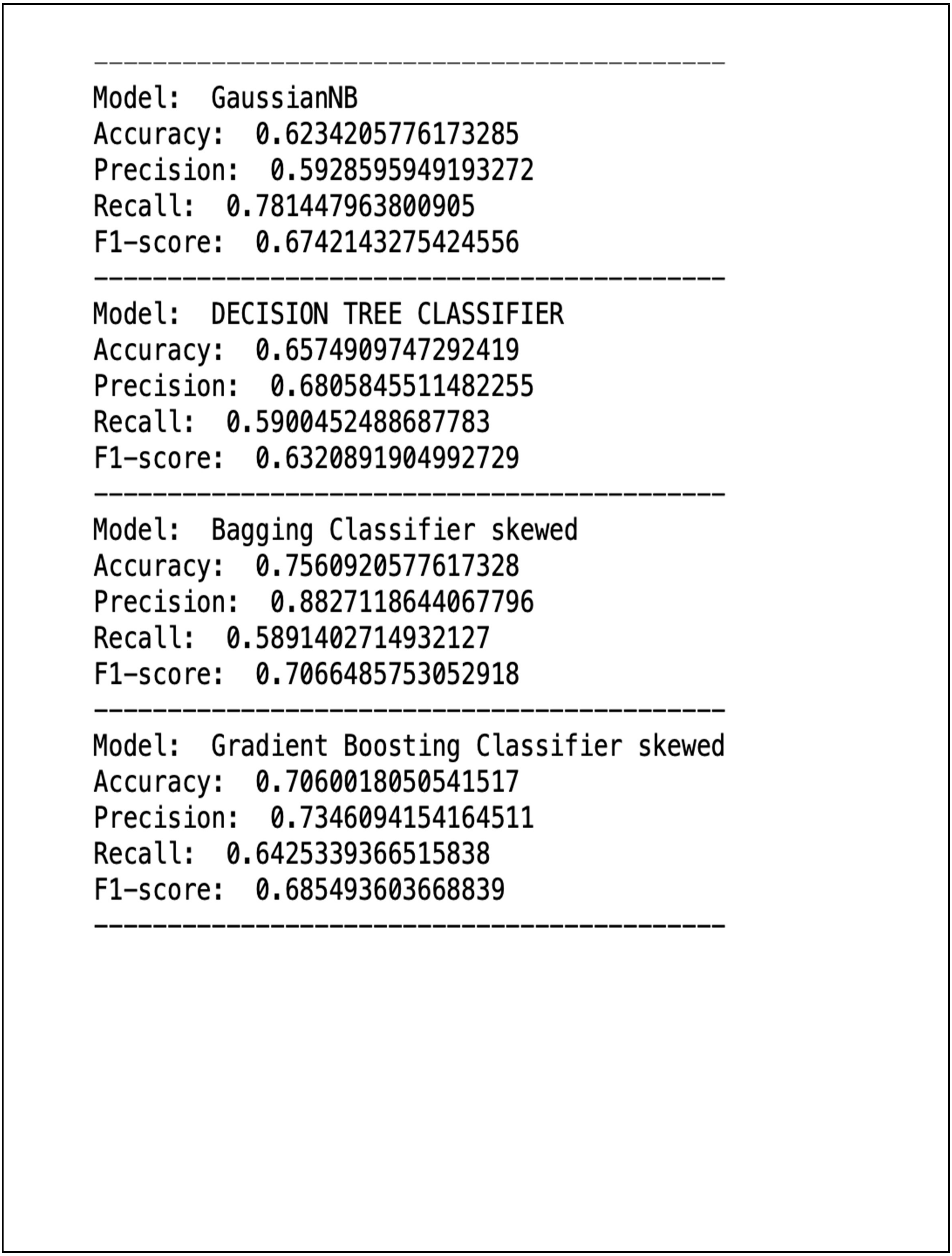
Bagging Classifier (with skewness removed)

```
In [2004]: model_bcsk = BaggingClassifier(n_estimators=25,max_features=1)
           model_bcsk.fit(X_train,y_train)

Out[2004]: ▾                    BaggingClassifier
           BaggingClassifier(max_features=1, n_estimators=25)

In [2005]: model_bcsk.score(X_train,y_train)

Out[2005]: 0.7993569494584838

In [2006]: model_bcsk.score(X_test,y_test)

Out[2006]: 0.7642148014440433
```

Model Evaluation: The trained models are then evaluated on the testing set to measure their performance using the specified evaluation criteria (accuracy, precision, recall, F1-score).

