Prediction of Marketing Campaign

Name:	**Mridul Sharma**
Registration No./Roll No.:	21176
Institute/University Name:	IISER Bhopal
Program/Stream:	EECS
Problem Release date:	August 17, 2023
Date of Submission:	November 19, 2023

1 Introduction

The main objective of this project is to construct a reliable supervised machine learning framework for predicting the success or failure of marketing campaigns based on customer responses. Our dataset comprises 2016 training instances and 224 test instances, encompassing 25 distinct features. The binary outcome, representing either success ('1') or failure ('0'), serves as the target variable. Within the training data, there are 301 instances classified as success (1) and 1715 instances classified as failure (0). Notably, the feature 'income' exhibits 21 missing values, addressed through mean/median imputation. As the nature of the target classes is discrete, framing the problem as a classification task is evident. The result allows us to judge the performance of the different models used with the help of different factors such as accuracy, precision, recall, etc.

2 Methods

Implementation will rely on the Python programming language, utilizing established machine learning models like k-Nearest Neighbors (kNN), Decision Tree, and others for classification. The methodology spans data pre-processing, model training with optimized hyperparameters, and rigorous evaluation metrics—accuracy, precision, recall, and F1 score. Cross-validation techniques will ensure robust model performance. The Different 'Machine Learning' models used in this project are: **1. AdaBoost**
2. Decision Tree
3. Logistic Regression
4. Multinomial Naive Bayes
5. Random Forest
6. k Nearest Neighbours
GitHub Link For Project - `https://github.com/mridul1404/Mridul_Sharma_ECS308_project`

3 Experimental Setup

The Different Evaluation criteria used to evaluate the different models are 'Accuracy', 'Precision', 'Recall', and 'F-measure'. The different Python libraries used for all the models in our project are, namely numpy, pandas, sklearn.

4 Results and Discussion

If **precision** is taken as the top priority to judge the performance of the model, then AdaBoost or random forest both might be suitable; in case of **recall**, kNN and AdaBoost both can be considered; in case of **F-measure**, once again, AdaBoost and Random Forest are good choices. So, in the end,

Table 1: Performance Of Different Classifiers Using All Features

Classifier	Precision	Recall	F-measure
Adaptive Boosting	0.6474	0.6279	0.6358
Decision Tree	0.6695	0.5322	0.5272
K-Nearest Neighbor	0.6315	0.6136	0.6212
Logistic Regression	0.5729	0.6226	0.5700
Random Forest	0.6699	0.6131	0.6311
Multinomial Naive bayes	0.5752	0.6261	0.5732

we can say that among the models tested in this project, **Random Forest** comes out to be the best classifier among each other.

5 Conclusion

The evaluation of the classification models highlights Random Forest as a well-balanced performer across all precision, recall, and F-measure, making it a strong candidate for accuracy. K-Nearest Neighbor demonstrates competence in capturing instances of both classes, while Adaptive Boosting consistently performs reliably. For future work, exploring ensemble methods, hyperparameter tuning, and addressing class imbalance are suggested, along with considerations for model interpretability and real-world deployment aspects.

References

1. Tom Mitchell. Machine Learning. McGraw Hill, 1997.
2. Leo Breiman. Random forests. Machine learning, 45(1):5–32, 2001.
3. C. D. Manning, P. Raghavan, and H. Schutze. Introduction to Information Retrieval. Cambridge University Press, New York, 2008.
4. K. Fukunaga and L. D. Hostetler. K-nearest neighbor bayes risk estimation. IEEE Transactions on Information Theory, 21(3):285–293, 1975.
5. R. Duda, P. Hart, and D. G. Stork. Pattern Classification. Wiley, 2000.