Advanced Bank Term Subscription Classification Techniques

Yiwei Han
University of Rochester
Rochester
yhan32@u.rochester.edu

Will Yang
University of Rochester
Rochester
hyang58@u.rochester.edu

ABSTRACT

Term deposit as a conventional mean to provide fluidity for financial institutions, finance operators often experience demands to make prediction of potential scale of future deposit in able to layout investment plan in advance. The following analysis regarding term deposit subscription prediction based on customer profile from a Portuguese bank. The analysis utilizes 16 features and multiple data mining processes classification techniques to classify and predict term subscription. This project is conducted as the final project of CSC240 Data Mining at the University of Rochester in Spring 2023.

KEYWORDS

Data Preprocessing, Classification, Term Deposit, Tensorflow, Scikit Learn

ACM Reference Format:
Yiwei Han and Will Yang. 2023. Advanced Bank Term Subscription Classification Techniques. In *Proceedings of ACM Conference.* ACM, New York, NY, USA, 8 pages. https://doi.org/0000001.0000001

1 INTRODUCTION

Classifying and predicting bank deposits from user data is essential in finance. A bank deposit is a form of investment in which a certain amount of money is deposited for a certain period of time at a fixed interest rate. Classifying term deposits as term deposits or not is important for banks to manage liquidity and make informed investment decisions. However, predicting whether a customer will apply for a term deposit is a complex task due to various factors such as customer demographics, economic indicators, and customer behavior. The Bank Full dataset is a rich collection of his over 45,000 observations from Kaggle, providing a wealth of information about client demographics, financial status, and past interactions with marketing his campaigns. With or without subscription, there are 17 functions including target variables. [10]

The goal of this project is to create a predictive model that can accurately classify whether a customer will apply for a fixed deposit with a bank. A data set containing information about customer demographics, economic figures, customer behavior, etc. was provided as a training set. The goal is to use this training data to build and train a model that best predicts whether a customer will request a fixed deposit from the bank given the features provided in the test set. Model performance is measured using various metrics such as precision, accuracy, recall, F1 score, and ROC curve. Approaches used in this project include exploratory data analysis to understand data structure, variable distributions, and feature correlations. Perform data preprocessing to handle missing data, encode and normalize categorical, binary, and ordinal features, and remove outliers from numerical features. Finally, various classification techniques are considered and the best model is selected based on the evaluation metrics. This report explains each step of the process and examines the final result. Our model achieved the highest accuracy among Kaggle participants.

Authors addresses: Yiwei Han, University of Rochester, RochesterNEW YORK; Will Yang, University of Rochester, RochesterNEW YORK.

ACM Conference, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/0000001.0000001

2 EXPLORATORY DATA ANALYSIS

Exploratory data analysis (EDA) is a key step in the data analysis process aimed at uncovering patterns, trends, and insights in data. This involves examining and summarizing the main characteristics of the dataset, such as its structure, characteristics, and relationships between variables. The EDA phase is often the first step in a data analysis project, providing an initial understanding of the data and informing subsequent analysis and modeling decisions. This report presents her EDA results for a given dataset, focusing on her two main components: data exploration and distribution of all attributes.

2.1 Data Exploration

2.1.1 Data Types. Initial investigations were based on the assumption that the binary classification model serves as a method for predicting the binomial outcome, which is the goal of this study. Binary classification models typically require all features to be encoded as numeric features. Therefore, we made a preliminary assessment of the types of data contained in the dataset. As shown in Figure 1, during analysis of the dataset, we observed that over 50% of the features were in non-numeric form, requiring coding for modeling. As a result, we implemented a data preprocessing step to transform categorical features into numerical features, which were used to train a binary classification model.

2.1.2 Missing Values. The quality of classification model results is highly dependent on the quality of the training data. There were no missing values in the dataset of this study. However, a significant number of categories contained 'unknown' data, which had to be handled appropriately to ensure the quality of the training data. This required careful handling during the data cleansing process.

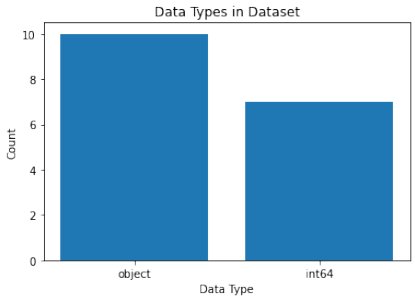

Figure 1: Features by Data Type

2.2 Numeric Attributes

Histograms is a commonly used visualization method to show the distribution of numerical features in a dataset, including patterns and outliers that may be present. By using histograms to visualize the distribution of numerical characteristics, you can gain insight into your data and better understand the underlying characteristics of the characteristics. In the image below, there are seven numeric attributes, most of which are skewed to the right by outliers.

