ITNPBD6 Assignment - Store Performance Classification

UNIVERSITY of STIRLING

In the highly competitive retail industry, the owner of a chain store considered implementing a machine learning model to predict whether upcoming stores will perform well or poorly. By analysing key performance factors, a proposal is designed to assist the organisation in making data-driven decisions to optimise and strategize the performance of existing and upcoming stores.

1. Project Methodology

1. Imported Pandas, NumPy, Matplotlib, Seaborn, and sci-kit-learn libraries.
2. Loaded the data into a Pandas dataframe as df and created a copy – df_copy.
3. Divided the data in a 70-30 ratio between train and test.

On the training set –

4. Removed duplicates & null values and dropped the columns 'Manager Name' & 'Town', '10min population', '20min population' & 'Country'.
5. Identified categorical, numerical – discrete and continuous values.
6. In categorical data, the 'Car Park' values were transformed to Yes and No, and outliers were removed, such as the value 'Village ' the in 'Location' column. Then, 'Car park' and 'Location' are one hot encoded, and the 'Performance' column has been label-encoded.
7. Removed outliers and hot-encoded 'Staff' and 'Store age' columns in Discrete data.
8. Checked for outliers and skewness in continuous data, transformed skewed data to Normal Distribution [2], and Checked for Zero Values.
9. Scaled all the Continuous variables and remaining Discrete variables (Window, Demographic score, Competition number & score) using MinMaxScaler [1].
10. The training data has been separated into X_train and y_train.
11. Selected the classifier models, specified hyperparameters, and fit the model to training data and trained using different hyperparameter values. And evaluated the cross-validation scores of all models on training data.
12. Then, chose the final model, prepared test data using the steps outlined above (4 to 9), and performance & scoring metrics have been assessed on the final model.

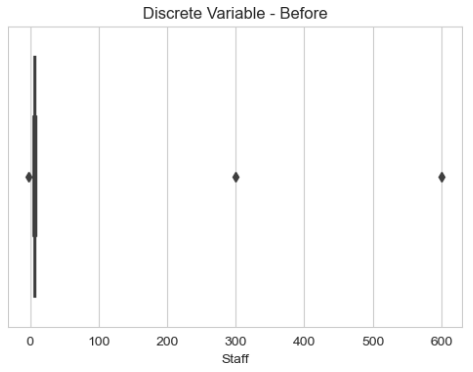
2. Variables

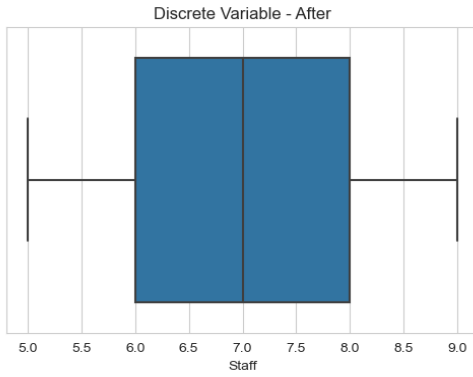
Type	Variables
Categorical	Car park, Location, Performance
Continuous	Store ID, Floor Space, 40min population, 30min population, Clearance space
Discrete	Staff, Window, Demographic score, Competition number, Competition score, Store age

- **The categorical feature** variables are hot encoded; however, the target variable "Performance" has been label-encoded as hot encoding may make the model more complex with more possible combinations, eventually leading to lower performance.
- **Columns '10min population,' '20min population,' and 'Country'** have been removed as they do not correlate with the target variable.
- Given the **large sample size**, the discrete variables 'Competition number', 'Competition score' and 'Window' were treated as continuous and scaled for normalisation while others were hot encoded.
- **All continuous variables** are normalised and scaled.
- Since there are only **a few continuous variables** and the dataset is small**, Principal Component Analysis has not been applied** to the train or test data.

3. Data Preparation

The data cleaning and pre-processing stages are outlined in the Project Methodology(steps 4–9), and a sample instance is provided below.



