**ASSIGNMENT 3. DATA ANALYSIS PROJECT FOR MARKETING CAMPAIGNS**

|  |  |
| --- | --- |
| Student Name: | Student ID: |
|  |  |
|  |  |
| Bui The Hai | 24683423 |
|  |  |
|  |  |

**Table of Contents**

[I. Introduction: 2](#_Toc150112930)

[II. Business understanding: 3](#_Toc150112931)

[III. Data preparation: 3](#_Toc150112932)

[IV. Modeling results: 5](#_Toc150112933)

[**APPENDICES** 15](#_Toc150112964)

[**REFERENCES** 16](#_Toc150112965)

1. **Introduction:**

In the realm of data-driven marketing, predicting, and classifying customer responses to campaigns is essential. This project delves into binary classification to forecast Telecom plan subscriptions. Leveraging demographic, economic, and key variables, we build and assess predictive models using diverse algorithms and accuracy metrics.

# **Business understanding:**

In the telecom industry, the cost of acquiring a new customer can be substantial. Keeping customers satisfied and retaining their loyalty is a long-term objective. So, information gained from this analysis will inform strategies for achieving this goal.

Moreover, the telecom industry is also influenced by macroeconomic factors, such as employment rates, consumer price indices and interest rates. By understanding these external factors, our analysis aims to provide more insights that help the company enhance their effectiveness and reduce costs, thereby contributing to their growth and success.

# **Data preparation:**

The below figure summary information about columns in the data frame.



**Figure 1. Summary information of columns the data frame**

The data frame has no missing value. It has 10 numeric and 10 object variables. Therefore, I will perform encoding these 9 columns***.*** Moreover, I also perform log transformations for skewed numeric variables. ***duration*** is highly skewed and ***pdays*** contain mostly ‘999’, which may present missing value. Therefore, these two variables should not be used for modelling part.

The following is the correlation matrix of variables:

A chart with numbers and a red triangle

Description automatically generated with medium confidence

**Figure 2. Correlation matrix**

The target variables is “y”. From the above correlation matrix, there are many highly correlated features. I will remove them from the list of variables and split it into two sets. The dataset contains no missing values. However, the target variable's scale differs from the other features. Scaling is applied to ensure uniform scales across features, enhancing algorithm performance. The accuracy score evaluates model performance. Before training, SMOTE addresses class imbalance, and the baseline model achieves an accuracy of 0.5, serving as a benchmark for further evaluations

# **Modeling results:**

# Cross validation:

# In this part, I will perform cross validation for Logistic Regression, KNN, Random Forest, Decision tree and Support Vector Machine (SVM). The result is as below:

|  |  |
| --- | --- |
| Baseline model – Accuracy | 0.5 |
| Logistic Regression - Cross-Validation Accuracy | **0.7283 (±0.0051)** |
| k-NN - Cross-Validation Accuracy | **0.8232 (±0.0023)** |
| Random Forest - Cross-Validation Accuracy | **0.9064 (±0.0029)** |
| Decision Tree - Cross-Validation Accuracy | **0.8686 (±0.0043)** |
| SVM - Cross-Validation Accuracy | **0.7436 (±0.0036)** |

# **Table 1. Cross validation**

# In comparison to the baseline models, all the tested models perform significantly better. The k-NN, random forest, and decision tree models exhibit the highest accuracy, with the Random Forest model leading the way at 90.64%. Notably, the Random Forest model demonstrates remarkable consistency in its predictions across various cross-validation sets, underscoring its effectiveness in making precise predictions for this specific problem.

