Banking Dataset: Customer Analysis

Elif Rana Tekin
Dept. of Artificial Intelligence Engineering
TOBB Economy and Technology University
Ankara, Turkey
elifranatekin2002@gmail.com

***Abstract*—This docume**

This project aimed to analyze data from a bank's direct marketing campaign and uncover relational contexts within the data. The primary focus was on the analysis and preprocessing of the data to reveal significant patterns and relationships. Following this, machine learning models were developed to predict whether customers would subscribe to a term deposit account. The dataset included various customer demographic information and campaign details. To enhance the robustness of the models, ensemble methods such as Boosting and Stacking were employed. The performance of the models was evaluated using various metrics (accuracy, recall, precision, F1 score), and the best-performing model was selected. The anticipated gains from the project included developing data analysis skills and gaining experience in discovering relational contexts through the active application of data mining techniques.

***Index Terms*—data mining, encoding, preprocess, exploratory data analysis**

I. Introduction

In the ever-evolving financial industry, the ability to accurately predict customer behavior is paramount for developing effective marketing strategies. This project focused on utilizing data from a bank's direct marketing campaign to analyze and predict customer responses, specifically whether they would subscribe to a term deposit account. The dataset, rich in customer demographics and campaign details, offered a strong foundation for uncovering meaningful patterns through data analysis and mining techniques.

The primary goal of this project was to conduct a thorough analysis and preprocessing of the marketing campaign data to reveal significant patterns and relational contexts. This involved examining various aspects of the data to understand the underlying factors influencing customer decisions. Following the data analysis, machine learning models were developed to predict whether customers would subscribe to a term deposit account. To enhance the robustness and accuracy of these models, ensemble methods such as Boosting and Stacking were employed.

The performance of the predictive models was evaluated using key metrics such as accuracy, recall, precision, and F1 score to ensure the selection of the most effective model. Engaging in this project offered a valuable opportunity to enhance data analysis skills and gain practical experience in applying data mining techniques. By the end of the project, a predictive model was developed, providing a deeper understanding of the relational contexts within the marketing campaign data, thereby contributing to the optimization of future marketing strategies in the banking sector.

II. Dataset

The dataset contains information about direct marketing campaigns of a Portuguese bank. The data source is UCI Machine Learning Repository. The name of the dataset is Bank Marketing and the link is **https://archive.ics.uci.edu/dataset/222/bank+marketing**. Data features can be listed as follows:

- It is a data set that is specialized in the business field.
- The data set has a multivariate structure.
- It contains 16 features and 45211 instances.
- Attribute values are Binary, Integer, Categorical and Date types.
- Target (y, dependent variable) takes Binary values.

A. Attributes Found in the Data Set and Their Descriptions

The dataset includes a variety of attributes, each with specific meanings and data types:

1) Binary Attributes:

- **default**: Indicates whether the customer has unpaid debt ('yes', 'no').
- **housing**: Indicates whether the customer has a housing loan ('yes', 'no').
- **loan**: Indicates whether the customer has a personal loan ('yes', 'no').

2) Integer Attributes:

- **age**: The age of the customer.
- **balance**: The annual average balance of the customer.
- **duration**: The duration of the last contact in seconds.
- **campaign**: The number of contacts performed during this campaign for the customer.
- **pdays**: The number of days since the customer was last contacted from a previous campaign (set to -1 if the customer was not previously contacted).
- **previous**: The number of contacts performed before this campaign.

3) Categorical Attributes:

- **job**: The occupation of the customer ('admin', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown').

	age	job	marital	education	default	balance	housing	loan	contact	day_of_week	month	duration	campaign	pdays	previous	poutcome
0	58	management	married	tertiary	no	2143	yes	no	NaN	5	may	261	1	-1	0	NaN
1	44	technician	single	secondary	no	29	yes	no	NaN	5	may	151	1	-1	0	NaN
2	33	entrepreneur	married	secondary	no	2	yes	yes	NaN	5	may	76	1	-1	0	NaN
3	47	blue-collar	married	NaN	no	1506	yes	no	NaN	5	may	92	1	-1	0	NaN
4	33	NaN	single	NaN	no	1	no	no	NaN	5	may	198	1	-1	0	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
45206	51	technician	married	tertiary	no	825	no	no	cellular	17	nov	977	3	-1	0	NaN
45207	71	retired	divorced	primary	no	1729	no	no	cellular	17	nov	456	2	-1	0	NaN
45208	72	retired	married	secondary	no	5715	no	no	cellular	17	nov	1127	5	184	3	success
45209	57	blue-collar	married	secondary	no	668	no	no	telephone	17	nov	508	4	-1	0	NaN
45210	37	entrepreneur	married	secondary	no	2971	no	no	cellular	17	nov	361	2	188	11	other