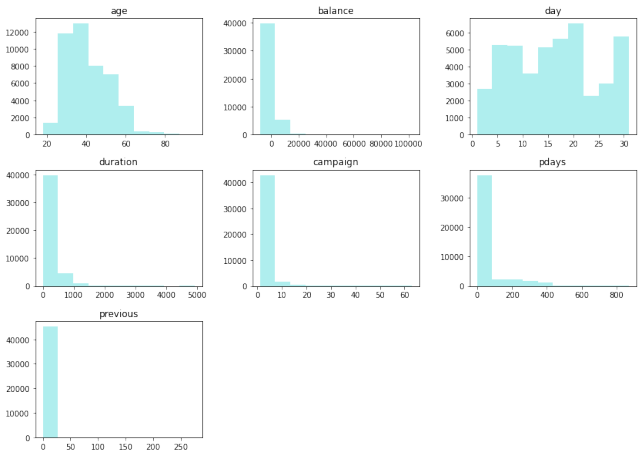

Figure 2: Distribution of Numerical Attributes

2.3 Categorical Attributes

Bar charts are often used to visualize the distribution of categorical features within a data set. Bar charts allow you to quickly and easily compare the distribution of each category, helping you identify imbalances and patterns in your data. By visualizing the distribution of categorical features using bar charts, you can gain insight into the features that underlie them and better understand their relationships with other features in your dataset. In the image below, the dataset has 5 categorical attributes and all categorical attributes within each category are not evenly distributed.

2.4 Ordinal Attributes

Bar chart is a commonly used visualization method to show the distribution of ordinal features in a data set. Bar charts allow you

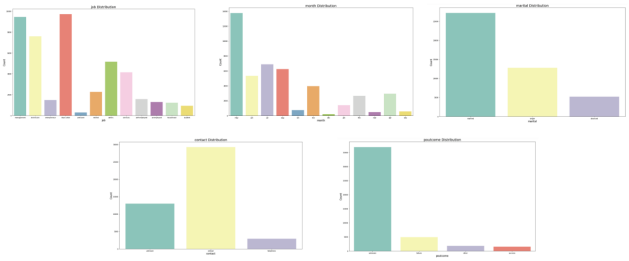

Figure 3: Distribution of Categorical Attributes

to quickly and easily compare the distribution of each category and help you identify patterns or imbalances that exist in your data. There is only one ordinal attribute Education in the dataset. From the image below, you can see that the most common category is secondary, followed by tertiary and primary.

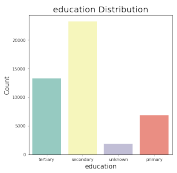

Figure 4: Distribution of Ordinal Attribute Education

2.5 Binary Attributes

Pie charts are often used to visualize binary features in datasets. The diagram is divided into two sections, each section representing one possible value of the binary trait. The size of each bin is proportional to the number of occurrences of that value in the dataset. By visualizing binary traits using pie charts, you can quickly and easily see the relative frequency of each value and gain insight into the overall distribution of traits. A dataset has three binary characteristics. Loans and defaults are unevenly distributed, with far more 'no's than 'yes', and housing is more evenly distributed, each accounting for 50% of the dataset.

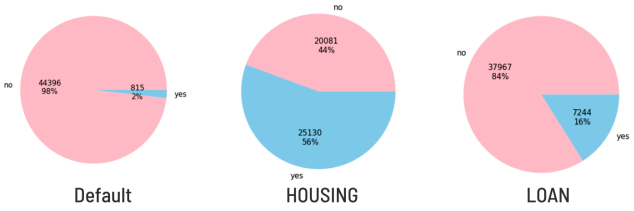

Figure 5: Distribution of Binary Attributes

2.6 Target

Target variable is also a binary attribute. The method to view the distribution of the target variable is also using pie chart. From the figure below, the majority of the target variable is "no", indicating

83% of customers do not have term deposit. This shows the dataset is very imbalanced and has the problem of oversampling, thus we need to further modify it during the data preprocessing step to reduce oversampling.

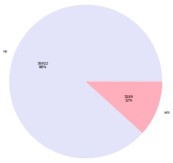

Figure 6: Distribution of the Target Variable

3 DATA PREPROCESSING

Open source libraries such as pandas and numpy provide powerful tools to make data analysis and preprocessing easier. In addition, machine learning libraries such as scikit-learn offer a wide range of algorithms that can be applied to datasets to solve complex problems. The key to achieving high performance is often to improve the quality of the data through effective cleaning, preprocessing, and feature engineering. To improve the performance of machine learning models, it is often necessary to preprocess the data to make it compatible with the assumptions made by the models. The preprocessing includes outlier analysis, normalization, feature selection, and encoding. Overall, the quality of the data is crucial in determining the accuracy and reliability of the models, and effective data preprocessing and feature engineering can make a significant difference in the performance of the models.