# Next, we will analyze each model for gaining more insights.

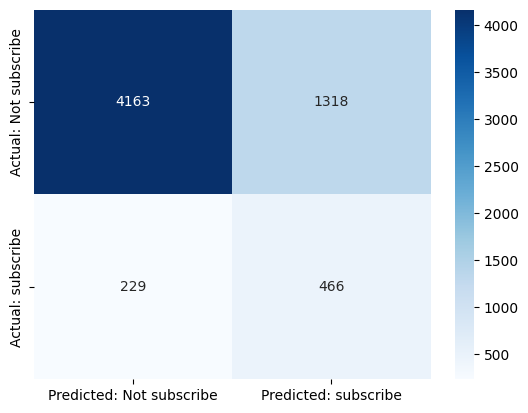
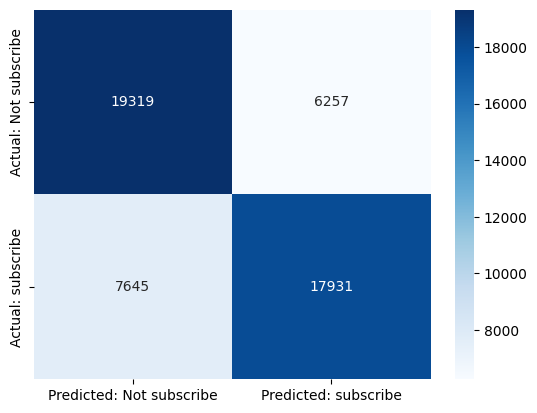
# Logistic regression:

In this part, I will train the logistics model classifier and perform L1 and L2 regularizations. These models are the following. The tables and figures below are the results of this model:

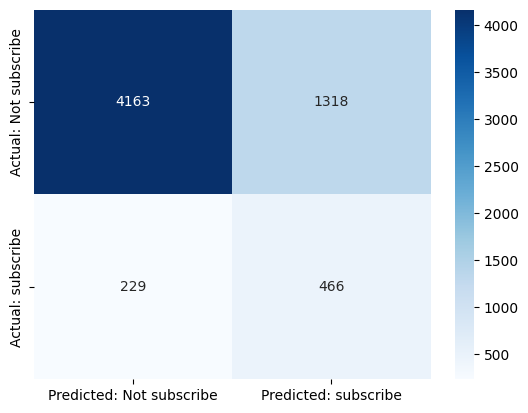
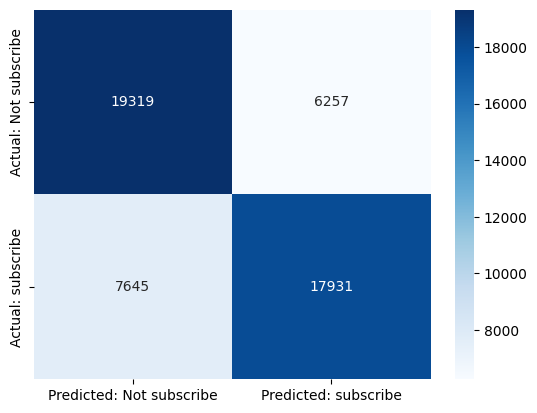
|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy Score | Baseline | Training | Validation |
|  | 0.5 |  |  |
| *Logistic Regression (before regularization)* |  | 0.728 | 0.749 |
| *Logistic Regression (after regularization)* |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Precision, Recall, F1 |  | Training | Validation |
| Precision (before regularization) |  | 0.728 | 0.870 |
| Recall (before regularization) |  | 0.728 | 0.749 |
| F1 (before regularization) |  | 0.728 | 0.790 |
|  |  |  |  |
| Precision (after regularization) |  | 0.728 | 0.870 |
| Recall (after regularization) |  | 0.728 | 0.749 |
| F1 (after regularization) |  | 0.728 | 0.790 |
| Log loss (0.549) |  |  |  |

***Table 2. Logistic regression model result***



**Training set Validation set**



**Training set (after using elastic net) Validation set (after using elastic net)**

**Figure 3. Confusion matrix of logistic regression**

A line graph with blue and orange lines

Description automatically generated

**Figure 4. ROC curve of logistic regression**

1. KNN:

# In this part, I will perform the KNN model for binary classification. After training the model, we will obtain the results as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy Score | Baseline | Training | Validation |
|  | 0.5 |  |  |
| KNN (n\_neighbors=15 and metric: ‘Euclidean') |  | 0.828 | 0.738 |
| KNN (n\_neighbors=55 and metric: ‘Euclidean') |  | 0.773 | 0.737 |
| KNN (n\_neighbors=100 and metric: ‘Euclidean') |  | 0.759 | 0.752 |