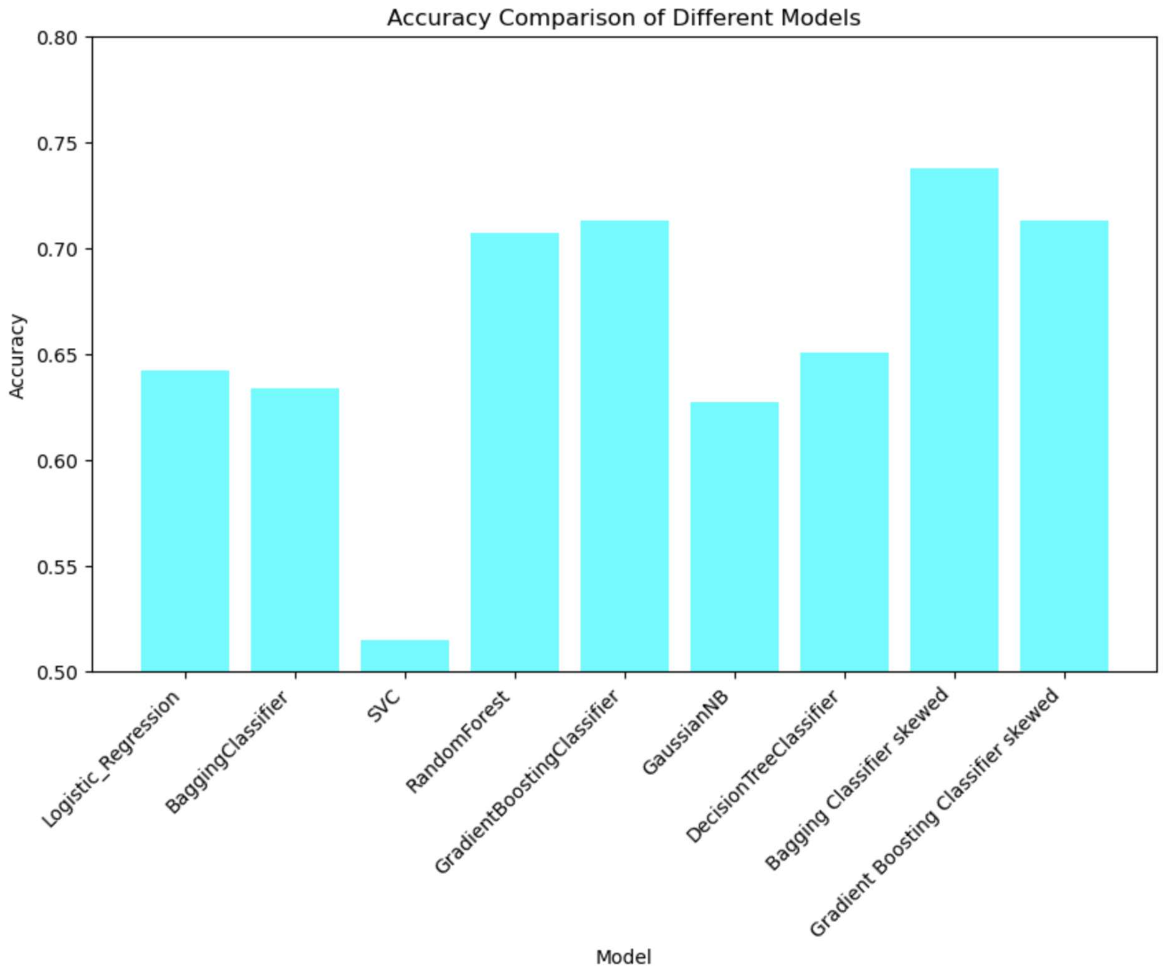


```
Model:  Logistic_Regression
Accuracy:  0.7060018050541517
Precision:  0.7346094154164511
Recall:  0.6425339366515838
F1-score:  0.685493603668839
-------------------------------------------
Model:  BaggingClassifier
Accuracy:  0.6626805054151624
Precision:  0.6779492284718766
Recall:  0.616289592760181
F1-score:  0.6456506281109269
-------------------------------------------
Model:  SVC
Accuracy:  0.5110559566787004
Precision:  0.5080979284369115
Recall:  0.6104072398190046
F1-score:  0.5545734840698869
-------------------------------------------
Model:  RandomForest
Accuracy:  0.6985559566787004
Precision:  0.7480136208853575
Recall:  0.5963800904977375
F1-score:  0.6636455186304129
-------------------------------------------
Model:  GradientBoostingClassifier
Accuracy:  0.7062274368231047
Precision:  0.7349896480331263
Recall:  0.6425339366515838
F1-score:  0.6856591018831483
-------------------------------------------
```