45211 rows × 16 columns

Fig. 1. Part of the dataset

- **marital**: The marital status of the customer ('divorced', 'married', 'single', 'unknown').
- **education**: The education level of the customer ('primary', 'secondary', 'tertiary', 'unknown').
- **contact**: The communication type used for contacting the customer ('cellular', 'telephone').
- **poutcome**: The outcome of the previous marketing campaign ('failure', 'nonexistent', 'success').

4) Date-Type Attributes:

- **day_of_week**: The day of the week of the last contact.
- **month**: The month of the last contact ('jan', 'feb', 'mar', ..., 'nov', 'dec').

5) Target (Dependent Variable):

- **y**: Binary attribute indicating whether the customer subscribed to a term deposit ('yes', 'no'). This attribute is not considered in the data analysis phases.

Figure 7 is a screenshot of the uncleaned dataset. The table was created by removing the y (target) column.

III. Related Works

Grid search is the most widely used parameter optimization technique in machine learning. It consists of training the model with all the possible combinations of the different values that its key parameters can take to find the most efficient combination. This operation can be very time-consuming if the model is slow or if its parameters and their possible values are numerous. The study reported in mentioned in Towards Explainable Machine Learning for Bank Churn Prediction Using Data Balancing and Ensemble-Based Methods [1] applied grid search on Random Forest on an imbalanced data. The fact that the data set used was specialized in the field of finance and the model gave good results, along with similarities such as the imbalance of the data, led to the use of this method in my project.

The methodology proposed in A New Hybrid Credit Scoring Ensemble Model with Feature Enhancement and Soft Voting Weight Optimization [2] integrates multiple base classifiers, each trained on different subsets of the original dataset, to create a unified ensemble model. The study conducted extensive experiments on a credit scoring dataset with imbalanced classes, employing various ensemble methods such as bagging, boosting, and stacking. The findings reveal that the proposed ensemble approach outperforms traditional methods, achieving higher accuracy and F1 scores, thus demonstrating the effectiveness of ensemble techniques in handling complex and unbalanced data.

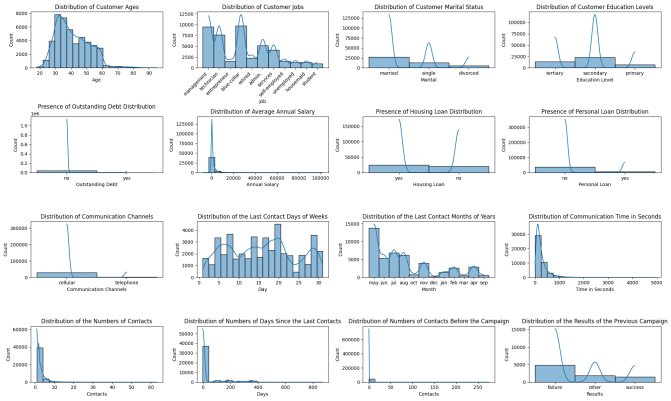

Fig. 2. Histogram Plots of Attributes

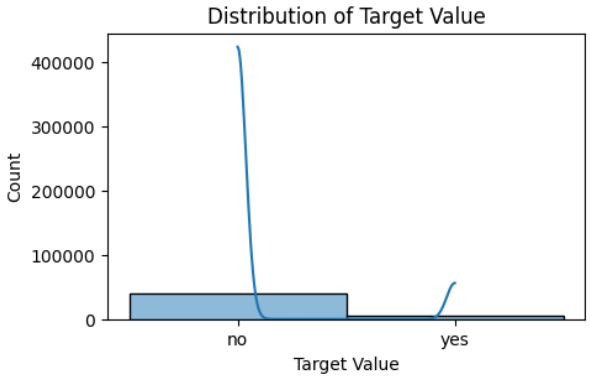

Fig. 3. Histogram Plot of the Target

IV. Exploratory Data Analysis (EDA)

This stage includes detailed examination of the data set. Statistical properties were calculated and visualized with the help of various graphics. General properties of the data set were examined by calculating statistical measures such as mean, median, variance and standard deviation. These measures helped to understand the general trends and distribution of the data set. Various graphics such as histograms, box plots and correlation matrix were created to better understand the distribution and relationships of the data.