3.1 Outlier Analysis

Outliers interrupt the data mining processing. Removing outliers is a crucial step in data preprocessing as it can have a significant impact on the results of the analysis. Outliers represents extreme values that are not representative of the data. These extreme values can distort the data, making it difficult to see patterns or relationships, leading to erroneous or biased conclusions. The metric that utilizes to remove outliers are to remove those tuples who has numerical features that are lower than 1.5*quartile1 or higher than 1.5 *quartile3.

3.2 Dropping Uncorrelated Data

The categorical attributes "Job" and "Education" has unknown data, but not much. By dropping tuples with unknown data of these columns, the information level of the dataset is improved and the accuracy would be leveled up.

More than 80% of the attribute "Poutcome" has "unknown" entries, which means this attribute does not provide much useful information. As a result, dropping this column is necessary for the data preprocessing.

3.3 Encoding

3.3.1 Categorical Attributes. When working with categorical attributes, it is often useful to use one-hot encoding to create dummy variables. One-hot encoding is a process of converting categorical data into a binary format, where each unique category is assigned a binary value. The process of creating dummy variables can be easily accomplished using the get dummies class in scikit-learn. This creates a binary column for each category in the original column, where a 1 represents the presence of the category and a 0 represents the absence of the category of a tuple.

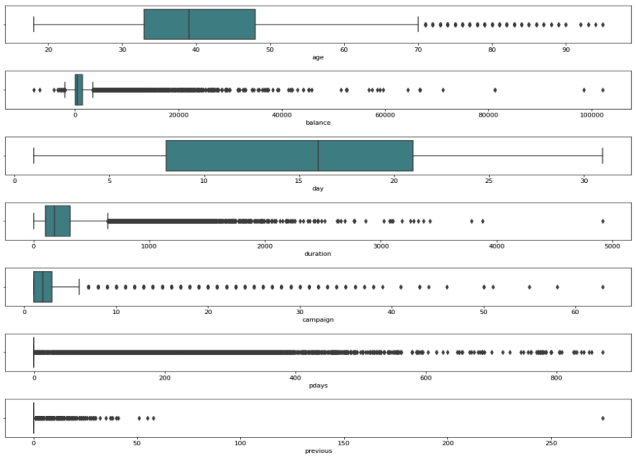

Figure 7: Box Plot of Numerical Attributes before Removing Outliers

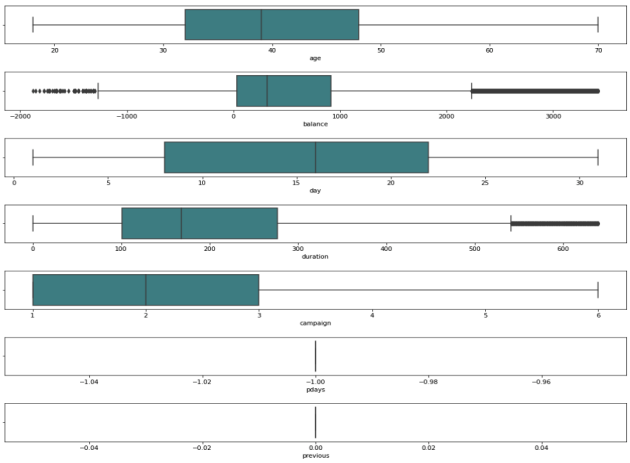

Figure 8: Box Plot of Numerical Attributes after Removing Outliers

3.3.2 Binary Attributes. Label encoding is a technique to convert categorical variables into numerical variables. In the case of binary data, where the variables take on two possible values, yes and no, label encoding can be used to map these values to numerical labels, typically 0 and 1. [2] For example, in a dataset containing a column for loan with values "yes" and "no", label encoding can be used to convert "no" to 0 and "yes" to 1. LabelEncoder() in Scikit-learn is a popular Python library that can be used to accomplish this goal.

3.3.3 Ordinal Attributes. Ordinal data refers to categorical variables that have a natural ordering or ranking. For example, the education column in a dataset may have values such as primary, secondary, tertiary, and unknown. To encode this ordinal data, we need to give each category a label in the specific order number. In this case, we can assign the label 1 to primary, 2 to secondary, 3 to tertiary, and 0 to unknown. This ensures that the ordinal relationship between the categories is preserved, while still allowing for numerical computation.

3.4 Normalization

Analyzing the numerical data, it is observed that several features exhibited varying ranges of values. To prevent any particular feature from having a disproportionate impact on the model, normalization techniques is applied using the MinMaxScaler method from the scikit-learn package. [7] This method scales each feature to a range between 0 and 1, ensuring that each feature contributes equally to the model's predictions.