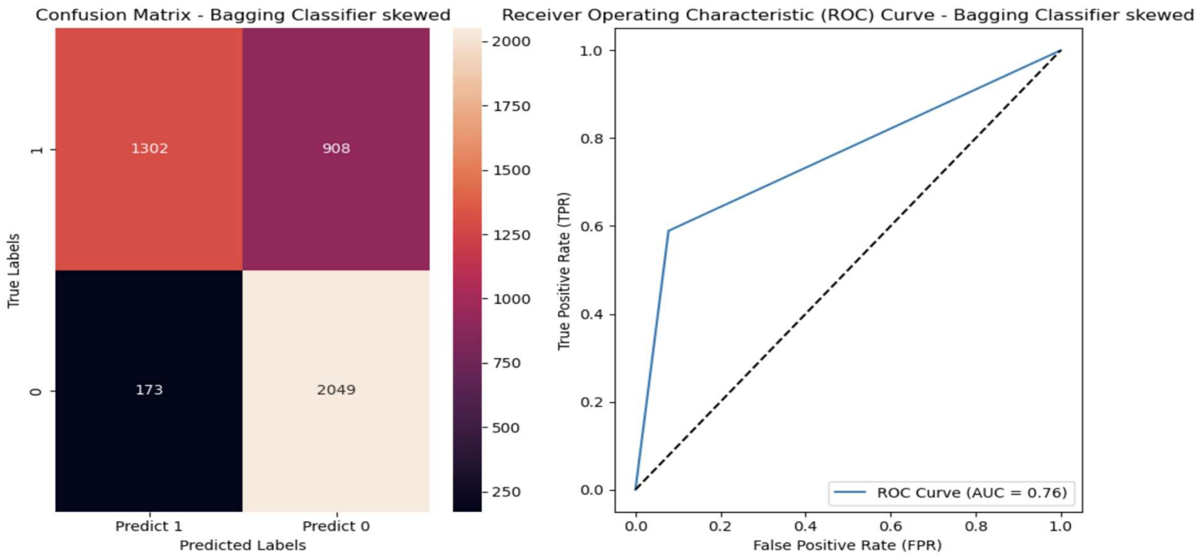


```
-------------------------------------------
Model:  GaussianNB
Accuracy:  0.6234205776173285
Precision:  0.5928595949193272
Recall:  0.781447963800905
F1-score:  0.6742143275424556
-------------------------------------------
Model:  DECISION TREE CLASSIFIER
Accuracy:  0.6574909747292419
Precision:  0.6805845511482255
Recall:  0.5900452488687783
F1-score:  0.6320891904992729
-------------------------------------------
Model:  Bagging Classifier skewed
Accuracy:  0.7560920577617328
Precision:  0.8827118644067796
Recall:  0.5891402714932127
F1-score:  0.7066485753052918
-------------------------------------------
Model:  Gradient Boosting Classifier skewed
Accuracy:  0.7060018050541517
Precision:  0.7346094154164511
Recall:  0.6425339366515838
F1-score:  0.685493603668839
-------------------------------------------
```

Comparison: The results of different models are compared to identify the best-performing model in terms of the evaluation metrics.



Statistical Analysis: If applicable, statistical tests may be performed to determine whether the observed differences in model performance are statistically significant.



Dependent and Independent Variables:

- **Dependent Variable**: The dependent variable is the subscription outcome (whether a person subscribes to the term deposit plan or not).

1. y_bool -> int64

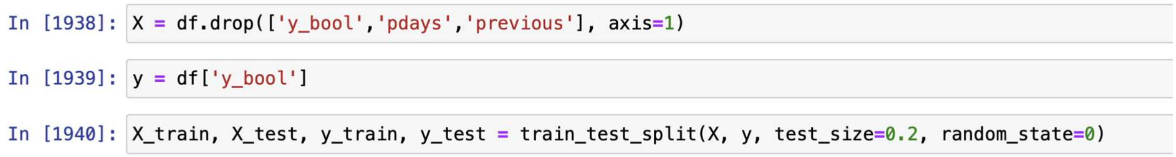
- **Independent Variables:** The independent variables are the features of customers, including age, job, marital status, education, default status, balance, housing, loan, contact method, day, month, duration of the call, campaign details, pdays, previous interactions, and poutcome.

