Store Performance Classification

ITNPBD6 Assignment 1. Student Number 3142459

1.Introduction

The aim is to build classification model to classify a store' performance good, or bad using data about stores in the UK.

2. Project Methodology

In particular, we will go through CRISP-DM as:

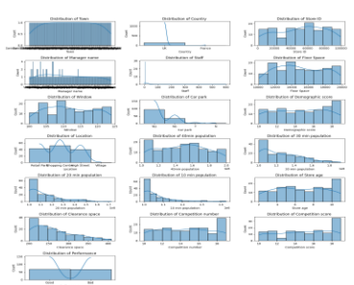
- Environment setup import required libraries
- Preparing Data
- Splitting Dataset into Training Set and Test set
- Training Model
- Evaluating Model using Confusion Matrix

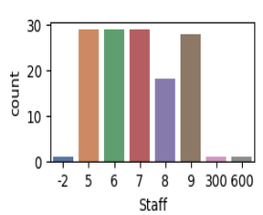
3. Preparing Data

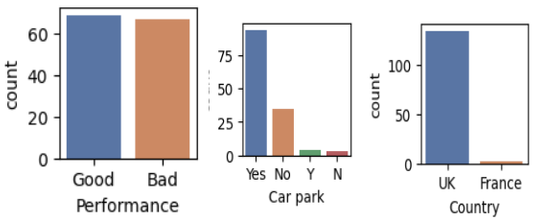
There are136 examples and 19 variables.

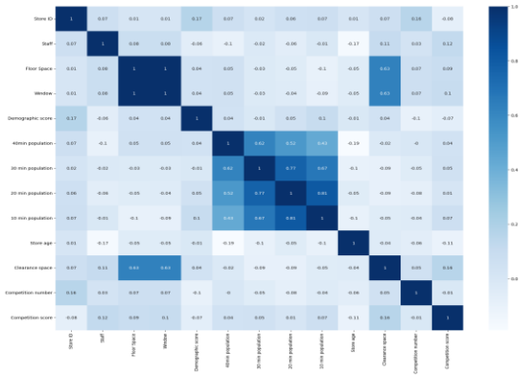
Variable Type	
Categorical	Nominal: Town, Country, Car park, Location, Manager name Ordinal : Performance
Numerical	Discrete: Store ID, Staff, Competition score, Competition n umber, Store age, Window, Demographic score, 40min population, 30 min population, 20 min population, 10 min population Continuous :Floor Space,, Clearance space

- There is no missing values.
- Performance as either a good or bad . It is a binary classification problem (balanced).
- Car park has inconsistency.
- Country has wrong data entry(France)
- Staff has outliers, replace them with median.



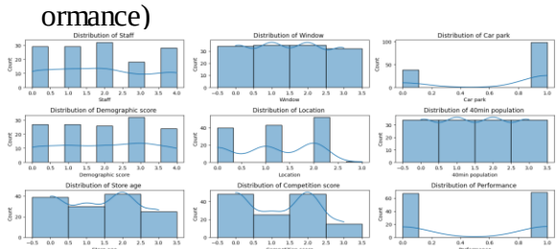






In the heatmap , there is high correlation like between Floor Space and Window as in the dark color. Competition number has strong negative correction with 40min population.

- Convert categorical variables ["Car park", "Location", "Performance"] to numerical using one hot encoding in preprocessing module in Sklearn
- classifying Staff, Store age, Competition score, 40 min population , Demographic score and Window like in the histogram below.
- Drop the variables : Town, Country, Store I D, each have unique value. Floor Space, , Competition numbe,30min, 20 min and10 min population , as these are correlated. Manager name has no impact.
- Split the data into an input X of 8 features and output y 1 binary target variable(Perf ormance)



4.Splitting Dataset into Training Set and Test set

The split ratio 70:30

5.Training Model

Training Logistic regression with respect to type of solver, ect to find best combination of parameter using GridSearchCV.This approach searches for the best set of hyperparameters from a grid of parameters

```
param_grid = [
    {'penalty' : ['l1', 'l2', 'elasticnet', 'none'],
    'C' : np.logspace(-4, 4, 20),
    'solver' : ['lbfgs','newton-cg','liblinear','sag','saga'],
    'max_iter' : [10, 20,30, 100]
    }
]
```

```
Tuned Logistic Regression Parameters: {'C': 0.23357214690901212, 'max_iter': 30, 'penalty': 'l1', 'solver': 'saga'}
Best score is 0.7894736842105263
```

Decision Tree consider the function to measure the quality of a split. The maximum depth of the tree. Number of features to consider, ect. RandomizedSearchCV approach is applied.

```
# Creating the hyperparameter grid
param_dist_dt = {"criterion": ["gini", "entropy"],
                 "max_depth": [3, 5, None],
                 "max_features": randint(1, 6),
                 "min_samples_leaf": randint(1, 6)
                 }
```

```
Tuned Decision Tree Parameters: {'criterion': 'gini', 'max_depth': None, 'max_features': 5, 'min_samples_leaf': 5}
Best score is 0.6631205673758864
```

