Predicting Bank Marketing Campaign Results

Amish Faldu
New Jersey Institute of Technology
Newark, USA
af557@njit.edu

Jaishnav Kapadia
New Jersey Institute of Technology
Newark, USA
jk756@njit.edu

Akaash Thawani
New Jersey Institute of Technology
Newark, USA
at953@njit.edu

Naga Ramya Vankayala
New Jersey Institute of Technology
Newark, USA
nv384@njit.edu

***Abstract*—For banking institutions, direct marketing campaigns are indispensable. They provide a direct channel to engage with clients and promote financial products. This project specifically targets the promotion of bank term deposits through phone calls.**

Our project focuses on analyzing data from direct marketing campaigns in banking. Our main goal is to create a predictive model that can forecast whether a client will subscribe to a bank's term deposit product. We have specific objectives: Examining client demographics, past campaign interactions, and economic indicators to understand the factors influencing client decisions. Developing a predictive model that can accurately predict whether clients will opt for the bank's term deposit product. Using the insights gained from our predictive model to enhance future marketing strategies and optimize resource allocation.

***Index Terms*—KNN, Decision Trees, Random Forest, Convolution 1D (neural network), XGBoost**

I. Introduction

We set out on a journey to explore various methodologies in our project COLAB[17]. We carefully navigated through different approaches, each representing a unique path towards our goal. Our exploration involved experimenting with a variety of algorithms, including KNN, Decision Trees, Random Forest, Convolution 1D (neural network), and XGBoost. A Look into Our Methods: We dive into the strategies we used to tackle the challenges of our project. With our understanding of AI, we approached the development of our methodology with careful consideration. We began by meticulously preparing our data, addressing missing values, and standardizing the dataset to ensure consistency. Then, we delved into the nuances of model training, fine-tuning each algorithm to achieve the best performance. We evaluated the performance of each model using a range of metrics. These metrics, such as accuracy, F1-score, precision, recall, and confusion matrices, offered valuable insights into the effectiveness of our methods. Additionally, for the convolution 1D neural network, we included loss as an additional metric to ensure a thorough evaluation. Furthermore, we used visualization techniques like ROC and AUC curves to gain deeper insights into the performance of our models. This careful approach not only improved our understanding of the algorithms but also guided our decision-making throughout the project.

II. Literature Review

Prediction of customer's response to telemarketing campaign of banks has been through many stages from the use of classical machine learning models [2]–[4] to complex deep learning architectures and finally through the use of state of the art techniques of ensemble learning.

Initially, Tekouabou et al. [1] employed Naïve Bayes (NB), Logistic Regression (LR), Decision Trees (DT), Artificial Neural Networks (ANN), and Support Vector Machines (SVM) to classify customer responses. Despite achieving a reported accuracy of 100 using decision trees, the absence of a confusion matrix and details regarding data split between training and testing sets render their results unreliable.

Additionally, Selma et al. [6] utilized an artificial neural network model for customer response prediction, achieving a high accuracy of 98 with a Multi-Layer Perceptron (MLP) architecture. However, their oversight of class imbalance within the dataset and lack of additional processing to address this issue cast uncertainty on the model's performance on imbalanced datasets.

Also , Saad et al. [7] used Undersampling Techniques to solve the imbalance problems , but their results and observations show that there may be a case of overfitting the data as the achieved massive accuracy gains compared to others, so we proposed to use new techniques like SMOTE to solve this problem.

Similarly, Farooqi et al. [5] employed decision trees, Naïve Bayes, K-Nearest Neighbors (KNN), and ANN, with decision trees outperforming other classifiers with an accuracy of 91.2. However, their evaluation solely on the training set undermines the robustness of their findings.

SMOTE (Synthetic Minority Over-sampling Technique) is considered an advanced technique for addressing class imbalance in machine learning. Unlike simpler methods such as random oversampling or undersampling, which may lead to overfitting or loss of information, SMOTE generates synthetic samples for the minority class by interpolating between existing minority class instances.

SMOTE works by creating synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. This effectively synthesizes new instances in the feature space, thereby balancing the class distribution without simply replicating existing minority class instances. By introducing synthetic samples, SMOTE helps in mitigating the risk of overfitting and improves the generalization ability of the model.

Through the examination of existing research, it becomes evident that no singular model consistently outperforms others in telemarketing data prediction. Challenges such as feature selection and class imbalance impede the identification of an optimal model. Hence, this study aims to mitigate class imbalance in the dataset, highlight significant features, and propose various machine learning models with the corrected data.