A. Statistical Features

The general characteristics of the dataset, such as mean, median, variance, and standard deviation, were computed and analyzed. These measures helped in understanding the overall tendencies and distributions within the dataset. Various visualizations, including histograms, box plots, and correlation matrices, were created to better understand the distribution and relationships within the data. These visualizations helped in observing potential relationships and trends within the dataset and also detecting outliers.

B. Outlier Detection

Box plots (Figure 5 and scatter plots (Figure 6) were used to determine the outliers. After identifying outliers they are

	age	balance	day_of_week	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

Fig. 4. Descriptive Statistics of the Dataset

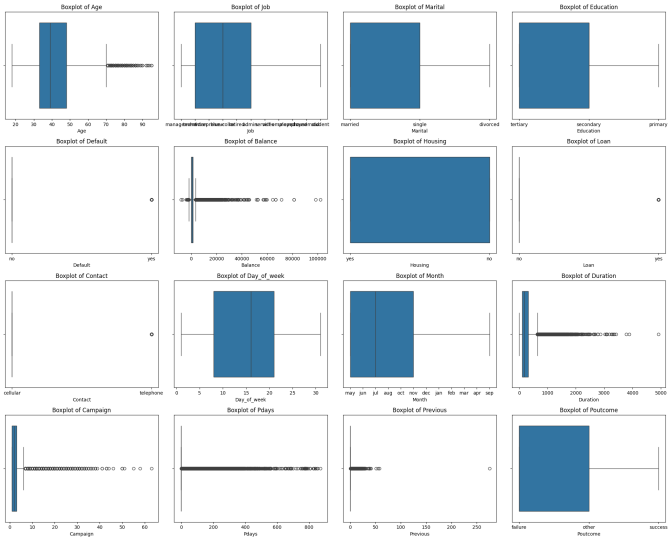

Fig. 5. Boxplots of Attributes

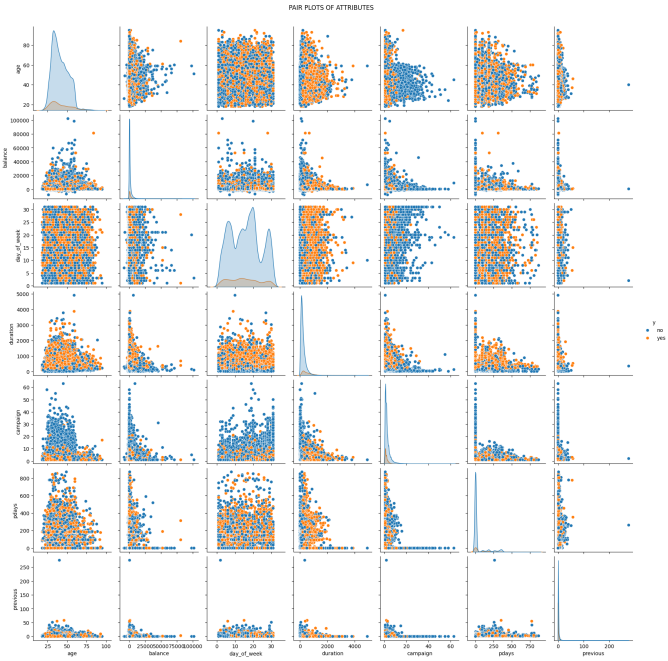

Fig. 6. Pair Plots of the Attributes

	Total Null Values
age	0
job	288
marital	0
education	1857
default	0
balance	0
housing	0
loan	0
contact	13020
day_of_week	0
month	0
duration	0
campaign	0
pdays	0
previous	0
poutcome	36959
y	0

Fig. 7. Total number of NULL values per attribute.

decided to remove or not from the dataset. The goal here was to prevent extreme values in the dataset from negatively affecting the analysis results. The 'balance' attribute, which indicates the annual average balance of customers, was the most prominent in terms of outliers when using box plots.

