1. Student Number: 3068000

STORE PERFORMANCE CLASSIFICATION MODELS

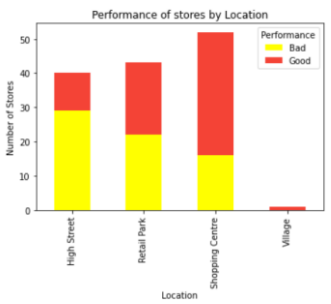
2. Project Methodology:

- Data Cleaning & Pre-processing
- Feature Engineering
- Splitting data into train/test/validation sets
- Model training and testing
- Model evaluation

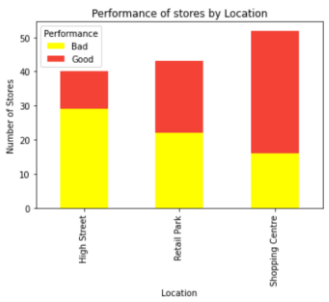
3. Data Cleaning & Pre-processing

When I check the performance of store in each location, I observe that our data has an extra location input 'Village'. Hence, I remove the rows that have 'Village' as their location before moving on to build our model.

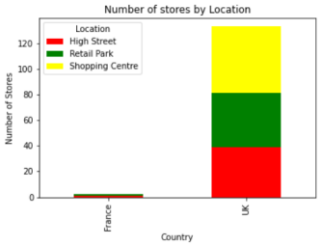
Before:



After:



I further observe through these scatter plots (showing the 'Staff' in each store of every location) that two locations have either outlier values (300 & 600) and one location has an unrealistic value of -2 as the number of staff, which can't be true. Hence, I remove these rows to further clean our data.



Moreover, there is an extra country in our dataset. As our model is for the stores in UK only, hence we remove the rows that have 'France' as a value in the country column.

I have removed the outliers and extra values, now we shortlist the columns that we need for our model building.
I further drop the 'Town', 'Country', 'Store ID' & 'Manager name' columns as they are not an important factor in determining the performance of a store hence, they are not needed in our performance classification model.

4. Feature Engineering

```
df.nunique()

Town                    130
Country                   1
Store ID                130
Manager name            116
Staff                     5
Floor Space             129
Window                   25
Car park                  4
Demographic score        10
Location                  3
40min population        130
30 min population       130
20 min population       130
10 min population       130
Store age                10
Clearance space          88
Competition number       10
Competition score        10
Performance               2
dtype: int64
```

'Car Park': When I check the unique values for each column in our dataset after data cleaning, I observe the column of 'Car Park' has 4 distinct values, when it should only have 2 (Yes or No). Hence, I dig further and see that the values are either Y, Yes, No or N. I change the column first to True/False values and then convert into 0 and 1 values.

'Location': All our independent variables are now in either numerical or numerical boolean form, other then 'Location' column. The 'Performance' variable is treated as categorical. I create dummy variables for 'Location' categorical column. The resulting data frame looks like this:

Staff	Floor Space	Window	Demographic score	40min population	30 min population	20 min population	10 min population	Store age	Clearance space	Competition number	Competition score	Loc_High Street	Loc_Retail Park	Loc_Shopping Centre	CP_bol	Performance
9	18526	121	10	1288374	1138224	1006986	1002340	3	238	16	16	0	1	0	1	Good
8	18569	121	11	1086225	1015321	1012182	1008436	4	384	15	19	0	0	1	1	Good
7	17092	117	14	1179395	1022959	1009496	1002169	5	261	15	12	1	0	0	0	Bad
7	11307	103	18	1398558	1085170	1003137	1002513	7	200	19	13	0	1	0	0	Bad
7	17888	119	19	1614716	1325848	1220059	1193318	2	394	17	11	0	1	0	1	Good

One consequence:
Creating dummy variables for 'Location' column would allow us to incorporate it in our model building. It also allows us to have all the variables included in x to be in the numerical form in one way or another.

5. Splitting data frame

Initially, I split the data into x and y values. With 'Performance' as our y variable and all other as out x variable. Further, I have split the data in x and y variable into 70/15/15 ratio, with 70% being used for training our models, 15% for the purpose of validation and 15% has been reserved for the purpose of testing our model.

5.1 Model training and testing

Model	Hyperparameters	Metric
Decision Tree	Criterion = 'gini' max_depth = 4	Confusion Matrix, Accuracy Score
Logistic Regression	Penalty = l2	Accuracy Score
Neural Networks	input_shape=(X_train.shape[1],), 32, dropout=0.2, 16, dropout=0.2, 'sigmoid', 'adam', 50 epochs, batch_size=32	Accuracy Score, Confusion Matrix

A sentence for the chosen hyperparameters:
Decision Tree: 'entropy' criterion with max_depth combinations resulted in a lower accuracy so gini criterion was used with max_depth which depicted higher accuracy.
Logistic Regression: Basic logistic regression and regularization of l1 penalty resulted in a lower accuracy. Hence, l2 was applied which increased the model accuracy.
Neural Networks: Adding the dropout layers increased the model accuracy which couldn't be achieved by using different activation methods or increasing the number of hidden layers in the neural network model. The accuracy on our test set came out to be 0.699 which can be interpreted as almost 70% accuracy.

6. Model Evaluation

The **final model** chosen is 'Neural Network' model with dropout layers. It works efficiently for the task of classification upon our training set after being trained and validated. Also, it provides us with an accuracy of almost 70% upon using the test set, which could not be achieved while testing various decision tree or logistic regression models with different parameters. Dropout is a regularization technique that can be used to reduce overfitting in neural networks. The idea is to randomly drop out (i.e., set to zero) some fraction of the neurons in each hidden layer during training. This helps to prevent the network from relying too much on any one neuron or set of neurons and encourages the network to learn more robust and generalizable features.

When we test our model on the test data split, we get the following confusion matrix:

Actual/Predicted	Good	Bad
Good	5	4
Bad	2	9

The confusion matrix generated indicates that our model made predictions for a total of 20 samples in the test set. Out of these, 5 were true negatives (TN), 4 were false positives (FP), 2 were false negatives (FN), and 9 were true positives (TP). This depicts a 70% accuracy of our model which means that if this model was used for store performance classification, 70% of the stores would be correctly predicted.

7. Reference:

For knowing the working of hidden layers in neural network model: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/