Mini Project 1

Vinay Sanga

1 Introduction

This project focuses on predicting the outcomes of a banking institution's direct marketing campaigns for term deposits based on historical data. By analyzing client information and campaign interactions, the goal is to determine whether a client will subscribe ('yes') or not ('no') to a term deposit, thereby enhancing the efficiency of future marketing strategies.

2 Data Preprocessing

Initially, I did the bare minimum changes to the dataset. This was done in order to get the models' performance on the organic dataset, thereby providing a baseline to compare our improvements against.
Afterwards, I did feature selection and oversampling to improve the performance. The initial and improved results are discussed in detail in the further sections.

- **Feature removal:** The `duration` variable was removed to ensure the model's applicability in real-world scenarios, as this feature is unknown before a call is made.
- **Missing Values:** The dataset did not have any missing values.
- **Categorical Variables:** Were one-hot encoded to transform them into a numerical format, which is necessary for model training. The `get_dummies` function was used, with `drop_first=True` to avoid the dummy variable trap.
- **Feature Scaling:** I standardized the data using `StandardScalar` to improve convergence.
- **Feature Selection:** There were 4 features (fig. 1)'cons.price.idx', 'euribor3m', 'nr.employed', 'emp.var.rate' which had high correlation. I dropped these features while remodeling.
- **Handling Imbalanced Data:** The class distribution was addressed by using

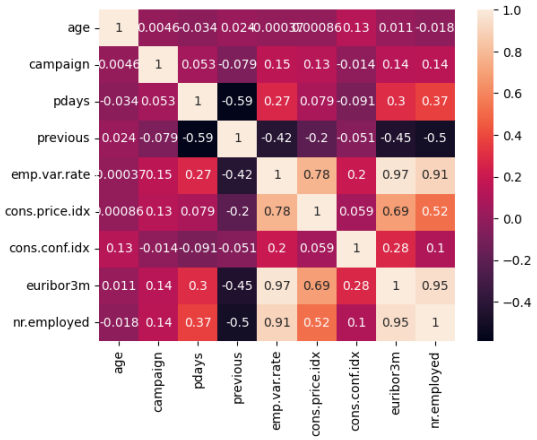

Figure 1: Features correlation heatmap

ADASYN oversampling technique to balance the classes, as there was a class imbalance skewed towards 'no'. In fig. 2, **0** refers to samples where clients did not subscribe ('no') and **1** refers to 'yes'.

```
Before resampling:
[(0, 36548), (1, 4640)]
After resampling:
[(0, 36548), (1, 36434)]
```

Figure 2: Resampling the data

3 Modelling

3.1 Algorithms and Training

For our predictive models, I chose Logistic Regression and MLP Classifier due to their distinct properties. Logistic Regression serves as a robust baseline that offers interpretability and simplicity, making it ideal for initial analysis. The MLP Classifier, a type of neural network, provides the capacity to capture complex non-linear relationships that may exist within the data. Both models were trained on the dataset, and their hyperparameters were carefully tuned to optimize performance. The models' ability to generalize was ensured through

cross-validation techniques, which also helped in mitigating any potential overfitting issues.

- **Logistic Regression:** Utilized with cross-validation (LogisticRegressionCV), ensuring robustness and generalizability. The model was trained with a maximum of 1000 iterations and a 10-fold cross-validation to optimize its performance.
- **MLP Classifier (Neural Network):** A more complex model than Logistic Regression, capable of capturing non-linear relationships. It was configured with early stopping to prevent overfitting, a maximum of 500 iterations, and a validation fraction of 10% to monitor the validation loss for early stopping.

3.2 Model Evaluation and Improvement

- Initially the accuracy was very high (over 90%) for both the models, but the precision and recall were poor (fig. 3). This clearly shows that accuracy alone is not the best metric for classification.
- Also, there was no feature selection performed in the first iteration of the experiment.

