Prediction Model to identify potential customers for a banking institution

Anubhav Jain
Cloud Computing
Apex Institute of Technology
Chandigarh, India
jain.anubhav.2001@gmail.com

Tanu Aneja
Cloud Computing
Apex Institute of Technology
Chandigarh, India
tanuaneja001@gmail.com

Rohit Kumar
Cloud Computing
Apex Institute of Technology
Chandigarh, India
rohitkumarrb31@gmail.com

Hardik
Cloud Computing
Apex Institute of Technology
Chandigarh, India
shardik247@gmail.com

Dr. Ranjan Walia
Cloud Computing
Apex Institute of Technology
Chandigarh, India
ranjanwalia@gmail.com

***Abstract*— This research delves into an in-depth analysis of the data derived from a comprehensive marketing campaign spearheaded by Banco de Portugal. The central objective of this initiative was to augment the subscription rates of customers towards fixed-term deposit products, prominently encompassing Certificates of Deposit (CDs). Drawing upon a rich reservoir of knowledge acquired from relevant coursework, an array of sophisticated machine learning algorithms were systematically deployed. These algorithms were meticulously applied to unravel a critical question: What actionable strategies can financial institutions adopt to effectively market fixed-term deposit products, thereby optimizing efficiency and amplifying success rates? Through a meticulous examination of the intricate layers of data, this study endeavors to furnish nuanced insights and actionable recommendations tailored to augment the efficacy of marketing endeavors within the banking domain, with a particular focus on enhancing the allure and uptake of fixed-term deposit offerings among discerning consumers.**

I. Introduction

In today's fast-paced digital world, where smartphones, TVs, and the internet are part of our daily lives, advertising is everywhere we turn. Businesses face an uphill battle to capture our attention amidst the constant barrage of ads bombarding us from all angles. With so much noise, it's increasingly challenging for them to make their products stand out and connect with potential customers. This raises an important question: How can businesses make their advertising efforts as effective as possible, ensuring they reach the right people and achieve success? This question becomes particularly crucial in industries like banking, where institutions are constantly vying for customers' attention and trust. Take, for example, fixed-term deposit products offered by banks. These accounts, such as Certificates of Deposit (CDs), offer customers a safe and reliable way to grow their savings over time. However, in a crowded market with numerous financial institutions competing for customers, banks must find innovative ways to promote their deposit products and attract new customers. To tackle this challenge, banks need to leverage data-driven insights and cutting-edge technology. By analyzing data from past marketing campaigns, banks can uncover valuable information about their customers' preferences, behaviors, and responses to different advertising tactics. This data can reveal patterns and trends that help banks understand what resonates with their target audience and what doesn't. Moreover, by harnessing the power of machine learning and predictive analytics, banks can take their marketing efforts to the next level. These advanced techniques enable banks to anticipate customer needs and tailor their advertising messages accordingly. By segmenting customers based on their demographics, interests, and financial habits, banks can deliver personalized marketing campaigns that are more likely to resonate with individuals and drive engagement. In this study, we aim to explore how banks can leverage data and technology to optimize their marketing strategies for fixed-term deposit products. By examining a wide range of factors, including customer demographics, campaign effectiveness, and market dynamics, we seek to uncover actionable insights that enable banks to refine their advertising approaches and achieve greater success. Through our research, we hope to provide banks with practical recommendations for enhancing their marketing efforts, ultimately helping them attract more customers and grow their deposit business. By staying ahead of the curve and embracing data-driven strategies, banks can navigate the complex world of modern advertising with confidence and achieve their business objectives more effectively.

II. Literature survey

Marketing campaigns play a crucial role in today's business landscape, where competition for consumer attention is fierce. Banks, in particular, are constantly seeking ways to effectively promote their products and services to customers. In recent years, the integration of data science and machine learning techniques has emerged as a promising approach to enhance marketing strategies.

Several studies have explored the application of machine learning algorithms in predicting the success of bank marketing campaigns. These studies often utilize datasets containing information about customer demographics, campaign details, and market conditions. By analyzing this data, researchers aim to identify patterns and factors that influence customer behavior, such as subscription rates to fixed-term deposit products.

One common approach involves the use of logistic regression models, which are well-suited for binary classification tasks like predicting whether a customer will subscribe to a product or not. Decision trees and ensemble methods like Random Forest and AdaBoost have also been widely employed due to their ability to capture complex relationships within the data.

