# Data Storm 3.0

## Technical Report

| | |
|---|---|
| Team Name | : Golems |
| Kaggle Username | : ds22-63 |
| Kaggle Display Name | : ds22-63 |

Team Member 1 : Kavindu Gunathilake
Team Member 2 : Nadun Siriwardhana
Team Member 3 : Sadesh Surendra
( IIT 3rd Year Software Engineering Undergraduates )

MAPE Scores
Public - 47.22152
Private - 61.88179

Github repository : https://github.com/nadunys/data-storm-3.0

# 1. Technical Report

## 1.1 Features

Data were loaded from the given CSV files and checked for the columns.

| | CategoryCode | ItemCode | DateID | DailySales |
|---|---|---|---|---|
| 0 | category_2 | 117610 | 11/6/2021 | 7 |
| 1 | category_4 | 836584 | 11/18/2021 | 16 |
| 2 | category_1 | 370195 | 1/24/2022 | 6 |
| 3 | category_2 | 172582 | 10/30/2021 | 5 |
| 4 | category_2 | 1006009 | 10/30/2021 | 5 |

A hypothesis was made to decide how to proceed in next steps.

- Create a numerical mapping for category codes
- Create a numerical mapping for Item codes
- Use week number of the year instead of the date given
- Group sales data to the respective week

According to those points, the columns were modified.

## 1.2 Feature Engineering Steps

| | WeekID | CategoryCode | ItemCode | Year | WeeklySales |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 2022 | 83 |
| 1 | 1 | 0 | 1 | 2022 | 66 |
| 2 | 1 | 0 | 5 | 2022 | 21 |
| 3 | 1 | 0 | 6 | 2022 | 621 |
| 4 | 1 | 0 | 10 | 2022 | 31 |

Each category code and item code were mapped in to a number in order to keep every feature numerical. As well as, the date column was dropped from the dataset and created a new column with week id which the date is belong to. The data were grouped using week id. Even though the month and year can be crucial to the prediction, due to lack of data distribution, they were decided to drop from the dataframe. The seasonal trends will still be counted into using week IDs.

## 1.3 Modeling approaches

Initially, a thorough research was done to figure out what kind of algorithm to use. The decision was to use ensemble learning algorithms for this problem. ( Both linier regressors and support vercor regressors tend to give lower accuracies )
From training dataset:

- ⅔ of data were used to train the model
- ⅓ of data were used as the test dataset
- Given validation dataset used only for model validation purposes

Data normilization step was run prior to training models using sklearn.preprocessing.Normailzer().

Selected features to train the model as follows,

- Normailzed week id
- Normalized item code
- Normailzed category code

The modeling stepes were initially run through following regression learning algorithms

- Random forest regressor
- Extra trees regressor
- Voting regressor

Since the evaluation scores are almost similar in all three of the algorithms extra optimization step was needed to be done.

Transformers were used to transform targets 'y' befor fitting regression models. The predictions were mapped back to the original space via an invers transform. As well as another preprocessor called QuantileTransformer was used to transform features to follow a uniform or a normal distribution.

| | Model | Score |
|---|---|---|
| 0 | GradientBoostingRegressor | 0.367665 |
| 1 | VotingRegressor | 0.808610 |
| 2 | RandomForestRegressor | 0.807084 |
| 3 | ExtraTreesRegressor | 0.585976 |
| 4 | HistGradientBoostingRegressor | 0.759186 |
| 5 | AdaBoostRegressor | 0.181599 |
| 6 | BaggingRegressor | 0.784980 |

With transformers several algorithms were tried out. The algorithms and the respective explained variance scores are shown in the above figure.
As the most effective regressor, voting regressor with two random forests was chosen and implemented the model.
The fianl model was given a explained variance score of 0.71 for the validation dataset.

As Evaluation metrics, **Uniform average score** and **explained variance score** were used.