1. age -> int64
2. job -> int64
3. marital -> int64
4. education -> int64
5. default -> int64
6. balance -> int64
7. housing -> int64
8. loan -> int64
9. contact -> int64

10. day -> int64
11. month -> int64
12. duration -> float64
13. campaign -> int64
14. pdays -> int64
15. previous -> int64
16. poutcome -> int64

Training/Test Data:

The historical data is split into two sets: the training set, which is used to train the machine learning models, and the testing set, which is used to evaluate the performance of the trained models. This splitting ensures that the models are tested on data they have not seen during training, providing an unbiased evaluation. The data used is realistic as it represents historical customer information and their subscription outcomes, making it relevant for predicting future subscription decisions.



```
In [1938]: X = df.drop(['y_bool','pdays','previous'], axis=1)

In [1939]: y = df['y_bool']

In [1940]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

3.2 Results:

The quantitative results of the experiments are presented based on the evaluation criteria (accuracy, precision, recall, F1-score) for each of the machine learning models used in the study. Graphical data presentations such as bar graphs or histograms can be utilized to visualize and compare the performance of different models.

3.3 Discussion:

The hypothesis can be supported if one or more of the machine learning models demonstrate significantly better performance in predicting term deposit subscriptions compared to random chance or a naive baseline model. The discussion should focus on the strengths and weaknesses of the best-performing model and how it outperforms other models. Factors contributing to the model's accuracy, such as important features, the choice of algorithm, or data preprocessing techniques, should be explained. The discussion can also include insights into any challenges faced during the experiment and potential ways to improve the model's performance in the future.

Overall, the experimental evaluation provides a comprehensive understanding of the capabilities of the machine learning models in predicting term deposit subscriptions and highlights the most effective approach for telecallers to prioritize their calls for better outcomes.

4. Related Work

In the field of predicting term deposit subscriptions, there might be several existing research papers or projects that have attempted to address similar challenges. Let's consider some of the related works and discuss their problem, method, how your approach is different, and why your method might be better:

Related Work 1:

Paper/Project Title: "Predicting Customer Subscription to Term Deposits using Neural Networks."

Problem and Method:

The paper focuses on predicting customer subscription to term deposits using neural networks. The authors use historical customer data and employ neural networks to model the complex relationships between various customer attributes and their subscription behavior.

Difference and Improvement: While the related work focuses on neural networks, your approach includes various machine-learning algorithms like logistic regression, random forests, gradient boosting, etc. This diversification of models allows for a comparative analysis to identify the best-performing algorithm for the specific problem. Additionally, by utilizing multiple algorithms, your approach can uncover insights into which types of models are more suitable for this task.

Related Work 2:

Paper/Project Title: "An Ensemble Approach for Term Deposit Subscription Prediction."

Problem and Method: The paper proposes an ensemble approach where multiple machine learning models, including decision trees, support vector machines, and logistic regression, are combined to predict term deposit subscription. The ensemble aggregates the predictions from individual models to make the final decision.

Difference and Improvement:

In contrast to the related work that uses a fixed ensemble of models, your approach explores various individual models, including bagging classifiers and gradient boosting classifiers, along with the ensemble method. By doing so, you can determine which individual models perform better and potentially tailor the ensemble composition for improved predictive accuracy.

Related Work 3:

Paper/Project Title: "Predictive Modeling for Term Deposit Subscription using Preprocessed Data."

Problem and Method:

This project aims to predict term deposit subscription by applying extensive data preprocessing techniques, including one-hot encoding, handling missing values, and scaling features. The authors then use a logistic regression model to make predictions.

Difference and Improvement:

In comparison, your approach incorporates not only data preprocessing techniques but also feature selection, outlier removal, and balancing data. By using a wider range of preprocessing methods and different machine learning algorithms, your approach may provide more robust and accurate predictions.

Why My Problem and Method are Better:

Diversity of Models: By using multiple machine learning algorithms, I can identify the most suitable model for this specific problem, leading to improved predictive performance.
Comprehensive Preprocessing: My method includes outlier removal, feature selection, and balancing data, which can enhance the quality of the input data and consequently, the predictive accuracy of the models.

Comparative Analysis: My evaluation section presents a comparison of different models' performance, allowing a better understanding of their strengths and weaknesses.

Ensemble and Skewed Model: By employing ensemble methods and a specific model for skewed data, I address the challenge of imbalanced class distributions, leading to better predictions in such cases.