Scatter plots along with histograms (Figure 2) were used as distribution indicators to ensure that the values identified as outliers were indeed outliers. Initially, the values were examined based on the target element 'y'. Subsequently, checks were also made based on other numerical values (duration, age) to obtain a more heterogeneous view. Since no high correlation was found between attributes in the heatmap, these values were selected based on their numerical range rather than their relationship level.

C. Missing Data Analysis

There are NULL values in 4 of the 16 attributes ('job', 'education', 'contact', 'poutcome'). There is no NULL value in the target. In Figure 7, total number of NULL values per attribute is demonstrated. How to handle these values (intuitive filling or dropping) is given in the Data Preprocessing stage.

D. Principal Component Analysis (PCA)

Principal Component Analysis (PCA) was performed to reduce the dimensionality of the dataset and to visualize the data in 2D and 3D spaces. This technique helps in identifying the directions (principal components) that capture the maximum variance in the data, thereby simplifying the complexity and aiding in better visualization and interpretation. 3D and 2D visualizations are shown in the Figure 8 and Figure 9,respectively.

The 'yes' and 'no' values in the target variable are 5289 and 39922 respectively, indicating that the data is highly unbalanced on the output basis. This imbalance indicates that the data is skewed towards the negative class, which could affect the performance of machine learning models

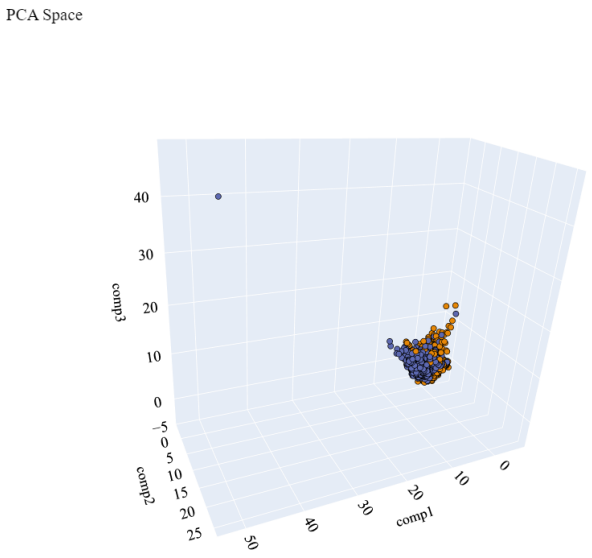

Fig. 8. 3D visualization of the dataset using PCA

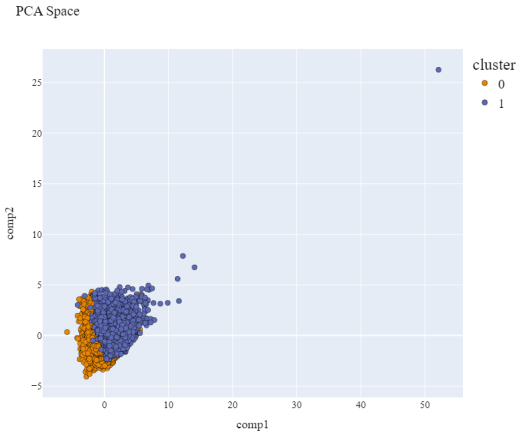

Fig. 9. 2D visualization of the dataset using PCA

and necessitates careful handling during model training and evaluation.

V. Methodology

A. Data Preprocessing

The data preprocessing phase included handling missing data, converting categorical data to numerical data, and scaling the data.