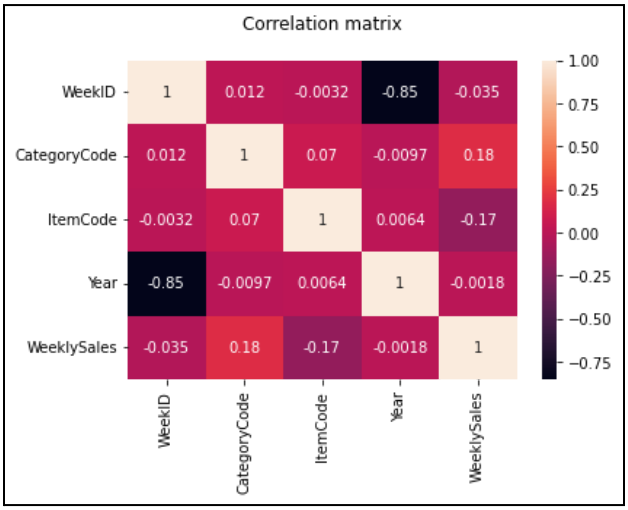
Jupyter notebooks were implemented using Visual Studio Code editor, with the help of following libraries,

- Scikit-learn

- Numpy
- Matplotlib
- Pandas
- Seaborn

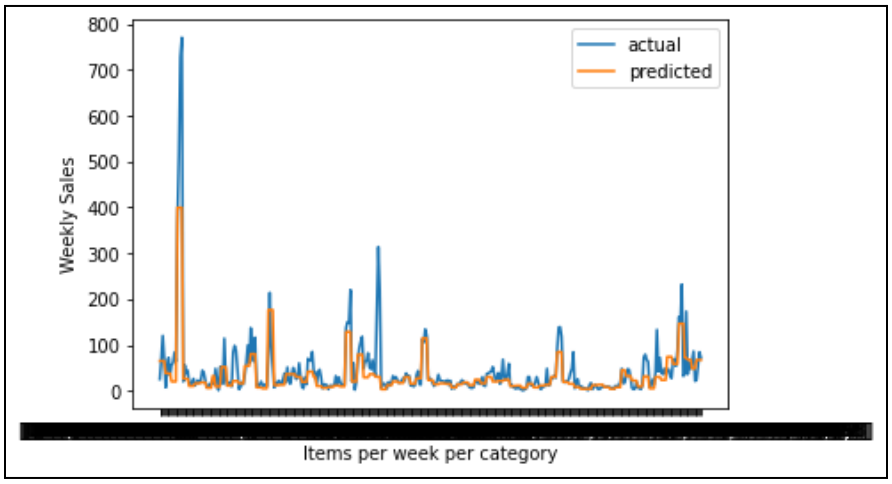
## 1.4 Plots and figures

Initially plotted all columns in the dataset to get a rough idea. Then used a correlation matrix to check the correlation coefficients between variables.




As the above correlation matrix week id, item code and year variables' relationships between weekly sales are weak. Those three variables are labels in the dataset, that is the reason for weak relationship therefore correlation matrix numeric values arn't valuable for this research.

Finally plotted the actual weekly sales values vs the predicted weekly sales values using the voting regressor with two random forests algorithms.



The graph shows the weekly sales for the items per week per category. Both actual and predicted values are plotted in the same graph therefore the accuracy of the model can be checked.

# 2. Interesting business findings

- Seasonal ternds
- Brands or categories that consistently get selled
- Sales growth

# 3. Additional attributes that the retail management should start collecting

## 3.1 Product price

For each inventory item defined with the item code and category, the inventory item price will be useful to gain an understanding of the sales of the selected product. *I.e.*, products in the same category, but sales records may vary with different item codes and different prices.
E.g.: For retail management, after purchasing the same inventory stock for items "B" and "A" in the "Soap" category at different retail prices, found that there are different inventory stocks available at the end of the selected period. We can therefore assume that sales depend on the retail price of the product.

## 3.2 Details of daily sales hours

At the moment the training dataset shows details of daily sales for each retail item. But depending on the situation, sales and demand may fluctuate throughout the day with time. Therefore, management can make decisions about stock availability and manage store congestion by predicting sales demand times.

## 3.3 Consistent data set across the year

This training dataset distribution is very limited. The CSV file does not contain values for some of the months. So, having a consistent data set will be help full to predicting the behavior of sales in each season. *I.e.* The Christmas and summer seasons have more sales than any other month of the year. This information form will be used to validate when we forecast sales for a specific month or period.

## 3.4 Geometric distribution of sales

# 4. Interventions that the management team can take as the next steps

The management team should take action to collect more data ( at least 2 years of data ) before using this model in actual business purposes. Then re evaluate the model and they should create new model using collected data.
The model will give the predictions up to some confidence level. But the management should validate and cross check those predicted values with the normal process for some time period, before fully automating the system.
After automating the process, the management can integrate this with a CRM system