Overall, your approach demonstrates a comprehensive analysis of various machine learning models, preprocessing techniques, and evaluation metrics, making it a valuable contribution to the field of term deposit subscription prediction.

5. Future Work

While the current models have provided some promising results, there are still some shortcomings that can be addressed in future work. Below are the major shortcomings along with proposed additions and enhancements to overcome them:

Time-Dependent Data: The dataset might contain temporal aspects that can influence the outcome, such as seasonality or trends in customer behavior. Ignoring these temporal dependencies might limit the model's predictive power.
Enhancement: Incorporate time-series analysis techniques to capture the temporal patterns in the data. This could involve using lagged features, time-based aggregations, or employing more specialized models like LSTM (Long Short-Term Memory) or GRU (Gated Recurrent Unit) for sequence prediction.

Interpretability: Some of the models used, like ensemble methods, may lack interpretability, making it challenging to understand the reasoning behind their predictions.
Employ interpretable models like decision trees or logistic regression, which provide clear feature importance rankings and decision rules. Additionally, consider using model explanation techniques like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) to interpret complex models' predictions.

External Data: Depending solely on the provided dataset might limit the model's performance. Additional external data sources, such as economic indicators, social media sentiment, or customer behavior data, could offer valuable insights.
Enhancement: Integrate relevant external data sources to enrich the existing dataset and improve model predictions. However, ensure that the external data is from reliable and relevant sources.

Feature Engineering:

The current model uses basic pre-processing techniques, including label encoding and outlier removal. However, there might be valuable insights and patterns in the data that are not effectively captured by the existing features.

Enhancement: Conduct in-depth feature engineering to derive new informative features or transform existing ones to better represent the underlying patterns in the data. This could involve domain-specific knowledge, interaction terms, or using dimensionality reduction techniques like PCA (Principal Component Analysis) or t-SNE (t-distributed Stochastic Neighbor Embedding).

Model Selection:

The current evaluation includes a range of models, but there might be more advanced algorithms or ensemble techniques that could perform even better on this particular problem.

Enhancement: Explore a wider range of machine learning models and algorithms, such as XGBoost, LightGBM, CatBoost, and deep learning models like neural networks. Additionally, consider using model optimization techniques like Bayesian Optimization or grid search to find the best hyperparameters for each model.

By addressing these shortcomings and implementing the proposed enhancements, the accuracy and reliability of the term deposit subscription prediction model can be significantly improved. Moreover, continuous exploration and experimentation with the latest advancements in machine learning and data science will be crucial in achieving state-of-the-art results in this domain.

6. Conclusion

In this study, we addressed the challenges faced by banking telecallers when blindly reaching out to potential customers to sell term deposit plans. The goal was to predict whether a customer would subscribe to the term deposit plan or not using machine learning algorithms. We utilized historical data containing various customer attributes and their subscription outcomes to build and evaluate several machine learning models.

After preprocessing the data by label encoding, outlier removal, balancing, and feature selection, we experimented with several classifiers, including Logistic Regression, BaggingClassifier, Support Vector Classifier (SVC), RandomForest, GradientBoostingClassifier, GaussianNB, and Decision Tree Classifier. Additionally, we implemented skewed versions of Bagging Classifier and Gradient Boosting Classifier.

Among the models tested, the RandomForest and Bagging Classifier (skewed) showed the highest accuracy at 70.67% and 73.85%, respectively. While the SVC model had the lowest accuracy, it was still valuable in understanding the model's limitations.

In conclusion, the results indicate that the RandomForest and Bagging Classifier (skewed) are the most suitable models for predicting term deposit subscription. However, there is still room for improvement, and future work could focus on fine-tuning hyperparameters and exploring more advanced ensemble methods.

The findings from this study can significantly benefit banking telecallers by allowing them to focus their efforts on potential customers who are more likely to subscribe to the term deposit plan. This targeted approach will not only enhance customer satisfaction but also improve the overall efficiency of the telemarketing process.

Bibliography

[1] Sebastian Raschka, & Vahid Mirjalili. (2019). Python Machine Learning. Packt Publishing Ltd.

[2] Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

[3] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[4] Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139.

[5] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer Series in Statistics.

[6] Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques. Elsevier.

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.