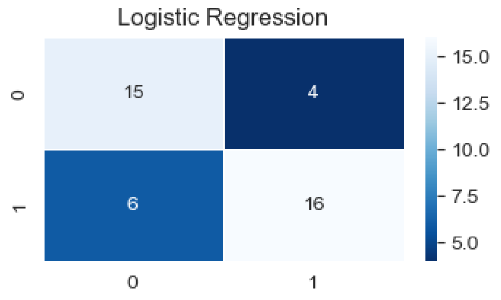


4. Model Training and Hyper Parameters

Model	Hyper Parameters(with CV fold=5)	CV Score(%)
RandomForest - 1	n_estimators=40, max_depth=20, min_samples_split=30, max_features='sqrt'	71
RandomForest - 2	n_estimators=50, max_depth=10, min_samples_split=30, max_features='log2'	73
RandomForest - 3	n_estimators=100, max_depth=20, min_samples_split=30, max_features='log2'	76
AdaBoost	(RandomForestClassifier(), learning_rate=0.01, n_estimators=500)	73
LogisticRegression-1	**(C=1, penalty='l1', solver='liblinear')**	**84**
LogisticRegression-2	(C=1, solver='liblinear', penalty='l2')	81
LogisticRegression-3	(C=1, solver='newton-cg', penalty='l2')	81
MLPClassifier-1	(activation='tanh', hidden_layer_sizes=700, max_iter=1533, solver='adam')	81
MLPClassifier-2	(activation='relu', hidden_layer_sizes=70, max_iter=941, solver='sgd')	80
MLPClassifier-3	(activation='tanh', hidden_layer_sizes=800, max_iter=2888, solver='adam')	84
StackingClassifier, referred [5]	(MLPClassifier(activation='tanh',hidden_layer_sizes=99,max_iter=941, solver='sgd')), (RandomForestClassifier(max_depth=10, max_features='sqrt', min_samples_split=15, n_estimators=20)), (LogisticRegression(C=1, penalty='l1',solver='liblinear')), final_estimator=LogisticRegression()	84
****CV – Cross Validation**		

The hyperparameters required to tune each model have been chosen from [3] and have been tuned using RandomizedSearchCV and by manually tuning made slight adjustments to get the improved results using cross-validation with 5 folds. In the final model of Logistic Regression, C is kept at 1 to increase the strength of regularization, the penalty term l1, and the solver is liblinear, which is found to be suitable for small datasets [3].

5. Final Model and Results



The training set was initially divided into X_train(feature) and y_train(target), after which the aforementioned models were defined, hyperparameters were chosen using RandomizedSearchCV and manual tuning, the models were fitted to training data and trained using a distinct set of hyperparameters and were later evaluated using 5-fold cross-validation. Following the analysis of the cross-validation scores, the best final model was chosen. To assess the test data, the test data was initially cleaned and processed before being split into X_test(feature) and y_test(target). The final model's predictions on X_test were later evaluated in the confusion matrix, and evaluation metrics such as accuracy, precision, and F1 score were analysed. According to the confusion matrix, the model **has an accuracy of 0.76, precision of 0.80, recall of 0.72, and F1 score of 0.76**, indicating that it performs fairly well on test data; however, False Negatives are slightly higher than False Positives.

The Logistic Regression, MLP & Stacking Classifier models consistently produced the highest overall scores, whereas the Random Forest showed high variance. And even though the MLP or Stacking Classifier provided identical scores as the Logistic Regression model while accounting for other factors, such as model complexity & computational costs, **the Logistic Regression with a Cross-Validation Score of 84% was chosen as the final model**. Given that we have less complex data with fewer features, a comparatively low complex Logistic model (which is trained by estimating the log-odds of the target variable being in a given category) is more efficient than an MLP(considering the number of layers, neurons & interconnections) or Stacking model(which involves multiple models), as it would help in **less cost of running and maintenance, being more efficient, being deployed faster and more scalable for real-time applications in the long run** [4]. Additionally, it may have **less chance of overfitting the data and is more interpretable** [4].

6. References

[1] https://vitalflux.com/minmaxscaler-standardscaler-python-examples/
[2] https://medium.com/analytics-vidhya/techniques-to-transform-data-distribution-565a4d0f2da
[3] https://scikit-learn.org/stable/modules/classes.html
[4] https://towardsdatascience.com/model-complexity-accuracy-and-interpretability-59888e69ab3d
[5] https://vitalflux.com/stacking-classifier-sklearn-python-example/