III. Data Description

We have used the Portuguese bank marketing dataset publicly available at UCI dataset repository [16]. The dataset consists of 41188 instances, 21 attributes. The attributes of the dataset are:

- age : age in years
- job : type of job
- marital : marital status
- education : educational level of the individual
- default: has credit in default?
- housing: has housing loan?
- loan: has personal loan?
- contact: contact communication type
- month: last contact month of year
- dayofweek: last contact day of the week
- duration: last contact duration, in seconds
- campaign: number of contacts performed during this campaign and for this client
- pdays: number of days that passed by after the client was last contacted from a previous campaign
- previous: number of contacts performed before this campaign and for this client (numeric)
- poutcome: outcome of the previous marketing campaign
- emp.var.rate: employment variation rate - quarterly indicator
- cons.price.idx: consumer price index - monthly indicator
- cons.conf.idx: consumer confidence index - monthly indicator
- euribor3m: euribor 3 month rate - daily indicator
- nr.employed: number of employees - quarterly indicator
- y - has the client subscribed a term deposit

IV. Data Analysis

Following points can be inferred from histograms (Fig. 1):

- Some attributes like duration, campaign are not normalized. So it needs to be normalized before training and testing
- ”pdays” and ”previous” attribute doesn’t contribute much to the final outcome as most of the customers that are contacted in campaign were new and only few of them were contacted again in current campaign.

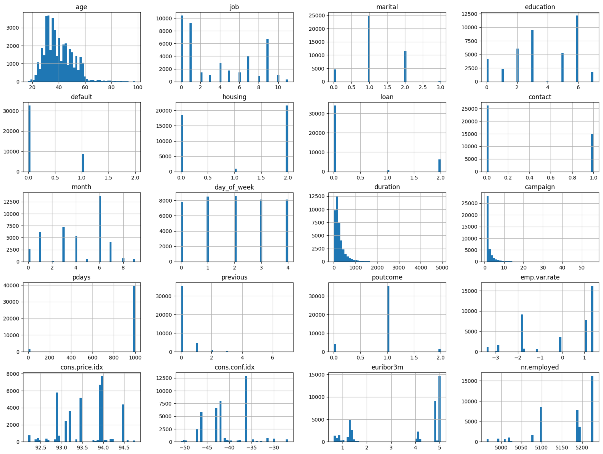

Fig. 1. Histograms of Attributes

- ”campaign” attribute, a count of how many times a person was contacted in this campaign also including last contacted, doesn’t add much value. As most of the values are 1 and small chunk of records have values less than 1.

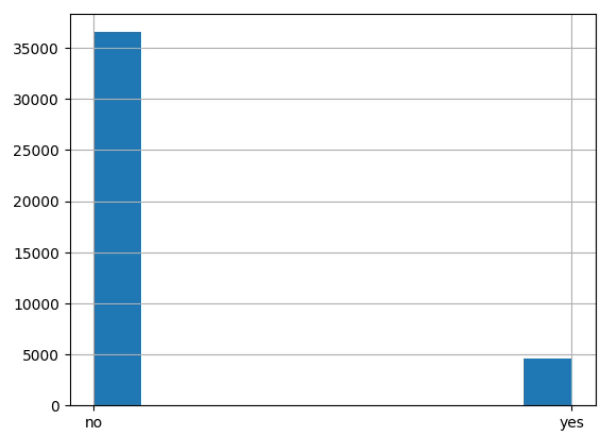

Fig. 2. Label Distribution

From label distribution (Fig. 2), it can be observed that the dataset has imbalanced class labels. More than 35000 records are labelled as no and less than 5000 records are labelled as yes.

V. Data Cleaning

Based on what we’ve observed in the histograms (Fig. 1), a few things stand out. Firstly, some attributes like duration and campaign aren’t spread out in a typical way, so we’ll need to normalize it before we start training and testing our models. Secondly, ’pdays’ and ’previous’, were removed because they don’t play a big role in predicting the final outcome.

In addressing the imbalance within the dataset, the Synthetic Minority Over-sampling Technique combined with Edited Nearest Neighbors (SMOTEENN) emerged as a viable solution. As depicted in Fig. 2, the label distribution highlights a substantial class imbalance, with a notable contrast between the 'yes' and 'no' labels, where 'no' instances dominate by a considerable margin. By leveraging SMOTEENN, which synthesizes minority class samples while simultaneously cleaning noisy data through nearest neighbor editing, a more balanced representation of the dataset can be achieved. Through this approach, the inherent biases stemming from class imbalance can be mitigated, fostering improved model generalization and predictive performance. However, the test dataset remanins untouched by this technique and was kept apart in its original form to represent real world scenario.