1) Missing Data Handling: The data cleaning process involved filling or removing NULL values as needed. Upon examining the dataset, it was found that the 'education' column had 1857 NULL values. These values were filled based on the types of jobs in the 'job' column. The values selected to replace the NULL value were decided according to the

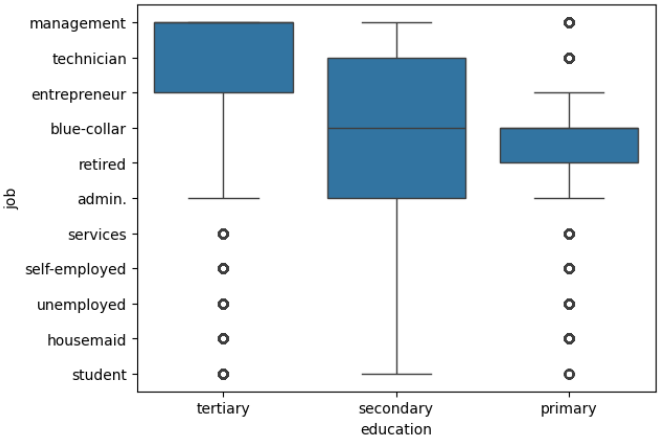

Fig. 10. education - job Distribution

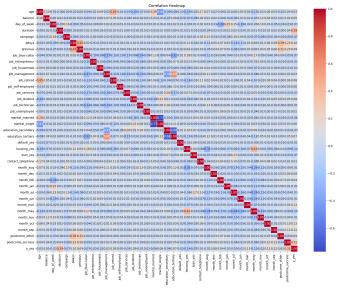

Fig. 11. Correlation Map

distribution of 'job' values and 'education' values (Figure 10). The majority of the education level completed by occupation type and the ratio of the majority to the remaining minority are the points to be considered in this regard.

Filling operation was applied only for 'education' column. No intuitive filling relation was found for the remaining columns which are 'job', 'contact', 'poutcome' as it might be estimated from Correlation Map in Figure 11. Due to the high number of NULL values (12- 13) and the difficulty of intuitively filling these columns based on their definitions, it was decided to drop these columns. Rows (instances) with 'job' and 'education' values that could not be filled by the process mentioned above were removed. Since the number of NULL values in these columns was small compared to the total number of data, it was not deemed harmful to discard these values (Figure 14 - 15).

As a result of filling missing values and dropping necessary columns, the dataset was reduced to 44,895 rows and, excluding the 'y' column, 15 columns.

2) Conversion of Categorical and Date-Type Variables: Categorical data needs to be converted to numerical data to be used in machine learning algorithms. For this project, the label encoding technique from the scikit-learn library was employed for most of the categorical variables. Label encoding assigns a unique integer to each category within a variable, converting the categorical data into a numerical format that

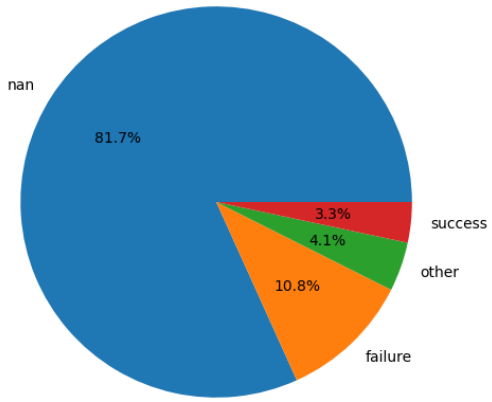

Fig. 12. Ratio of NULL Nalues in 'poutcome' Column

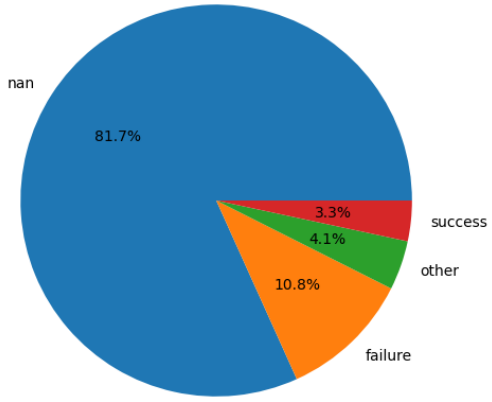

Fig. 13. Ratio of NULL Nalues in 'pdays' Column

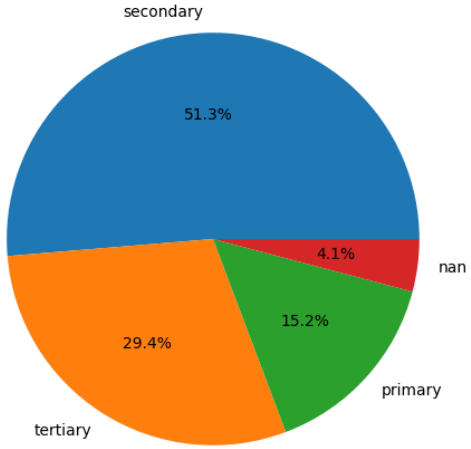

Fig. 14. Null Value Control After Filling in the 'education' Column According to the 'education'-'job' Relationship

