# Team PredictiveX (DataStorm036)

Team Members

Isuru Gunarathne
Supun Gamlath

GitHub: https://github.com/IsuruGunarathne/datastorm-4

Highest Total F1 score: 0.70833

# Approach

We used a relatively simple approach (approach 1) to this problem as the data set was small. Even though we had about 500,000 transactions, we decided it would be best to extract per-store data from those transactions and use the 100 labelled entries provided as training data for a model that would predict the shop_profile for a given set of features derived from the transactions of a given store.

There was some confusion as to whether our approach was correct, the reason being the following statement

*"The initial solution they developed was to classify the existing outlets as High, Moderate, and Low based on a salesperson's point of view. However, they now seek an advanced analytics solution to automate this process for the new stores."*

If the classification provided in the 100 entries was based on a salesperson's POV, then using them to train the model would only make it behave as a salesperson classifying the stores. In this case, the correct approach would have been to tackle this as an unsupervised learning problem, where the model learns common patterns and clusters similar shops into 3 categories. (for simplicity we will call this 'approach 2')

The reason we abandoned approach 2 was;

1) Our first submission using approach 1 got a relatively high F1 score of 0.6666
2) The dataset didn't have enough entries to make unsupervised learning feasible.
3) This was mostly an assumption. It is possible that for evaluation the competition used classification based on the '*salesperson's point of view'.* In this case, even a perfect model would score relatively badly due to the presence of bias in the classification used for scoring.

After concluding that approach 1 was the better approach we proceeded with the following tools.

## Tools used

Pandas - for handling data (CSV/data frames)
NumPy - for processing data frames and calculations
Seaborn - for data visualisation (graphs, heatmaps etc.)
Xgboost - The model used
sklearn - for feature engineering and overall machine learning

We decided to use Kaggle's built-in notebooks (connected to GitHub as the VCS) for all our work as it provided easy access to most libraries and dependencies we would be needing without much hassle.

# Feature Engineering

## Features extracted from transactions

- Shop_area_sq_ft - provided along with store classification
- Total_vol - total sales volume per store
- Vol_per_sq_ft - an evaluation metric used generally to rank store performance (total sales volume/shop area)
- Total_customers -  total number of unique customer_id s that visited a shop
- Total_unique_items - total number of unique items the store sold
- Single_day_max_vol - maximum volume for a single day for a given store
- Single_day_min_vol - minimum volume for a single day for a given store
- N_transactions - total number of transactions for a given store
- Customer_id_unique - same as total_customers
- Customer_id_repeating - number of customers that visited the store more than once
- Customer_id_single - number of customers that visited the store only once

## Feature selection

Starting off the first submission we made was solely based on the three features, shop_area_sq_ft, total_vol and total_customers. These were based purely on intuition and we used them as a starting point for all other feature ideas.

For subsequent models, we used techniques such as correlation heatmaps, MI scores and covariance as metrics to select features

As we added more features we saw an increase in F1 scores during cross-validation, but upon submission, we saw relatively low F1 scores, this led to the conclusion that adding more and more features given the limited number of training entries would only result in the model overfitting the training data.

Therefore for the final few submissions we used models that had only 2 to 4 features

# Final Model

The final model and initial model both performed equally upon submission, resulting in F1 scores of 0.6666. The final model used only shop_area_sq_ft and total_vol, we eliminated total_customers due to the high correlation between total_vol and total_customers.

The final model had a cross-validation F1 score of 0.52

As the deadline was extended we had 2 tactics in mind to increase our score, one was to replace total_vol with total_customers as total_customers had a higher covariance with the outcome.

The second tactic, the one we ended up using was, observing our previous predictions and manually checking for differences and similarities (this was only possible as there were only 24 predictions). Upon inspection we saw that there were only 6 entries that were different between the 2 best submissions, we then proceeded to use the 2 additional submissions we got to change one of those 6 and see how it affects the score, if the score decreased, the conclusion would be that the change we did was wrong and doing the opposite change on the other submission would increase our score.

Based on this we changed the profile of shop 97 to moderate (it was previously categorised as low)

Upon submission, this increased the score to 7.0833, at which point we decided to stop as a further modification would be pure guesswork and would deviate us from the objective of the competition.

# Business Insights

The Business insights we can derive from this task are quite limited as it’s simply a task of classifying stores

A recommendation we can provide is a model to classify users based on their behaviour, This would be an unsupervised learning task but would be feasible based on the volume of transaction data available.

This could be useful for targeted marketing of selected offers, this could help in customer retention in the long run.