Evaluation metrics such as the Area Under the Curve (AUC) score are commonly used to assess the performance of predictive models. These metrics provide insights into the model's ability to differentiate between positive and negative outcomes, thereby guiding the selection of the most effective marketing strategies.

Furthermore, researchers often conduct hyperparameter tuning to optimize model performance. By fine-tuning parameters such as regularization coefficients and tree depth, they aim to improve the accuracy and reliability of their predictions.

The literature also highlights the importance of feature selection and interpretation. Identifying key variables that influence customer behavior, such as economic indicators and communication preferences, allows marketers to tailor their campaigns more effectively.

In conclusion, the integration of data science and machine learning techniques offers significant potential for enhancing bank marketing campaigns. By leveraging advanced analytical tools and methodologies, banks can gain valuable insights into customer behavior and develop targeted strategies to maximize campaign success.

[1]

Decision Trees are considered to be one of the most popular approaches for representing classifiers. Researchers from various disciplines such as statistics, machine learning, pattern recognition, and Data Mining have dealt with the issue of growing a decision tree from available data. This paper presents an updated survey of current methods for constructing decision tree classifiers in a top-down manner. The chapter suggests a unified algorithmic framework for presenting these algorithms and describes various splitting criteria and pruning methodologies.

[2]
Decision tree classifiers are regarded to be a standout of the most well-known methods to data classification representation of classifiers. Different researchers from various fields and backgrounds have considered the problem of extending a decision tree from available data, such as machine study, pattern recognition, and statistics. In various fields such as medical disease analysis, text classification, user smartphone classification, images, and many more the employment of Decision tree classifiers has been proposed in many ways. This paper provides a detailed approach to the decision trees. Furthermore, paper specifics, such as algorithms/approaches used, datasets, and outcomes achieved, are evaluated and outlined comprehensively. In addition, all of the approaches analyzed were discussed to illustrate the themes of the authors and identify the most accurate classifiers. As a result, the uses of different types of datasets are discussed and their findings are analyzed.

[3]
Presently, Customer Churn Prediction (CCP) becomes a tedious task among decision-makers and machine learning (ML) communities. Since the Internet of Things (IoT) and Cloud Computing (CC) platform generates a massive amount of customer data, it is necessary to construct a CCP model using the customer data from IoT devices. In this view, this paper devises a new model using optimal meta-heuristic based feature selection with Gradient Boosting Tree classification for CCP. The presented CCP model involves four main processes, namely data acquisition, preprocessing, feature selection and classification. At the earlier stage, the data collection process takes place utilizing IoT gadgets such as laptops, smartphones, wearable, etc. The IoT gadgets transmit the sensed data to the cloud data server (CDS). Then, gathered data is preprocessed and the missing values are imputed effectively.
Next, the ant colony optimization (ACO) algorithm is applied as a feature selector to select an optimal set of features. Finally, the GBT algorithm is employed as a classifier to classify the data into churn or non-churn. A comprehensive simulation was performed to indicate the betterment of the proposed model. The experimental results stated that the ACO-GBT model has reached a maximum sensitivity of 95.82%, specificity of 74.59%, accuracy of 92.71%, Fscore of 95.73% and kappa value of 70.71%.

[4]
In this paper we propose models for assessing the efficiency inlarge networks of bank branches. We distinguish bank branch efficiency into market and cost components suitably modified to capture different tiers of bank-management. The paper proposes a methodology which includes the use of multivariate analysis in order to ensure the homogeneity of the branches assessed and then data envelopment analysis for assessing efficiency. The methodology is applied on a sample of 580 branches of a commercial bank in the UK. The results obtained reinforced previous claims regarding the presence of high technical inefficiencies and economics/diseconomies of scale at the branch level from a production and cost point of view. Furthermore, the decision to pre-cluster the network of branches into homogenous groups has had profound implications on the magnitude of the assessed efficiencies.

[5]
We consider the problem of dynamically apportioning resources among a set of options in a worst-case on-line framework. The model we study can be interpreted as a broad, abstract extension of the well-studied on-line prediction model to a general decision-theoretic setting. We show that the multiplicative weight-update rule of Littlestone and Warmuth [10] can be adapted to this mode yielding bounds that are slightly weaker in some cases, but applicable to a considerably more general class of learning problems. We show how the resulting learning algorithm can be applied to a variety of problems, including gambling, multiple-outcome prediction, repeated games and prediction of points in ℝn. We also show how the weight-update rule can be used to derive a new boosting algorithm which does not require prior knowledge about the performance of the weak learning algorithm.