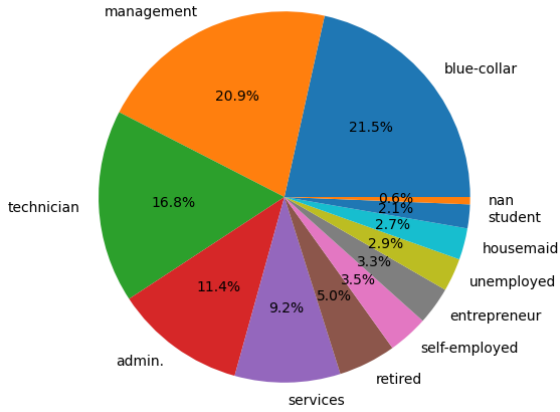

Fig. 15. Ratio of NULL Nalues in 'job' Column

machine learning algorithms can understand. This method is particularly useful when dealing with ordinal categorical variables where the order of the categories is important.

Additionally, for the 'month' column, which is another date-type attribute that takes string values as abbreviations of month names and represents the month when the last contact with the customer occurred, a specific type of encoding known as ordinal encoding was applied. Ordinal encoding is especially suitable for this variable because the months have a natural order from January to December. By applying ordinal encoding, each month is assigned a unique integer that reflects its position in the calendar year, thereby maintaining the sequential nature of the data.

Although the 'day_of_week' attribute is categorized as a date-type attribute, it takes integer values ranging from 1 to 31. Given this integer-based format, the 'day_of_week' attribute did not require any additional encoding.

3) Scaling: The Robust Scaling technique was used for scaling the dataset. This method was chosen for its effectiveness in reducing the impact of outliers. It scales the data using the median and interquartile range (IQR), aiming to prevent outliers from negatively affecting the model's performance.

B. Model Generation, Improvement and Evaluation

This section details the process of model selection, improvement, and evaluation for predicting whether customers will subscribe to a term deposit account. The initial models used were RandomForestClassifier and LogisticRegression. Subsequent improvements were made using hyperparameter tuning with Grid Search and Cross-Validation and Ensemble Methods (Boosting and Stacking).

1) Initial Models & Performance: RandomForestClassifier and LogisticRegression were the models trained and tested at this stage.

	precision	recall	f1-score	support
0.0	0.918912	0.970743	0.944117	7588.000000
1.0	0.643660	0.381541	0.479092	1051.000000
accuracy	0.899062	0.899062	0.899062	0.899062
macro avg	0.781286	0.676142	0.711604	8639.000000
weighted avg	0.885426	0.899062	0.887543	8639.000000

Fig. 16. Decision Tree Classification Report

	precision	recall	f1-score	support
0.0	0.899385	0.982472	0.939094	7588.000000
1.0	0.620000	0.206470	0.309779	1051.000000
accuracy	0.888066	0.888066	0.888066	0.888066
macro avg	0.759692	0.594471	0.624437	8639.000000
weighted avg	0.865395	0.888066	0.862533	8639.000000

Fig. 17. Logistic Regression Classification Report

- The **RandomForestClassifier** was initially used to understand the baseline performance. The confusion matrix and classification report for this model are shown below. The model performed well for the majority class ('no'), but it struggled with the minority class ('yes'), highlighting the class imbalance issue.
- The **LogisticRegression** was also evaluated to provide a comparison with the Decision Tree model. This model similarly showed high performance for class '0' but lower performance for class '1'.

The Decision Tree model was initially used to understand the baseline performance. The confusion matrix and classification report for this model are shown below. The model performed well for the majority class ('no'), but it struggled with the minority class ('yes'), highlighting the class imbalance issue.

C. Hyperparameter Tuning with Grid Search & Cross-Validation

To improve model performance, Grid Search and Cross-Validation were applied for hyperparameter tuning. Grid Search systematically works through multiple combinations of parameter tunes, cross-validating as it goes to determine which tune gives the best performance.