VI. Performance Evaluation Metrics

A. Precision

Precision assesses the proportion of correctly classified instances among those identified as positives. Mathematically, it is represented as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

B. Recall (Sensitivity)

Recall measures the proportion of correctly classified positive instances out of all actual positive instances. It can be expressed as:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

C. Specificity

Specificity evaluates the proportion of correctly classified negative instances out of all actual negative instances. It is calculated as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{3}$$

D. Accuracy

Accuracy quantifies the overall correctness of the classifier, considering both true positives and true negatives. It is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

E. F1 Score

The F1 Score represents the harmonic mean of Precision and Recall, providing a balanced evaluation. It is given by:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

F. Area Under the ROC Curve (AUC-ROC)

AUC-ROC evaluates the area under the receiver operating characteristic (ROC) curve, which illustrates the trade-off between true positive rate (sensitivity) and false positive rate (1-specificity). Higher AUC-ROC values signify better classifier performance.

VII. Different Models with Tuning

We explore the performance of various machine learning models on the same dataset through rigorous hyperparameter tuning. By employing a systematic approach to model selection and hyperparameter optimization, we aim to identify the most effective algorithm for the given task. Each model undergoes a comprehensive tuning process, where we experiment with different hyperparameters and techniques to maximize predictive accuracy and generalization performance. Through this comparative analysis, we gain insights into the strengths and weaknesses of each model and highlight the importance of hyperparameter tuning in improving model performance. By leveraging multiple algorithms and tuning strategies, we strive to uncover the best-performing model architecture tailored to the characteristics of the dataset

A. XGBoost Model

We employed XGBoost classifier with 10-fold grid search cross-validation strategy to fine-tune its hyper-parameters. Objective function used was 'binary:logistic', tailored for binary classification tasks .The parameters explored include:

- 'eta': [0.3, 0.6] - exploring learning rates of 0.3 and 0.6.
- 'max_depth': [4, 8, 12] - varying maximum tree depths between 4, 8, and 12.
- 'sampling_method': ['uniform', 'gradient_based'] - evaluating sampling methods as 'uniform' and 'gradient_based'.
- 'grow_policy': ['depthwise', 'lossguide'] - considering growth policies as 'depthwise' and 'lossguide'.

The best hyper-parameters discovered are:

- 'eta': 0.3
- 'max_depth': 12
- 'sampling_method': 'uniform'
- 'grow_policy': 'depthwise'

The classifier achieved AUC of 0.88 on test dataset with the best hyper-parameters.

B. Convolutional 1D NN Model

Training the Conv-1D neural network involves a series of steps crucial for its effectiveness. First, we construct the network architecture using a sequence of layers including convolutional, activation, batch normalization, and dropout layers. This architecture is tailored to handle one-dimensional data efficiently, making it suitable for our task. Next, we set the hyper-parameters such as the learning rate (1e-1), batch size (64), and number of epochs (20). These parameters play a vital role in determining how the model learns from the data and improves over time. We then define the loss function as the Cross-Entropy Loss, which is well-suited for classification tasks, and opted for the stochastic gradient descent (SGD) optimizer to update the model's parameters during training. Additionally, we incorporate step learning rate scheduler to adjust the learning rate dynamically, which helps in optimizing the model's performance. The convolution network achieved AUC of 0.86 on test dataset.