[6]
Ensemble classification is a data mining approach that utilizes a number of classifiers that work together in order to identify the class label for unlabeled instances. Random forest (RF) is an ensemble classification approach that has proved its high accuracy and superiority. With one common goal in mind, RF has recently received considerable attention from the research community to further boost its performance. In this paper, we look at developments of RF from birth to present. The main aim is to describe the research done to date and also identify potential and future developments to RF. Our approach in this

review paper is to take a historical view on the development of this notably successful classification technique. We start with developments that were found before Breiman's introduction of the technique in 2001, by which RF has borrowed some of its components. We then delve into dealing with the main technique proposed by Breiman. A number of developments to enhance the original technique are then presented and summarized. Successful applications that utilized RF are discussed, before a discussion of possible directions of research is finally given.

[7]
We introduce a method to predict or recommend high-potential future (i.e., not yet realized) collaborations. The proposed method is based on a combination of link prediction and machine learning techniques. First, a weighted co-authorship network is constructed. We calculate scores for each node pair according to different measures called predictors. The resulting scores can be interpreted as indicative of the likelihood of future linkage for the given node pair. To determine the relative merit of each predictor, we train a random forest classifier on older data. The same classifier can then generate predictions for newer data. The top predictions are treated as recommendations for future collaboration. We apply the technique to research collaborations between cities in Africa, the Middle East and South-Asia, focusing on the topics of malaria and tuberculosis. Results show that the method yields accurate recommendations. Moreover, the method can be used to determine the relative strengths of each predictor.

III. Methodology

A. *Programming in Python*

Python provides a number of packages and libraries for the convenience of the programmer. The whole project iscoded using *Python 3*. Packages/libraries used are *numpy* for array manipulation, *pandas* for dataframe operations, and *matplotlib* and *seaborn* for visualization. The *sklearn* libraries were also critical in providing packages for machine learning algorithms, tasks, and by giving the user the control to set important attributes of those algorithms as they wished. The dataset is stored in a dataframe and is intensively queried and manipulated using facilities provided by the Python 3 environment. Other data structures such as arrays, lists, and dictionaries are used as needed[1].

B. *Data cleaning and exploratory analysis*

The dataset was provided by the U. C. Irvine Machine

Learning Repository and contained information on 41,188 clients across 20 different features, both categorial (marital status, job type, education, etc.) and numeric (age, number of days since previous contact, etc.). The target variable is a binary “Yes” (client subscribed) or “No” (client did not subscribe).

The first step is to load the dataset into a dataframe for easy manipulation and exploration using the *pandas* package. The ‘duration’ feature was dropped due to the risk of data leakage. This feature measures the length of the phone call between the bank’s marketing representative and the customer. Since this time cannot be known until after the call has ended (when the outcome for that customer is already known), including it in a predictive model would not provide realistic results.

The next step was to explore and clean the categoricalvariables such as ‘job type,’ ‘marital status,’ ‘education,’ etc. Plots for each were produced that looked at their relativefrequency as well as normalized relative frequency. In Python,these graphs are created using the *seaborn* package.

Many of these features contain unknown values so the next question is how to deal with this missing data. Simply discarding these rows would lead to a huge reduction in the amount of data and thus greatly interfere with the results. Instead, these missing values are imputed using other independent variables to infer the missing values. While this does not guarantee that all the missing data will be restored, a majority of it will be. For instance, cross-tabulation between ‘job’ and ‘education’ was used based on the hypothesis that a person’s job will be influenced by their education. Thus, a person’s job is used to predict their education level. The Python function *cross tab* was created for this cross-tabulation step. A similar cross-tabulation process was carried out for the ‘house ownership’ and ‘loan status’ features. It’s important to note that in making these imputations, care was taken to ensure the correlations made sense in the real world. If not, the values were not replaced. Throughout this process, dataframes using the *pandas* package were invaluable. Python provides quickness, ease of modifiability and ease of replacement of values throughout the dataset thanks to this tool.

The next task is to deal with missing data among the numerical features. In this particular dataset, all missing values were encoded as ‘999.’ It’s quickly noted that while onlythe ‘pdays’ (number of days since that customer had been contacted from the previous campaign) column contained suchvalues, they made up the majority of the data for this feature. In other words, this column was missing more data than it contained. Further exploration showed that this missingness was due to customers who had not been contacted previously at all. To deal with this, the numerical feature ‘pdays’ was replaced with a categorical feature based on whether the customer had never been contacted, contacted 5 or less days ago, 6-15 days ago, etc.