The MinMaxScaler method preserves the shape of the original distribution and is particularly useful for algorithms that use distance measures. By normalizing the data, we can improve the accuracy and efficiency of our models, as well as reduce the risk of overfitting on any particular feature.

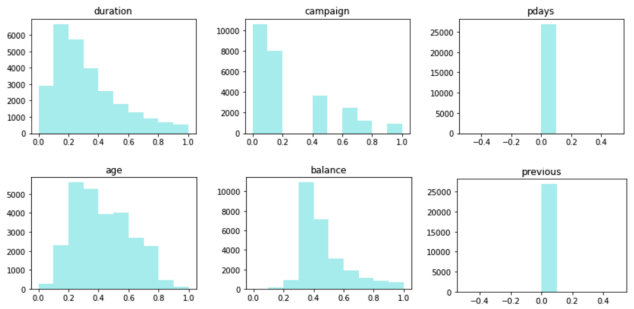

Figure 9: Distribution of Numerical Attributes after Normalization

4 MODELS AND ANALYSIS

Several machine learning models are explored for binary classification tasks, including logistic regression, decision trees, support vector machines (SVM), K-nearest neighbors (KNN), XG Boost, and deep learning models. The objective of this analysis is to compare the performance of these models on our dataset and identify the most effective model for the task at hand.

4.1 Model Preparation

To ensure the accuracy and generalizability of the model, two important processes were performed. First, the labeled data was split into a training set (70%) and a test set (30%) using a train-test split technique. This allowed for testing the model on data that was not used in the training process and evaluating its ability to generalize to new data.

Secondly, to address the issue of class imbalance in the target variable, a Synthetic Minority Over-sampling Technique (SMOTE) was employed. This technique oversamples the minority class (customers who subscribed to a term deposit) by creating synthetic examples of the minority class, thus balancing the class distribution and avoiding bias towards the majority class. [3]

By utilizing these techniques and metrics, we can confidently assess the effectiveness of the model in predicting customer subscription to a term deposit and make informed decisions to improve marketing campaigns and customer engagement.

4.2 Baseline Model: Logistic Regression

Logistic regression is a statistical method used for binary classification problems. The logistic regression model uses a linear combination of input features and applies a sigmoid function to the result to transform it into a probability value between 0 and 1. Mathematically, the model can be expressed as $p(y=1|x) = 1 / (1 + \exp(-z))$ where $p(y=1|x)$ is the probability of the input x belonging to class 1 z is a linear combination of the input features x and corresponding weights w, and a bias term b: $z = w.T\ x + b$. [6] During training, the weights and bias are learned by minimizing a loss function using an optimization algorithm that adjusts the weights and bias to better fit the training data. Once the model is trained, it can be used to predict the class of new inputs by thresholding the predicted probability at 0.5.

Our accuracy for logistic regression is 90.01%. From the confusion matrix, it indicates that the logistic regression model generates a lot more false positive cases than false negative cases. This is probably because the cost of making a false positive error may be lower than that of making a false negative error. Moreover, logistic regression assumes a linear relationship between the independent variables and the log odds of the dependent variable. If the relationship is non-linear, then logistic regression may not perform well and may produce more false positives.

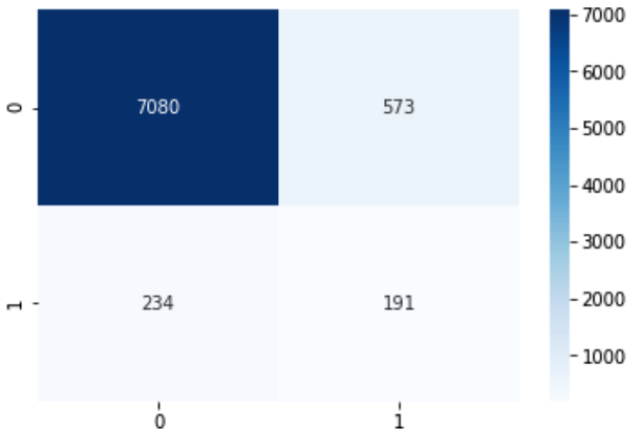

Figure 10: Confusion Matrix of Logistic Regression

4.3 Model Enhancement: Decision Tree

Decision Tree algorithm works by recursively splitting the data into two groups based on the values of the input features. At

each internal node, the algorithm selects the feature that best separates the data into the two classes, based on a metric such as Gini impurity or information gain. The splitting continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of samples required to split an internal node. It can be used to classify new instances by traversing the tree from the root to a leaf node based on the values of the input features. The class label associated with the leaf node is then assigned to the instance. [8]

The accuracy for the decision tree model is 90.81%, which is better than the baseline logistic regression model. This is because logistic regression assumes and performs better when the dataset has linear relationship. The distribution of the confusion matrix is similar to the one of logistic regression. Decision trees can model non-linear relationships between the input features and the target variable, which makes the performance better. However, using the 5-fold cross validation method, the decision tree model has overfitting problem, which the model performs well on some parts of the data but poorly on others. The same result is shown on theh ROC curve of the decision tree model.