TABLE I
COMPARISON OF PREVIOUS METHODOLOGIES

S.No.	Ref.	Year	Methodology	Model-1	Model-2	Model-3	Model-4
1	[1]	2019	DT,ANN,RF	(DT) Accuracy = 1.00	(NN) Accuracy = 0.93	(RF) Accuracy = 0.89	-
2	[2]	2018	LR,LASSO,RF,ANN	(LR) Accuracy = 0.85	(LASSO) Accuracy = 0.857	(RF) Accuracy = 0.90	(ANN) Accuracy = 0.89
3	[3]	2019	KNN,DT,RF,NB	(KNN) Accuracy = 0.89	(DT) Accuracy = 0.82	(RF) Accuracy = 0.88	(NB) Accuracy = 0.81
4	[4]	2019	DT,NB,RF,KNN	(DT) Accuracy = 0.90	(NB) Accuracy = 0.87	(RF) Accuracy = 0.89	(KNN) Accuracy = 0.88
5	[5]	2019	DT,KNN,ANN,NB	(DT) Accuracy = 0.91	(KNN) Accuracy = 0.87	(ANN) Accuracy = 0.89	(NB) Accuracy = 0.87
6	[6]	2019	ANN	(ANN) Accuracy = 0.98	-	-	-
7	[7]	2019	DT,LR,RF,KNN	(DT) Accuracy = 0.92	(LR) Accuracy = 0.79	(RF) Accuracy = 0.94	(KNN) Accuracy = 0.87
8	[8]	2017	MLPNN, DT, LR,RF	(MLPNN) Accuracy = 0.82	(DT) Accuracy = 0.84	(RF) Accuracy = 0.86	(LR) Accuracy = 0.83
9	[11]	2019	SVM,KNN,DT,RF	(SVM) Accuracy = 0.76	(DT) Accuracy = 0.81	(KNN) Accuracy = 0.76	(RF) Accuracy = 0.83
10	[13]	2019	RBFNN	(RBFNN) Accuracy = 0.953	-	-	-
11	[9]	2021	LR,NB,SVM,RF	(SVM) Accuracy = 0.92	(LR) Accuracy = 0.90	(NB) Accuracy = 0.69	(RF) Accuracy = 0.99
12	[10]	2022	SVM,LR,RF,DT	(SVM) Accuracy = 0.90	(LR) Accuracy = 0.79	(RF) Accuracy = 0.94	(DT) Accuracy = 0.91
13	[12]	2022	(MWMOTE) Many	(SVM) Accuracy = 0.81	(NB) Accuracy = 0.75	(DT) Accuracy = 0.76	(RF) Accuracy = 0.93
14	[14]	2022	RF	(RF) AUC = 0.8611	-	-	-
15	[15]	2022	PNN , ELM	(PNN) Accuracy = 0.90	(ELM) Accuracy = 0.69	-	-

TABLE II
MODEL EVALUATION METRICS

Model	Accuracy	Precision	Recall	F1-Score	AUC
Convolutional 1D NN	0.82	0.37	0.91	0.53	0.86
XGBoost	0.89	0.50	0.88	0.63	0.88
Decision Tree	0.88	0.47	0.85	0.61	0.86
KNN	0.88	0.47	0.70	0.56	0.80
Random Forest	0.85	0.42	0.95	0.58	**0.89**

C. KNN Model

We employed KNN classifier with 2-fold grid search cross-validation strategy to fine-tune its hyper-parameters. The hyper-parameters explored include:

- 'n_neighbors': [3, 6, 9, 12] - choosing number of neighbors considered for prediction.
- 'weights': ['uniform', 'distance'] - governing the weighting strategy for neighbors.
- 'p': [1, 2] - the power parameter for the Minkowski distance metric.

The best hyper-parameters discovered are:

- 'n_neighbors': 3
- 'weights': 'distance'
- 'p': 1

The classifier achieved AUC of 0.80 on test dataset with the best hyper-parameters.

D. Decision Tree Model

We then leveraged Decision Tree Classifier, a simple tree based classifier with 10-fold grid search cross-validation to fine-tune hyper-parameters. The hyper-parameters explored were:

- 'criterion': ['gini', 'entropy'] - dictates the criterion for splitting nodes.
- 'splitter': ['best', 'random'],
- 'max_depth': [10, 20, 30, 40, 50] - governing the maximum depth of the tree.

The best hyper-parameters discovered are:

- 'criterion': 'gini'
- 'max_depth': 40
- 'splitter': 'best'

The classifier achieved AUC of 0.86 on test dataset with the best hyper-parameters.

E. Random Forest Model

Finally, we trained ensemble-based Random Forest classifier with same 10-fold grid search cross-validation strategy to fine-tune its hyper-parameters and select best model. The parameters explored include:

- 'n_estimators': [100, 200, 300] - .
- 'max_depth': [8, 10, 12].
- "ccp_alpha": [5e-4, 1e-3].

The best hyper-parameters discovered are:

- 'ccp_alpha': 0.0005
- 'max_depth': 12
- 'n_estimators': 200

The classifier achieved AUC of 0.89 on test dataset with the best hyper-parameters.

F. Suggestions

The dataset we used has meaningless entries like "unknown" in all fields, which doesn't help classifier find decision boundary. The dataset was also quite small, with only 18 meaningful attributes. Adding more attributes like wages, economic conditions, job position, family size, and location could improve it. We tested 5 different classifiers, but there's still room to improve by trying new classifiers, adjusting the ones we used or using different sampling techniques.

CONCLUSION

In conclusion, our project aimed to predict client subscriptions to bank term deposit products using direct marketing campaigns. We developed a predictive model, with the Random Forest performing the best among all algorithms tested.

Despite challenges like varying execution times, particularly with the Random Forest algorithm, our optimizations ensured project efficiency. Our findings offer insights for improving direct marketing strategies in banking, emphasizing the importance of careful model selection and algorithm optimization using grid search cross-validation. Overall, our project contributes to advancing predictive analytics in the banking sector, with implications for future research and industry applications.