Finally, a heatmap was created to show us whether there

is strong correlation between the target variable and any independent variables. The heatmap is created using Spearman correlation, which measures the degree to which the rankings of each variable (as opposed to the actual values) align, thus minimizing the effect of outliers[2]. Once this is measured, those variables are expected to be significant during the modeling stage. This graphic was created using Python’s *seaborn* package and the specially written function *drawheatmap,* which takes a dataframe as an input. The code for this function can be seen in the Jupyter notebook for this project.

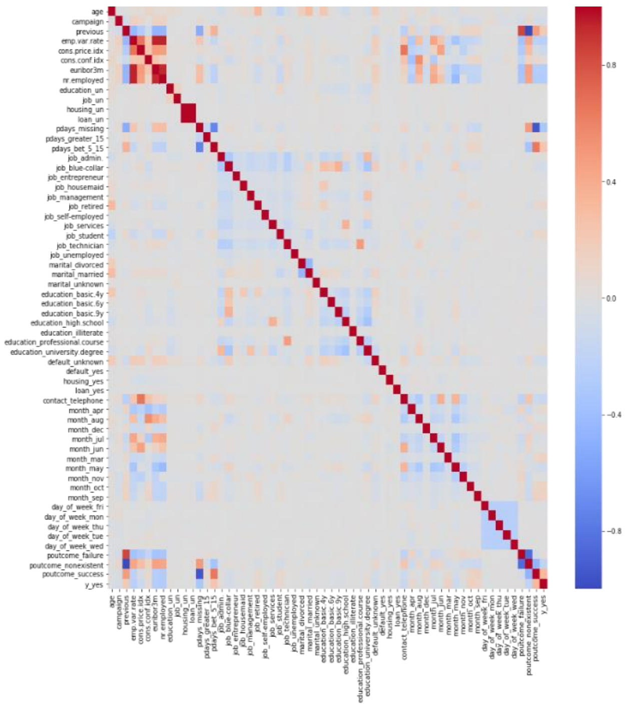

Fig. 1. Spearman correlation heatmap of rankings for each variable

For performing predictive analysis, many well known ma- chine learning models should be fit on training data to learn parameters of the model and then they can be run on test set to get the prediction. Models used in our project are discussed below. – – –

C. Model Building

The dataset is divided into training data and test data with the intention of using the training data to find the parameters of the particular model being used (fitting the model on the training data) and then applying this to thetest data to determine the model's performance and to draw conclusions about its predictive capability. This can be done with a *sklearn.cross validation.train test split* function call by specifying split ratio. the algorithm identifies the model that correctly classifies it. Predictions are made by a majority vote of the weak learners' predictions, weighted by their individual accuracy.

Gradient Boosting: Python provides the *sklearn.ensemble.GradientBoostingClassifier* package 1+e for Gradient Boosting classification. The Gradient Boosting model is a generalized version of AdaBoost. The objective is to minimize the loss of the model by adding weak learners using a gradient descent-like procedure. One new weak learner is added at a time and existing weak learners in the model are frozen and left unchanged.

D. Model Evaluation

For evaluating all the models built, AUC score is used. This is chosen as the scoring metric because it has been established that for cases where classes are unbalanced (such as this), AUC score is a better evaluation criterion than the accuracy score. For each model, five-fold cross-validation is performed over the training set. The *kfold* function from *sklearn* was used extensively for this step. The mean AUC score is calculated for each set of selected parameters. The final model (and hyper- parameters) are selected based on the highest out-of-sample mean AUC score.

E. Hyper-parameter Tuning

For all each model implemented, the hyper-parameters were tuned to obtain the optimal performance of the classifier.

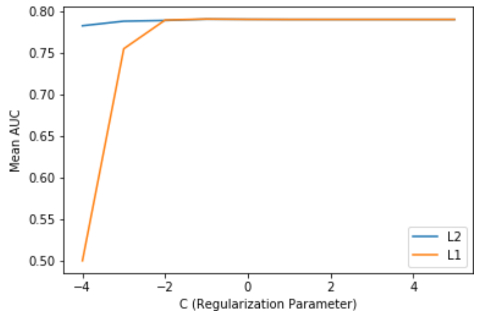

Fig. 2. Hyper-parameter tuning for Logistic Regression

Logistic Regression: For Logistic Regression, two hyper-parameters were tuned: the penalty type ('L1' or 'L2' penalty) and the regularization coefficient ('C': 10^{-4} to 10^{5} on the log scale). Below is the graph of mean AUC vs. C for the different penalty types. From the figure, it is clear that the classifier is quite robust to the C values and the penalty type. We obtain a maximum mean AUC of 0.7903 for C = 0.1 and penalty = 'L1'. This graph was created using Python's *matplotlib* package and a function we created called *plot mean auc LR* which can be seen in the accompanying Jupyter notebook.