|  |  |  |  |
| --- | --- | --- | --- |
| Precision, Recall, F1 | Baseline | Training | Validation |
| KNN (n\_neighbors=15 and metric: ‘Euclidean') |  | 0.833, 0.828, 0.827 | 0.862, 0.824, 0.827 |
| KNN (n\_neighbors=55 and metric: ‘Euclidean') |  | 0.774, 0.773, 0.773 | 0.865, 0.737, 0.781 |
| KNN (n\_neighbors=100 and metric: ‘Euclidean') |  | 0.759, 0.759, 0.759 | 0.869, 0.752, 0.792 |

***Table 3. KNN model results***

|  |  |
| --- | --- |
| **(Confusion matrix on training set,**  **n\_neighbors=15)** | **(Confusion matrix on validation , n\_neighbors=15)** |
| **(Confusion matrix on training set, n\_neighbors=55)** | **(Confusion matrix on validation set, n\_neighbors=55)** |
| **(Confusion matrix on training set, n\_neighbors=100)** | **(Confusion matrix on validation set, n\_neighbors=100)** |

|  |  |
| --- | --- |
| A line graph with blue and orange lines  Description automatically generated  **(ROC curve for k=15)** | **A line graph with blue and orange lines  Description automatically generated**  **(ROC curve for k=100)** |

**Figure 5. KNN confusion matrix and ROC curve**

# Decision tree:

In this part, I will perform the Decision Tree model for binary classification. I will use Bayesian to estimate the hyperparameters for a classification tree. I also adjust parameters to avoid overfitting on the dataset. The results of these models are as below:

***Tree 5 parameters: {random\_state=42, min\_samples\_split=5, max\_depth=4}***

|  |  |  |  |
| --- | --- | --- | --- |
| Accuracy Score | Baseline | Training | Validation |
|  | 0.5 |  |  |
| Decision tree |  | 0.73 | 0.75 |

|  |  |  |  |
| --- | --- | --- | --- |
| Precision, Recall, F1, Best tree |  | Training | Validation |
| Precision |  | 0.733 | 0.876 |
| Recall |  | 0.733 | 0.751 |
| F1 |  | 0.733 | 0.792 |

***Table 4. Decision tree model result***

|  |  |  |
| --- | --- | --- |
| **Training** | **Validation** | **Testing** |
|  |  | A yellow and purple squares with green and blue numbers  Description automatically generated |

# **Figure 6. Decision tree confusion matrix**

# Random forest:

In this part, I will perform the Random Forest model for binary classification. I will use **Bayesian** to estimate the hyperparameters for a random forest. The results of these models are as below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Baseline | Training | Validation |
| Accuracy Score | 0.5 |  |  |
| Random forest with default parameter |  | 0.99 | 0.86 |

|  |  |  |  |
| --- | --- | --- | --- |
| Precision, Recall, F1 for Best Forest |  | Training | Validation |
| Accuracy Score |  | 0.99 | 0.86 |
| Precision |  | 0.98 | 0.87 |
| Recall |  | 0.98 | 0.86 |
| F1 |  | 0.98 | 0.86 |
|  |  |  |  |

**Table 4. Model result of random forest**

|  |  |
| --- | --- |
| Training set (Best parameter) | Validation set (Best parameter) |

# ***Figure 7. Random forest confusion matrix***

# Support vector machine (SVM):

The final model I will train is Support Vector Machine. Before training, I will apply SMOTE algorithm (Synthetic Minority Over-sampling Technique) to balance the class distribution and reduce bias towards the majority class. After using **Bayesian** for estimation, we will obtain the best results with parameter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *{'C': 0.10833608158822433, 'gamma': 0.9668253077390225}* | Baseline | Training | Validation | Testing |
| Accuracy Score | 0.5 |  |  |  |
| Support Vector Machine |  | 0.78 | 0.80 | 0.80 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Precision, Recall, F1 for Best Forest |  | Training | Validation | Testing |
|  |  |  |  |  |
| Precision |  | 0.793 | 0.855 | 0.857 |
| Recall |  | 0.789 | 0.804 | 0.805 |
| F1 |  | 0.789 | 0.825 | 0.826 |