REFERENCES

[1] Koumetio Tekouabou, Cédric Stéphane Cherif, Walid Hassan, Silkan. (2019). A data modeling approach for classification problems: application to bank telemarketing prediction. 1-7. 10.1145/3320326.3320389.

[2] M. Asif. "Predicting the Success of Bank Telemarketing using various Classification Algorithms." Degree project, Masters in applied statistics, Örebro University School of Business, (2018)

[3] E. Zhang, "The Application of KNN in Bank Marketing", BCPBM, vol. 36, pp. 285–290, Jan. 2023, doi: 10.54691/bcpbm.v36i.3444.

[4] A. Ilham, L. Khikmah, I.B.A.I. Iswara. Long-term deposits prediction: a comparative framework of classification model for predict the success of bank telemarketing. Journal of Physics: Conference Series 1175(1), 012035, 2019.

[5] R. Farooqi , N. Iqbal. Performance evaluation for competency of bank telemarketing prediction using data mining techniques. International Journal of Recent Technology and Engineering, 8(2), 5666-5674, 2019.

[6] G. Mumtaz et al., "Classification and Prediction of Significant Cyber Incidents (SCI) Using Data Mining and Machine Learning (DM-ML)," in IEEE Access, vol. 11, pp. 94486-94496, 2023, doi: 10.1109/ACCESS.2023.3249663.

[7] S. E. Saeed, M. Hammad and A. Alqaddoumi, "Predicting Customer's Subscription Response to Bank Telemarketing Campaign Based on Machine learning Algorithms," 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, 2022, pp. 1474-1478, doi: 10.1109/DASA54658.2022.9765152.

[8] J. Asare-Frempong and M. Jayabalan, "Predicting customer response to bank direct telemarketing campaign," 2017 International Conference on Engineering Technology and Technopreneurship (ICE2T), Kuala Lumpur, Malaysia, 2017, pp. 1-4, doi: 10.1109/ICE2T.2017.8215961.

[9] Y. Dubey, P. Wankhede, T. Borkar, A. Borkar and K. Mitra, "Diabetes Prediction and Classification using Machine Learning Algorithms," 2021 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON), Dhaka, Bangladesh, 2021, pp. 60-63, doi: 10.1109/BECITHCON54710.2021.9893653.

[10] H. Al-Zubaidi, M. Dweik and A. Al-Mousa, "Stroke Prediction Using Machine Learning Classification Methods," 2022 International Arab Conference on Information Technology (ACIT), Abu Dhabi, United Arab Emirates, 2022, pp. 1-8, doi: 10.1109/ACIT57182.2022.10022050.

[11] N. Bhartiya, S. Jannu, P. Shukla and R. Chapaneri, "Employee Attrition Prediction Using Classification Models," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-6, doi: 10.1109/I2CT45611.2019.9033784.

[12] S. P. Raja, B. Sawicka, Z. Stamenkovic and G. Mariammal, "Crop Prediction Based on Characteristics of the Agricultural Environment Using Various Feature Selection Techniques and Classifiers," in IEEE Access, vol. 10, pp. 23625-23641, 2022, doi: 10.1109/ACCESS.2022.3154350.

[13] A. N. Puteri, Dewiani and Z. Tahir, "Comparison of Potential Telemarketing Customers Predictions with a Data Mining Approach using the MLPNN and RBFNN Methods," 2019 International Conference on Information and Communications Technology (ICOIACT), Yogyakarta, Indonesia, 2019, pp. 383-387, doi: 10.1109/ICOIACT46704.2019.8938496.

[14] H. Zhao, X. Zuo and Y. Xie, "Customer Churn Prediction by Classification Models in Machine Learning," 2022 9th International Conference on Electrical and Electronics Engineering (ICEEE), Alanya, Turkey, 2022, pp. 399-407, doi: 10.1109/ICEEE55327.2022.9772553.

[15] L. Chen, X. Li, L. Ma and S. Bi, "Probabilistic neural network based apple classification prediction," 2022 International Conference on Advanced Mechatronic Systems (ICAMechS), Toyama, Japan, 2022, pp. 70-75, doi: 10.1109/ICAMechS57222.2022.10003415.

[16] Moro,S., Rita,P., and Cortez,P.. (2012). Bank Marketing. UCI Machine Learning Repository. doi: 10.24432/C5K306.

[17] "Python implementation of the project" [Code]. Available: https://colab.research.google.com/drive/1WLOExhJ2LJUD-YrdxNo5eekuh5trBGqG