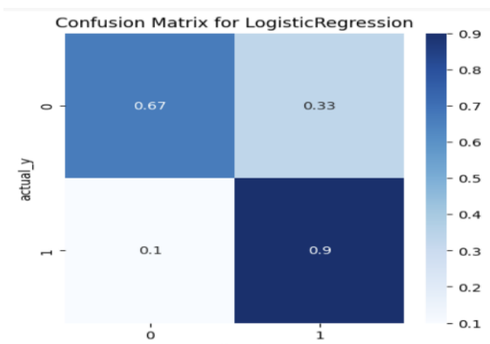
Neural Network

```
params_NN={'batch_size':[100, 20, 50, 25, 32],
           'nb_epoch':[1, 20, 10, 30, 40],
           'unit':[6,4,3],

           }
```

```
Tuned Neural Network Parameters: {'batch_size': 25, 'nb_epoch': 40, 'unit': 3}
Best score is 0.5894736826419831
```

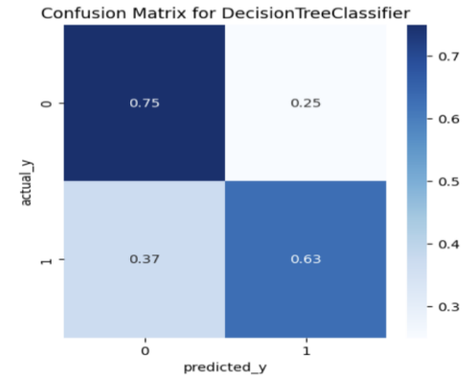
6. Evaluating Model

Confusion matrix to evaluate accuracy classification model.

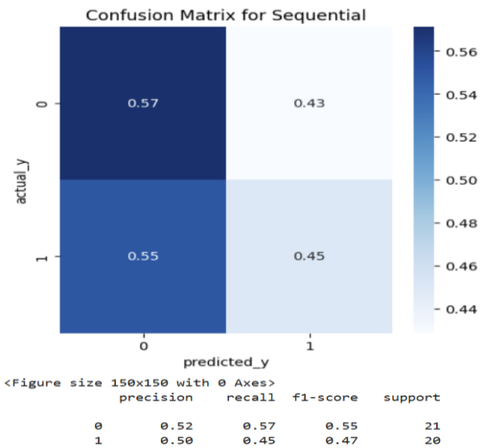



```
<Figure size 150x150 with 0 Axes>
              precision    recall  f1-score   support

           0       0.88      0.67      0.76        21
           1       0.72      0.90      0.80        20

    accuracy                           0.78        41
   macro avg       0.80      0.78      0.78        41
weighted avg       0.80      0.78      0.78        41
```

	Train Accuracy	Test Accuracy
Logistic Regression	0.82105	0.78049

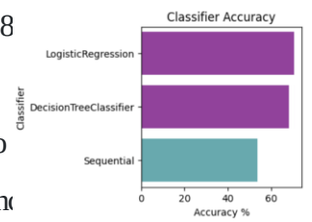



```
<Figure size 150x150 with 0 Axes>
              precision    recall  f1-score   support

           0       0.68      0.75      0.71        28
           1       0.71      0.63      0.67        27

    accuracy                           0.69        55
   macro avg       0.69      0.69      0.69        55
weighted avg       0.69      0.69      0.69        55
```



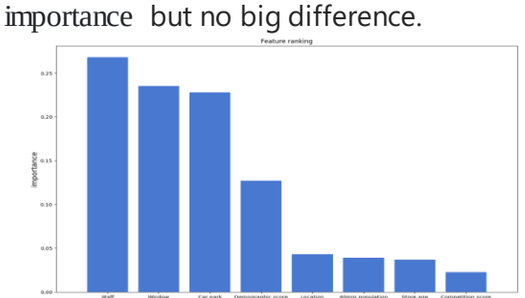

```
<Figure size 150x150 with 0 Axes>
              precision    recall  f1-score   support

           0       0.52      0.57      0.55        21
           1       0.50      0.45      0.47        20

    accuracy                           0.51        41
   macro avg       0.51      0.51      0.51        41
weighted avg       0.51      0.51      0.51        41
```

The model accuracy .78,.68 and .51. The best outcome is for logistic regression to classify the store performanc



with .9 correctly predicted as good (1 class) out of actual, .67 to predict the bad performance class. The F-score is .80 which is good.

Notice: accuracy for training > accuracy for test; possibly due to small sample/many features (valliage store and without parking are not represent well, .also tried models with different features selected (ranked according to score for importance but no big difference.



7. References

1.Logistic Regression Classifier Tutorial | Kaggle

2. Scikit-learn DecisionTreeClassifier

3 tf.keras.Model | TensorFlow v2.12.0

4. Hyperparameter tuning - GeeksforGeeks