- The Random Forest model was tuned using Grid Search, optimizing parameters such as the number of trees, depth of the trees, number of features to consider at every split, minimum samples required to split a node. The tuned model showed improved performance, especially for class '1'.
- Similarly, the Logistic Regression model was optimized using Grid Search to find the best regularization parameter. The results showed marginal improvements, particularly in precision and recall for class '1'.

	precision	recall	f1-score	support
0.0	0.919725	0.970875	0.944608	7588.000000
1.0	0.648649	0.388202	0.485714	1051.000000
accuracy	0.899988	0.899988	0.899988	0.899988
macro avg	0.784187	0.679538	0.715161	8639.000000
weighted avg	0.886747	0.899988	0.888780	8639.000000

Fig. 18. Classification Report of Random Forest Using Grid Search Parameters

	precision	recall	f1-score	support
0.0	0.899192	0.982736	0.939110	7588.000000
1.0	0.621387	0.204567	0.307802	1051.000000
accuracy	0.888066	0.888066	0.888066	0.888066
macro avg	0.760290	0.593651	0.623456	8639.000000
weighted avg	0.865395	0.888066	0.862306	8639.000000

Fig. 19. Classification Report of Logistic Regression Using Grid Search Parameters

D. Model Performance with Ensemble Methods

To further enhance model performance, ensemble methods such as Boosting and Stacking were employed. Ensemble methods combine the predictions of multiple models to improve accuracy and robustness. The models used for ensemble approaches in this project are Random Forest and Logistic Regression models, which were selected as "best" with grid search in the previous part.

- AdaBoost, or Adaptive Boosting, was applied using Random Forest as the base estimator. AdaBoost works by iteratively adding models that correct the errors of the combined ensemble. This method significantly improved the performance for class '1', demonstrating the strength of ensemble methods in handling imbalanced datasets.
- Stacking is an ensemble learning technique that combines multiple classification models via a meta-classifier. The base-level models are trained on the entire dataset, and the meta-model is trained on the outputs of the base-level models. For stacking applications in this project, Random Forest and Logistic Regression models were used as final estimators.

The models after hyperparameter tuning or ensemble method application showed better precision, recall, and F1-

	precision	recall	f1-score	support
0.0	0.919725	0.970875	0.944608	7588.000000
1.0	0.648649	0.388202	0.485714	1051.000000
accuracy	0.899988	0.899988	0.899988	0.899988
macro avg	0.784187	0.679538	0.715161	8639.000000
weighted avg	0.886747	0.899988	0.888780	8639.000000

Fig. 20. AdaBoost Using Random Forest Estimator Classification Report

	precision	recall	f1-score	support
0.0	0.905955	0.910253	0.908099	7588.000000
1.0	0.329064	0.317793	0.323330	1051.000000
accuracy	0.838176	0.838176	0.838176	0.838176
macro avg	0.617509	0.614023	0.615714	8639.000000
weighted avg	0.835772	0.838176	0.836957	8639.000000

Fig. 21. Stacking Using Random Forest as the Final Estimator Classification Report

	precision	recall	f1-score	support
0.0	0.900000	0.982077	0.939249	7588.000000
1.0	0.621170	0.212179	0.316312	1051.000000
accuracy	0.888413	0.888413	0.888413	0.888413
macro avg	0.760585	0.597128	0.627780	8639.000000
weighted avg	0.866078	0.888413	0.863464	8639.000000

Fig. 22. Stacking Using Logistic Regression as the Final Estimator Classification Report

scores, providing more reliable predictions for the term deposit subscription. By employing these advanced techniques, the performance of the models improved significantly, particularly in handling the class imbalance in the dataset. The overall performances are demonstrated in Figure 23

Metric	Random Forest	Logistic Regression	Random Forest Using Grid Search Parameters	Logistic Regression Grid Search Parameters	AdaBoost Using Random Forest Estimator	Stacking Using Random Forest as the Final Estimator	Stacking Using Logistic Regression as the Final Estimator
Accuracy	0.899062	0.888066	0.899368	0.888066	0.895358	0.838176	0.888413
Recall	0.676142	0.594471	0.679538	0.593651	0.626492	0.614023	0.597128
Precision	0.781286	0.759692	0.784187	0.760290	0.790263	0.617509	0.760585
F1 Score	0.711604	0.624437	0.715161	0.623456	0.664751	0.615714	0.627780

Fig. 23. Accuracy, Recall, Precision, F1 Score Metrics for All Models Evaluated

The models, after hyperparameter tuning and the application of ensemble methods, demonstrated varying degrees of improvement in accuracy. Hyperparameter tuning consistently enhanced model performance, while ensemble method Boosting maintained robustness.