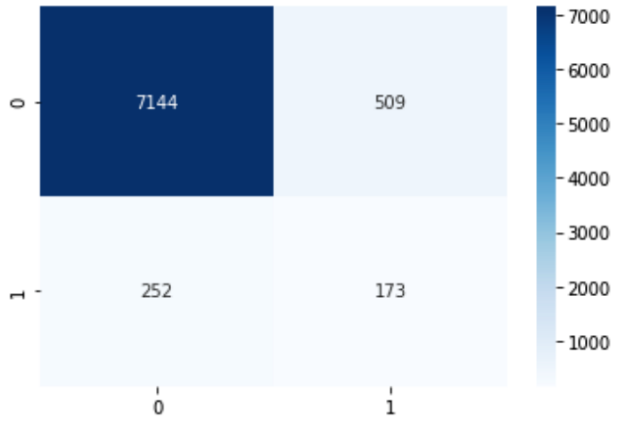

Figure 11: Confusion Matrix of Decision Tree

4.4 Model Enhancement: KNN

In KNN, the classification of a new data point is determined by the majority vote of its k nearest neighbors in the training set. To classify a new data point, the algorithm computes the distance between the new point and all the points in the training set. It then selects the k training points that are closest to the new point based on this distance measure. The algorithm then assigns the new point to the class that is most common among its k nearest neighbors. [4]

Our accuracy for KNN is not high, it's 89.6%. KNN has a lower accuracy because it is a non-parametric algorithm. As a result, it may assign a new data point to a class that is not the true class but happens to have more nearby training examples. This can lead to a higher rate of misclassification. Also, by looking at the confusion matrix, KNN produces way more false positive cases and true negative cases, and less true positive and false negative cases than logistic regression and decision tree.

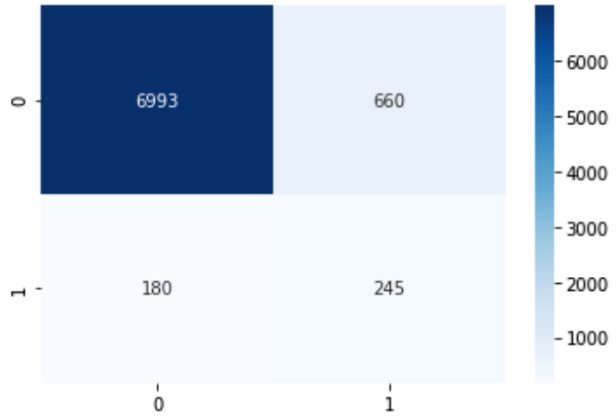

Figure 12: Confusion Matrix of KNN

4.5 Model Enhancement: SVM

The goal of SVM is to find a hyperplane that best separates the two classes in the input data. To do this, SVM considers a subset of the input data, called the support vectors, which are the data points closest to the decision boundary. SVM tries to find the hyperplane that maximizes the distance between the support vectors and the decision boundary. [5] In binary classification, SVM outputs a prediction of 1 or 0 depending on which side of the decision boundary a given input data point falls on.

The accuracy for the SVM is the lowest, which is 87.67%. This is because if the input data is not linearly separable, SVM may not be able to find a decision boundary that effectively separates the two classes. Moreover, looking at the confusion matrix of SVM, it is very similar to the one of KNN. They both produce way more false positive cases and true negative cases, and less true positive and false negative cases. This suggests that SVM and KNN are similar methods for the dataset, and both do not provide good results. Therefore, more enhanced methods should be tried.

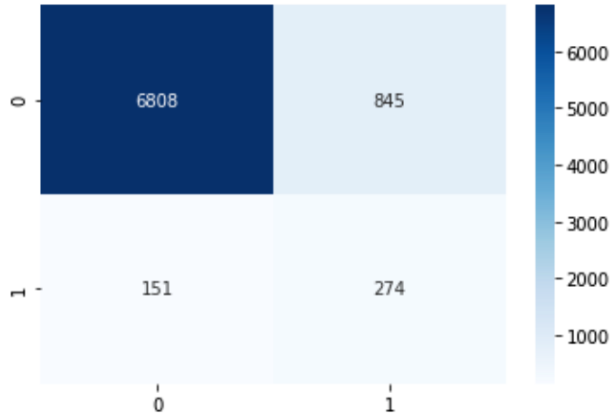

Figure 13: Confusion Matrix of SVM

4.6 Model Enhancement: XGBoost

The XGBoost algorithm works by iteratively adding decision trees to a model, with each tree being designed to correct the errors of the previous tree. The algorithm uses gradient descent to optimize the objective function. At each iteration, the algorithm evaluates the objective function on the training data and adjusts the weights of the instances that were misclassified, giving more weight to the instances that were misclassified by the previous trees. In addition to gradient boosting, XGBoost also includes regularization, which helps to prevent overfitting, and parallel processing, which allows it to handle large datasets efficiently. [1]