**Table 5. Model result of Support Vector Machine**

|  |  |  |
| --- | --- | --- |
| Training set | Validation set | Testing set |

# ***Figure 8. Support vector machine confusion matrix***

# **Evaluation:**

# Model evaluation and recommendation:

Overall, Random Forest and SVM are the top-performing models with the highest accuracy, strong precision, and recall. It shows the most promising results for making accurate predictions. However, Random Forest seems to be overfitted on validation while SVM shows the consistency among three sets. Decision Tree also performs well on both training and validation set. The higher value of precision, recall and F1 in validation set proves that decision tree model is not overfitted and can be used for prediction.

Logistic Regression shows reasonable performance with an accuracy of 74.9% before and after regularization. It excels in precision but has lower recall, implying it's proficient at identifying non-subscribers but may miss potential customers. The low log loss of 0.5501 indicates the model's effective predictions on the validation data.

On the other hand, k-Nearest Neighbors (KNN) achieved its best performance with n\_neighbors=15, with a validation accuracy of 73.8%. The ROC curve suggests good model discrimination, but the model illustrates the trade-off between precision and recall.

The choice of the model should depend on the specific goals and trade-offs between precision and recall. ***Support Vector Machine (tree 5: min\_samples\_split=5, max\_depth=4),*** due to its high accuracy, predictive power and the consistency performance on both training, testing and validation set, might be the strongest candidate. The below figure is the summary of the performance of above **SVM model**.

|  |  |  |
| --- | --- | --- |
| Training | Validation | Testing |
|  |  |  |
| **Accuracy score=0.789** | **Accuracy score=0.804** | **Accuracy score=0.805** |
|  | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Precision, Recall, F1 for Best Forest |  | Training | Validation | Testing |
|  |  |  |  |  |
| Precision |  | 0.793 | 0.855 | 0.857 |
| Recall |  | 0.789 | 0.804 | 0.805 |
| F1 |  | 0.789 | 0.825 | 0.826 |

# Model analysis (SVM model) and insights pulling:

# A graph with different colored bars Description automatically generated

# Dual Coefficients are pivotal in SVM algorithms, signifying the weight given to each support vector for decision-making.

# The graph above shows SVM dual coefficients. "***euribor3m***" has a strongly negative coefficient, indicating that higher "***euribor3m***" rates are linked to lower campaign subscription likelihood (***matching EDA findings, details in EDA and appendix***). This is likely due to the significance of default and loans. Interest rates can burden people financially, influencing customer behavior (more in EDA analysis). Keep an eye on economic factors like interest rates for marketing strategy adjustments to boost campaign effectiveness.

# 

# The positive dual coefficient for "Education" indicates that higher levels of education positively influence the decision to subscribe. Customers with education levels beyond primary schooling are more likely to subscribe. This suggests that the marketing campaign may resonate more with individuals with higher education, such as university degrees or professional courses. Tailoring marketing messages to appeal to this educated audience segment could lead to higher subscription rates.

# The positive feature importance for the number of contacts implies that increasing the number of contacts may increase the likelihood of subscription. It suggests that as the number of contacts during the campaign increases, the model is more likely to predict a certain outcome. The company should fine-tune the contact strategies to make them more effective. This could involve reaching out to potential subscribers more frequently or at strategic times.

# Ethical consideration

It is recommended that the results of this experiment should not be misused for purposes other than business objectives. It is because the data of this experiment contains sensitive information about customers, including age, gender, and so on.

Finally, monitoring the machine learning model's performance in a routine is crucial to ensure that the model is still accurate and can generate good predictions from unseen data. The regular evaluation also helps identify potential issues, ensuring the model can make ethical and practical recommendations.

# **APPENDICES**

A graph with lines and dots

Description automatically generated with medium confidence

***Average number of CPI, EURIBOR3M, Employee Variation Rate, Subscription Rate,***

***and rate of default monthly***

# **REFERENCES**

Sexton, R. L. (2010). *Exploring Economics* *(5th ed.).* Mason, OH: South Western Educational Publishing.