The ROC curve provides a graphical representation of a model's performance across all classification thresholds. The curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings, offering insights into the trade-off between sensitivity (recall) and specificity for the models used in this project. The Random Forest models, both before and after hyperparameter tuning via Grid Search, demonstrate strong performance, with ROC curves that are closest to the top-left corner of the plot. This indicates a high TPR with a low FPR, which is desirable for a classification model. The tuned Random Forest model (green line) slightly outperforms the initial model (blue line), showing the benefit of hyperparameter optimization.

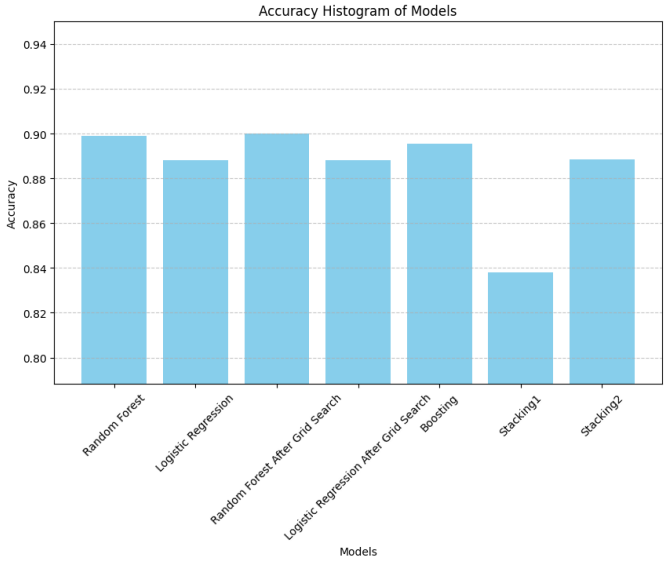

Fig. 24. Accuracy Histogram of Models

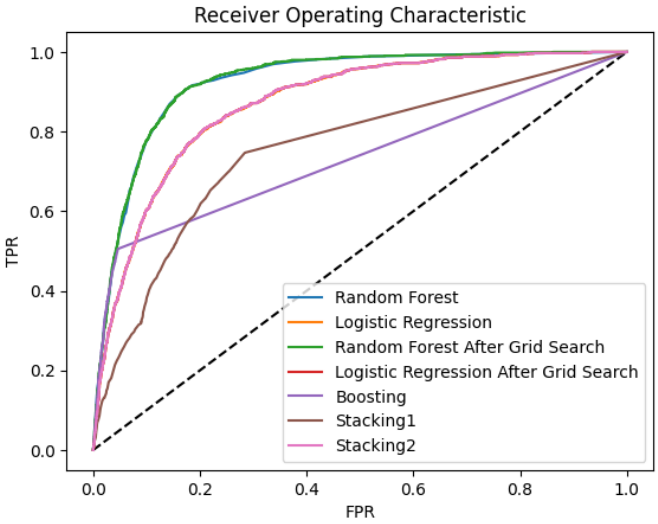

Fig. 25. Receiver Operating Characteristics of Models

REFERENCES

[1] Koumetio Tekouabou, Cédric Stéphane & Gherghina, Ştefan Cristian & Toulni, Hamza & Mata, Pedro & Martins, José. (2022). Towards Explainable Machine Learning for Bank Churn Prediction Using Data Balancing and Ensemble-Based Methods. Mathematics. 10. 2379. 10.3390/math10142379.

[2] R. D. Ranpara and P. S. Patel, "An Ensemble Learning Approach to Improve Credit Scoring Accuracy for Imbalanced Data," 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 2023, pp. 1-5, doi: 10.1109/ICIICS59993.2023.10420986.