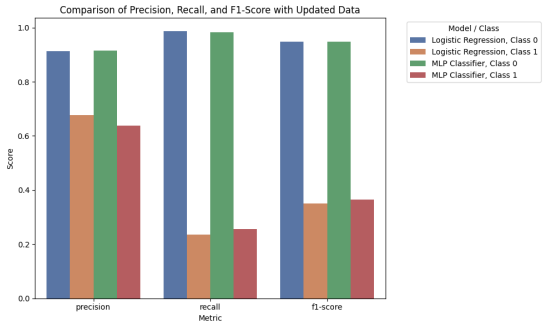

Figure 3: Precision, Recall and F1-Score **before** feature selection and resampling

- During the next iteration, I did feature selection and removed 4 highly correlated features.
- Diving deeper into the dataset showed that there is a huge class imbalance (fig. 4) of almost 9:1. Due to this recall was extremely low. To mitigate this, I used oversampling to balance the classes which brought this ratio to almost 1:1 (fig. 2). After this, the models performed really well.
- Even though the accuracy was reduced (86% for logistic regression and 88% for MLP), but the precision and recall (fig. 5) went up drastically. This is a much better model than the previous one.

```
no      88.734583
yes     11.265417
```

Figure 4: Class imbalance

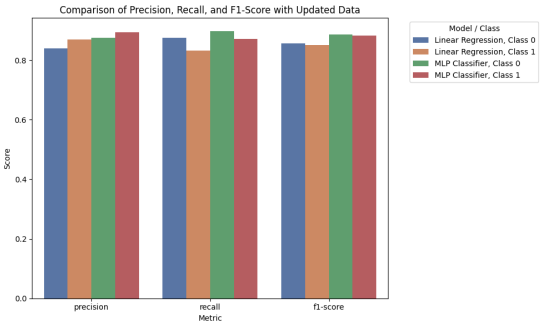

Figure 5: Precision, Recall and F1-Score **after** feature selection and resampling

- The same thing can be seen from ROC curve before changes (6) and after the changes (7).

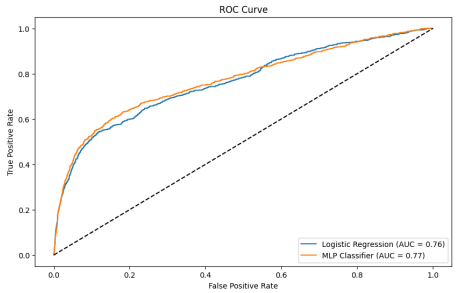

Figure 6: ROC Curve **before** feature selection and resampling

- Logistic Regression has an AUC of 0.92, indicating a strong ability to classify both the positive and negative classes correctly.
- MLP Classifier has an AUC of 0.95, which is exceptional, and shows it performs very well at distinguishing between the classes after rebalancing.

4 Conclusion

- Rebalancing the data has significantly improved the performance of both Logistic Regression and MLP Classifier models.

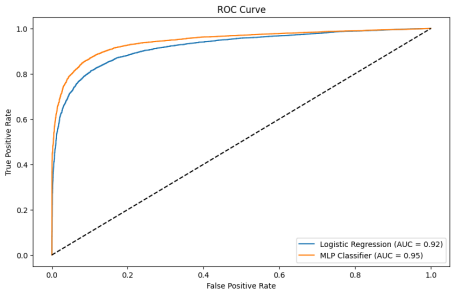

Figure 7: ROC Curve **after** feature selection and resampling

- The MLP Classifier, in particular, shows substantial gains in recall and F1-score for the minority class, and its AUC indicates excellent predictive capabilities post-rebalancing.
- The Logistic Regression model also benefits from rebalancing, as evidenced by the improved metrics.
- Overall, the rebalancing of the dataset has enhanced the ability of both models to accurately predict the minority class.
- The bottlenecks were highly correlated features and imbalanced dataset. They were overcome by appropriate measures as discussed above.
- The MLP Classifier shows superior performance as compared to Logistic Regression.