ITNPBD6: Wallace Communications - Student ID: 2710017

Project Methodology – CRISP [1]

Business Understanding

- From briefing paper: need to identify customers likely to take a mobile contract. A binary classification exercise with Y/N target
- High call costs, suggest need to accurately identify customers but avoid 'False Positives'

Data Understanding

- Tools: VSCode, Jupyter Notebook, Python, Pandas, Scikit-learn [4]
- Loaded the CSV file. Data exploration & visualisation
- Identify data issues, eg: missing, duplicates, outliers, unbalanced

Data Preparation

- Refined over several iterations of Prep and Modelling (see variables)
- Data split 80:20 using `train_test_split` to: X_train, X_test, y_train, y_test (cross-fold validation to be used on train)
- Repeatable data transformation using `Pipeline` & `ColumnTransformer`, fitted to train data then applied to separate train and test
- Categorical variables encoded using `OneHotEncoder`
- Numerical variables scaled using `RobustScaler`

Modelling

- Selected 3 candidate model types
- Model training and hyper parameter tuning (see below)
- The best trained model for each type saved for evaluation

Evaluation

- Transformations applied to the reserved test data
- Each model used to make and evaluate predictions (see below)

Deployment

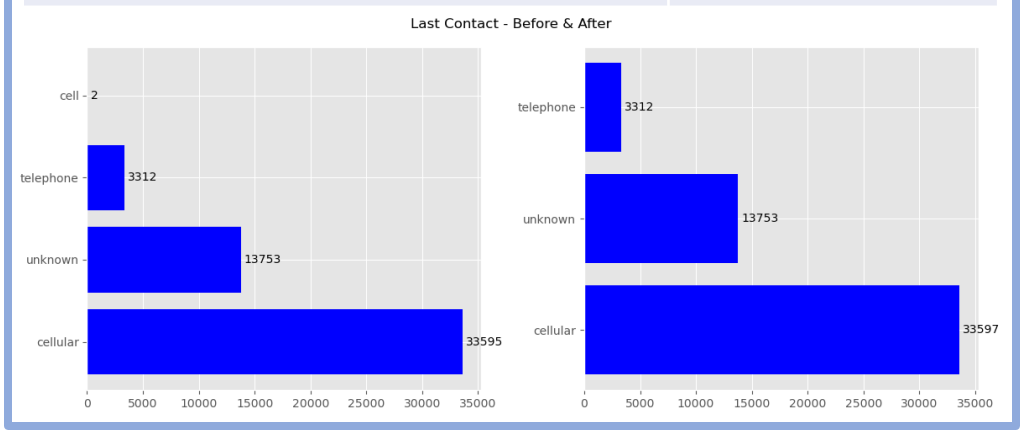
- The data transformation pipeline, selected model type and best hyperparameters were used to retrain the model on ALL data
- This model was saved using Pickle for use on new data

Variables & Data Preparation

- 20 original variables -> 9 dropped leaving 10 features and 1 target
- 50,662 original data items -> 50,660 retained (see notes on outliers and balancing)

- Dropped of note: ID (flat & wide); Town (101 categories & no significant contribution to modelling); Country (Towns all UK, despite 5 other countries recorded)
- Columns dropped only after iteratively modelling and identifying their contribution
- Unbalanced & outliers: CurrentBalance, ThisCampaign
- Unbalanced target: NewContractThisCampaign
- Minor cleaning: HasTVPackage, LastContact (2 occurrences of 'cell' -> 'cellular')

Variable	Type
Age, ThisCampaign	Numeric - Discrete
CurrentBalance	Numeric - Continuous
Job, Married, Education, Housing, HasTVPackage, LastContact, OutcomePreviousCampaign	Categorical
NewContractThisCampaign	Target - Categorical



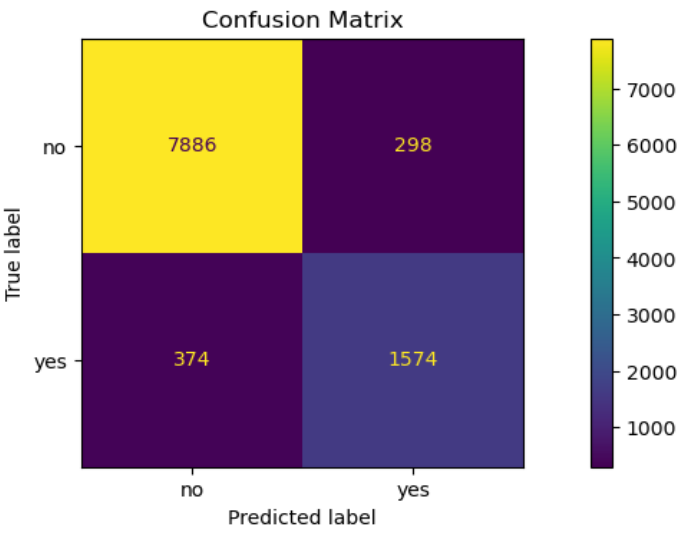
Model Training and Hyper Parameters

- Pipelines created for each model type with a grid of several types of hyper parameters, each with several values
- A scoring metric of `Precision` seemed to best fit with the business objectives, but `F1-Score` and `ROC-AUC` were also used for comparisons
- Multiple model training & evaluation iterations were then completed. Each using cross-fold validation, `GridSearchCV`, CV=5:
  - i) hyper parameter types & values changed; 2) fitted against train data 3) cross-validation completed; 3) inspected top 10 ranked by the scoring metric
- The best combinations of hyperparameter types and their values were identified, `best_estimator_` saved and used to evaluate across the 3 models

Model	Hyper Parameters (Best Combinations)	Validation Metric (Best Score + Elapsed Secs)
Logistic Regression	Solver: saga, Penalty: l2, C: 5, ClassWeight: balanced, MaxIter: 500	Precision (0.35, 14)
Random Forest	Criterion: entropy, nEstimators: 100, MaxDepth: 50, ClassWeight: balanced,	Precision (0.81, 64)
MLPClassifier	HiddenLayerSizes: (25,5), Activation: tanh, alpha: 0.1, MaxIter: 500	Precision (0.36, 17)

Model Evaluation: Random Forest

- Training results: the Random Forest model performs significantly better than the other two
- Business perspective: model evaluation needs to consider how comprehensively clients for marketing are predicted, but without including too many that are not interested (as this is a waste of expensive call centre time). The confusion matrix helps with this
- Ideal quadrants: True/Positives (bottom right) are high but False/Positives (top right) are low. Also, too many False/Negatives (bottom left) mean that potential customers are getting missed
- Specific metrics: high *Precision* whilst maintaining a high *Recall*. The F1-Score helps show both
- **The Random Forest model gave good results: *Precision*: 0.84, *Recall*: 0.81 and *F1-Score* 0.82**



Final Comments

- Feature importance: 73% of the performance comes from just 4 features: CurrentBalance, Age, ThisCampaign (all numeric) plus OutcomePreviousCampaign.
- Unbalanced & Outliers: For several features plus the target. Early iterations attempted to address these (eg using SMOTE [2] and repeatable outlier removal > STDev*5). But, this surprisingly did not improve model performance so not utilised
- However, the use of `RobustScaler` and a balanced `ClassWeight` had a better impact whilst also avoiding the loss of data points

References

[1] P. Chapman *et al.*, 'CRISP-DM 1.0', 1999.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, 'SMOTE: synthetic minority over-sampling technique', *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[3] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*, 3rd Edition. O'Reilly Media, Inc., 2022.

[4] F. Pedregosa *et al.*, 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.