Logistic Regression: Python provides the package *sklearn.linear model.LogisticRegression* for Logistic Regression. LR a is well known classification model.

The linear model fits the training data to the equation $y = w_0 + w_1x_1 + w_2x_2 + \ldots$ (where y stands for the target variable, w_0 stands for the y intercept, $x_1, x_2, x_3, \ldots$ are feature vectors, and $w_1, w_2, w_3, \ldots$ are their corresponding weights) while the logistic regression algorithm uses the same decision boundary with bit modifications as shown:$P(X) = {}^{1}{-}_{y}$

Decision Trees, Random Forest Classifier and Gradient Boosted Trees: For Decision Trees, Random Forest, and Gradient Boosted Trees, two hyper-parameters were tuned: minimum samples split (the minimum number of samples required to split an internal node) and minimum samples leaf (the minimum number of samples required to be at a leaf node). These two parameters help control the depth of thetrees and thus help to control the model's complexity.

Below are the graphs of mean AUC vs. leaf values for different split values. From the figures, it is clear that the classifiers were sensitive to the hyper-parameter chosen. These figures were created using *matplotlib* and the function *plotAUCDTRF* in the Jupyter notebook.

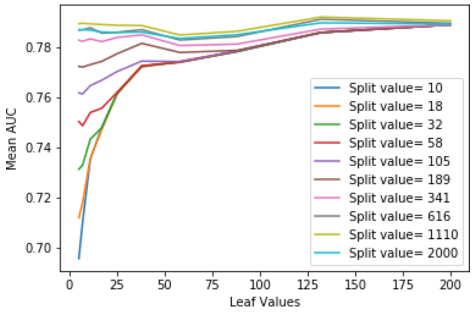

Fig. 3. Hyper-parameter tuning for Decision Trees

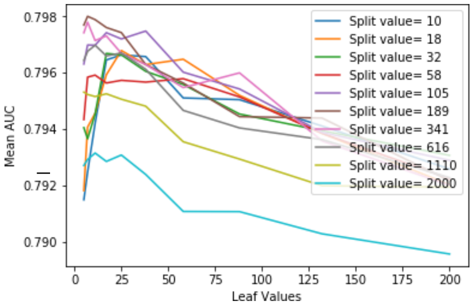

Fig. 4. Hyper-parameter tuning for Random Forest Classifier

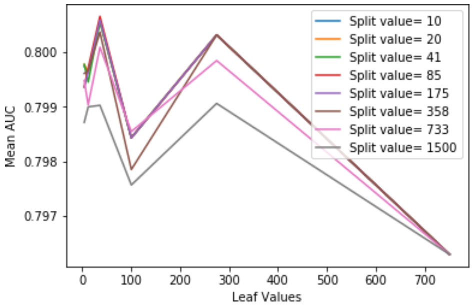

Fig. 5. Hyper-parameter tuning for Gradient Boosted Trees

Hyper-Parameter Tuning of Trees			
Model	Best Leaf Value	Best Split Value	Mean AUC
Decision Tree Classifier	132	1110	0.7919
Random Forest Classifier	7	189	0.7979
Gradient Boosted Trees	37	85	0.8006

In the table, the best tree based models are summarizedwith the hyper-parameters and the AUC score obtained forthe same.

AdaBoost Classifier: For the Ada-Boost classifier, only onehyper-parameter is tuned: the number of estimators. The higherthe number of estimators, the more complex the model andthe higher the chance of overfitting becomes. In Figure 6,we see a graph of mean AUC vs. number of estimators. As before, this graph was created with *matplotlib* and the function*plot mean auc Ada Boost*. From the figure, it is clear that the classifier is pretty sensitive to the estimator values. We obtain the maximum mean AUC of 0.8157 for $n_{\text{estimators}} = 1000$.

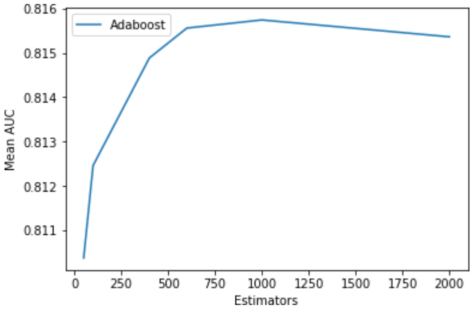

Fig. 6. Hyper-parameter tuning for Ada-Boosting Classifier

Results

From the above results, the best out of sample model performance was obtained for the AdaBoost Classifier with $n_{\text{estimators}} = 1000$. On the test data, the best AUC score achieved was 0.8036.