The XG Boost method achives the highest accuracy among all our models, which is 94.62%. By using the XG Boost method, it applies regularization method, which improves the overfitting problem of the decision tree method. Moreover, it also not assume a linear relationship of the dataset, which is better then the logistic regression method. From the confusion matrix, it is obvious that it significantly reduces the number of false positive cases than decision tree and logistic regression. Overall, XGBoost's combination of handling imbalanced data, ensemble learning, and regularization techniques makes it more effective at reducing false positive cases.

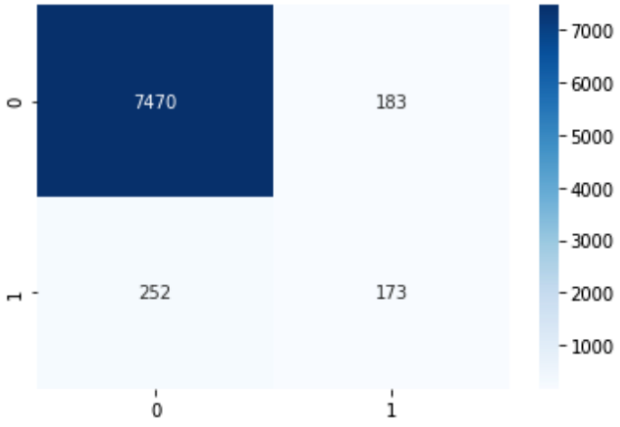

Figure 14: Confusion Matrix of XG BOOST

4.7 Model Enhancement: Tensorflow

Deep learning framework by tensorflow is another method applied other than the conventional machine learning models. We created a neural network model using the Keras two dense layers, one with 64 neurons and another with 1 neuron. The activation function is set to be ReLU for the first layer, sigmoid for the second layer. Loss function is binary cross-entropy, and optimizer is Adam. We set the batch size to 32, and train the model for 1000 epochs.

The accuracy for tensorflow method is 94.02%, which is a bit less than XG Boost model, but also significantly higher than the rest. From the confusion matrix, we can see that it is similar to the XG Boost method, which produce far less false positive cases than other models. The use of a neural network model with multiple layers allows the model to learn more complex representations of the input data. This can help the model better capture the patterns and relationships in the data, leading to higher accuracy on the binary classification task.

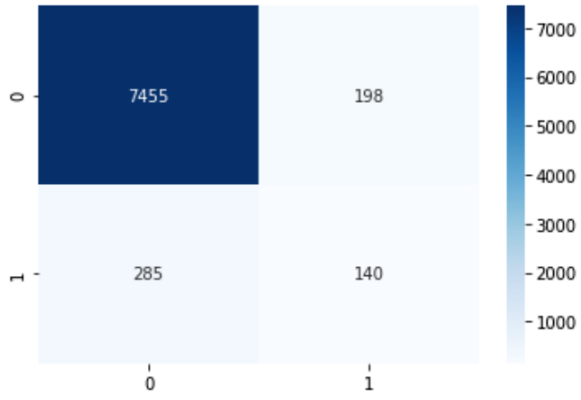

Figure 15: Confusion Matrix of Tensorflow

4.8 Model Enhancement: Ensemble

A voting classifier is an ensemble learning technique that combines the predictions of multiple individual machine learning models to make a final prediction. [9] The idea behind the voting classifier is that it takes the majority vote from different individual models to make a final prediction. In this case, we have a voting classifier using KNN, SVM, XG BOOST, Decision tree and logistic regression. Each of these models uses a different algorithm to make a prediction, and by combining them, we hope to improve the overall accuracy of the model.

The accuracy of the voting classifier is 92.93%. Though the accuracy improves, it is still less than tensorflow and XG Boost. While the voting classifier can be a powerful technique for improving the accuracy, it may not always be effective, especially if the individual models are not diverse or well-optimized, and the models' hyperparameters may not be tuned properly.

5 ADVANCED TECHNIQUES

In this part, advanced techniques are performed together with our best accuracy model XG BOOST in the previous section. Firstly, the data has a binary target variable, thus using K_means clustering to cluster the dataset into 2 clusters might help improve accuracy. Secondly, different age range typically has different features, while within an age range, it tends to be similar. As a result indicates, decision was made to cluster the age attribute for each gap of 10 years.

5.1 Clustering First

K-means clustering is a popular clustering algorithm that groups data points into k clusters based on their distance from the centroid of each cluster. In the context of binary classification, clustering the data using k-means can help to identify potential clusters of

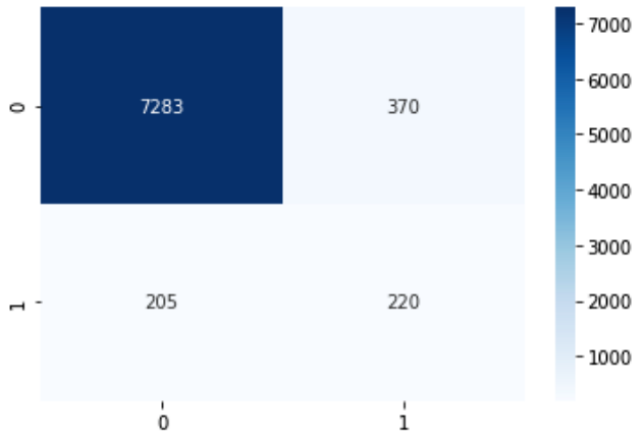

Figure 16: Confusion Matrix of Voting Classifier

data points that may share similar characteristics or patterns. K = 2 is chosen to cluster our data first, then use the XG Boost to do the binary classification, since it achieved the highest accuracy before.

After clustering first, the accuracy is 94.02%, which is a bit less than without clustering first method. This is because XG Boost is a highly complex algorithm that is designed to capture complex patterns and relationships in the data. If the binning process results in a loss of information or introduces biases in the data, the XG Boost algorithm may not be able to accurately capture these patterns, leading to lower accuracy.

5.2 Binning First

Binning is a common data preprocessing technique used to group continuous numerical data into discrete categories or bins. This technique can be useful in reducing noise or variation in the data and can make it easier to identify patterns or trends. For dataset, data are binned according to age, with an interval of 10 years.

After binning, XG Boost is chosen to perform the classification task, since it produces the highest accuracy before. The accuracy using XG Boost classifier after binning the age attribute is 94.03%, a little bit lower than not binning the age data. The reason for the low accuracy is the same as clustering the data first. Moreover, in a business and sociological point of view, term subscription is probably determined more base on the income and economic status of a person, rather than age.

6 COMPARISON

In this section, the models used on our dataset are compares with various evaluation metrics to show the overall performance, including accuracy, F-1 score, ROC curve and confusion matrix.

6.1 Accuracy

Below is the accuracy table. The XG Boost method achieves the highest accuracy, followed by Tensorflow and Voting Classifier, while SVM has significantly low accuracy than the rest of the models.

Table 1: Accuracy Table

	Accuracy
Logistic Regression	90.01%
Decision Tree	90.81%
SVM	87.67%
KNN	89.60%
Tensorflow	94.02%
XG Boost	94.62%
Voting Classifier	92.93%

6.2 Other Metrics

Other metrics that can evaluate the model includes precision recall and F-1 Score. Precision is the ratio of true positive predictions to the total number of positive predictions. Recall is the ratio of true positive predictions to the total number of actual positive cases.

F-1 score is the harmonic mean of precision and recall, and it is a combined measure of the model's accuracy and completeness. It balances the trade-off between precision and recall and provides a single metric that summarizes the model's performance. The F-1 score ranges from 0 to 1, with a higher score indicating better performance. XG Boost method has the highest F-1 score, indicating the overall performance is the best, which incorporate with the highest accuracy of the model.

Table 2: Metrics of Models

	Precision	Recall	F-1
KNN	27.07%	57.65%	36.84%
SVM	24.49%	64.47%	35.49%
Decision Tree	26.46%	42.95%	32.64%
Logistic Regression	25.00%	44.94%	32.13%
Voting Classifier	37.37%	51.53%	43.32%
XG BOOST	48.60%	40.71%	44.30%

6.3 ROC Curve

The AUC score measures the ability of the model to distinguish between positive and negative classes, by calculating the area under the ROC curve. The ROC curve is a plot of true positive rate against false positive rate for different classification thresholds. A higher AUC score indicates better performance of the model in correctly predicting the positive and negative classes.

From the ROC curve, classifier 4 corresponds to XG BOOST, performs the best since the AUC score is higher. Moreover, seeing the shape of the ROC curve, we can see the orange line and dark blue lines, which corresponds to decision tree and KNN classifier, shows that these two methods has an over fitting problem.

6.4 Confusion Matrix

As the confusion matrix shown above for each model, we could generalize the models to be 4 groups base on the distribution of the conrsion matrix. The first group is logistic regression and decision tree model. The second group is KNN and SVM. The third group is Tensorflow and XG boost and the last group is voting classifier.