The importance of the features (in terms of how greatly they affected the coefficients) was also plotted. This provides valuable insight toward understanding which features contribute the most toward the models’ performance.

From the feature importance plot, it can be inferred that Europe’s Libor rate, age of the applicant, employment variation rate, campaign, consumer confidence index, consumer price index, mode of contact (= telephone), and number of employees are some of the most important features in predicting the outcome. The below graph was created using the *seaborn* package and the function *plotfeatureimportance.*

A. *Discussion*

Based on the feature importance plot, some recommendations can be made to the bank’s marketing team:

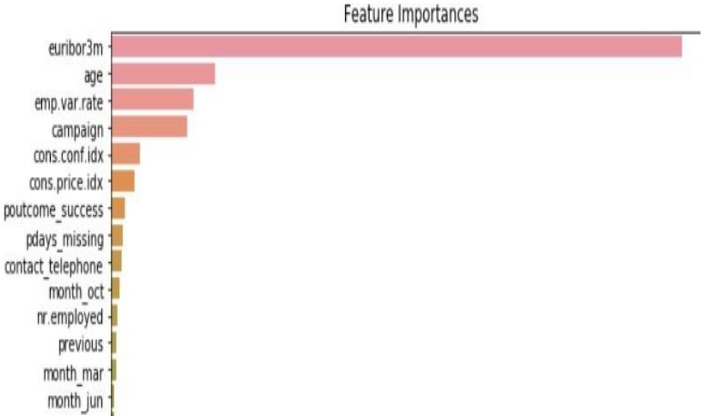

Fig. 7. Most important Features based on the AdaBoost model

- The marketing team should collaborate with economic experts so that as soon as they have some signals indicatingthe Libor going up (or the economic situation improving, i.e., consumer price index or consumer confidence index goes up), they can expect more customers to subscribefor the term deposit and should pro-actively reach out to them before the bank’s competitors do.
- The marketing team should target relatively old age customers who would be looking for safe and profitable investment options. The marketers should ensure to convey the peace of mind and steady source of income these products provide as a value proposition to these customers.
- Although the ‘duration’ (length of marketing phone call) variable was not used in the prediction models for various reasons cited earlier, the correlation of the ‘duration’ variable with the target variable shows that the higher the duration, the more likely it is that the customer will subscribe to the term deposits (correlation = 0.405). This makes intuitive sense because longer duration shows that the customer is interested in the product. Hence, the marketers should try to make the call engaging and increase the duration of the call.
- The telephone seems to be the most preferred mode of communication.
- The marketing team should prioritize those customers to whom they previously reached out during previous campaigns. They are likely to subscribe for the term deposit.

Conclusion

From this project, we learned how banks can improve their marketing campaigns by focusing their efforts on certain prime-grade clients and also how they can recognize market conditions which are favorable to increase client subscription for the fixed-term products they are offering. All of this was possible by implementing data science and machine learning methods in Python. Tools such as dataframes, arrays, for loops, etc. were all critical for the success of this project. A large number of other tools and techniques from the Python for Data Science course were used and these were invaluable for making our analyses and predictions. This project demonstrated how powerful Python can be for data science applications.

References

[1] Rokach, Lior & Maimon, Oded. (2005). Decision Trees. 10.1007/0-387-25465-X_9.

[2] Jijo, Bahzad & Mohsin Abdulazeez, Adnan. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. Journal of Applied Science and Technology Trends. 2. 20-28

[3] S. Venkatesh Dept. of Computer Science Metaheuristic based Optimal Feature Subset Selection with Gradient Boosting Tree Model for IoT Assisted Customer Churn Prediction.

[4] Athanassopoulos, Antreas D., Non-Parametric Frontier Models for Assessing the Market and Cost Efficiency of Large Scale Bank Branch Networks. Journal of Money, Credit, and Banking, Vol. 30 No. 2, May,1998

[5] Y. Freund, R. Shapire Computational Learning Theory, 1995, Volume 904 A decision-theoretic generalization of on-line learning and application to boosting

[6] Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: from early developments to recent advancements. Systems Science &amp; Control Engineering, 2(1), 602–609

[7] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. New York: Chapman & Hall.research collaborations using link prediction and random forest classifiers .