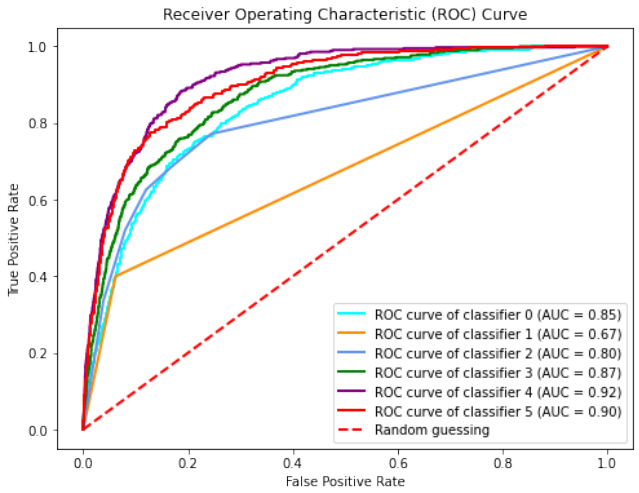

Figure 17: ROC Curve

Comparing group one and group three, group three has significantly low false positive and high true postive, while remain the other two classes similar. The second group has the highest false positive and true negative, while lowest false negative and true postive among all groups. Group four, which is voting classifier, has four categories in the confusion matrix balance of the three categories. Since it is voting classifier, it has characteristics of all classifier, and it makes sense that the four categories are in between.

7 CONCLUSION

In this analysis, we explored various binary classification techniques with the objective of training a model capable of effectively predicting whether a bank customer will subscribe to a term deposit or not based on 16 descriptive features. One of the most noteworthy challenges of this dataset was the class imbalance in the target variable, with a majority of the observations belonging to the negative class. To address this issue, we applied various techniques such as oversampling the minority class and using classification metrics such as AUC-ROC, accuracy, and F-1 to evaluate the performance of the models.

After preprocessing the features, we tested several classification models. The best performing model was XG Boost, which uses an ensemble of decision trees and is particularly effective in handling large datasets with complex relationships between features. XG Boost incorporates regularization techniques to prevent overfitting and improve the model's generalization performance. Additionally, it uses boosting, a technique that combines weak learners into a stronger ensemble, to improve the model's accuracy.

Moreover, we found out that clustering the data or binning the data according to age first do not necessarily increase the accuracy. This is because those methods would result in a loss of information, or introduce bias to the dataset. Also, in a business point of view, term subscription is probably determined more base on the economic status of a person, rather than age.

Overall, the performance of the model was evaluated using various classification metrics, and XG Boost was found to outperform other models in terms of accuracy and F-1 score. By addressing the challenges of class imbalance and feature selection, we were able to train a model that accurately predicted the bank customers who subscribed to a term deposit.

This high level of accuracy helps organizations identify the key factors that contribute to a customer's decision to choose a subscription, and refine and personalize marketing campaigns to more effectively target the right audience. Additionally, understanding the underlying customer behavior facilitates the development of customized products and services that address the specific needs and preferences of various customer segments. This can lead to increased customer satisfaction, loyalty, and long-term revenue.

ACKNOWLEDGMENTS

We would like to thank Professor Pawlicki for his guidance and teachings on the value of many data mining techniques. The T.A's have also been incredibly helpful throughout the semester, and their support has been invaluable. It has been a pleasure working with them, and their contributions have been vital to our success.

REFERENCES

[1] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, and Tianyi Zhou. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2* 1, 4 (2015), 1–4.

[2] Thomas J. Fan. 2022. sklearn.preprocessing.LabelEncoder. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[3] Alberto Fernández, Salvador Garcia, Francisco Herrera, and Nitesh V Chawla. 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research* 61 (2018), 863–905.

[4] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. 2003. KNN model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. 986–996.

[5] Vikramaditya Jakkula. 2006. Tutorial on support vector machine (svm). *School of EECS, Washington State University* 37, 2.5 (2006), 3.

[6] Michael P LaValley. 2008. Logistic regression. *Circulation* 117, 18 (2008), 2395–2399.

[7] Guillaume Lemaitre et al. 2023. sklearn.preprocessing.MinMaxScaler. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[8] J. Ross Quinlan. 1996. Learning decision tree classifiers. *ACM Computing Surveys (CSUR)* 28, 1 (1996), 71–72.

[9] Dymitr Ruta and Bogdan Gabrys. 2005. Classifier selection for majority voting. *Information fusion* 6, 1 (2005), 63–81.

[10] KRANTI WALKE. 2021. bank-full.csv (Ensemble Techniques). https://www.kaggle.com/datasets/krantiswalke/bankfullcsv