TEAM_PAN

KAGGLE USERNAME: - ds22-49@rotaract.social
Kaggle Display Name: - ds22-49
GitHub Link: - https://github.com/springyboiii/DataStorm2022.git
Lowest Total MAPE (Public): - 54.48363

Team Members

Arvinth Sugumar
Prathushan Inparaj
Nishaanthini Gnanavel

Overview

Our whole process can be divided into 4 main steps.

1. **Data Pre-Processing:** We used **pandas** to import, export and handle data frames, and **numpy** for matrix operations on the dataset. **sklearn** was used for data analysis and making machine learning models. **Matplotlib** was used to plot and visualise data during various analyses**Label Encoding** was used to handle categorical variables. Since we are working with the datetime, we converted the dates which were in strings to datetime. We derived some new features from the date. After this our features increased from 3 to 9. And also, our target variable is weekly sales, so we derived weekly sales from daily sales.

2. **Feature Engineering:** We created some new features and selected most important features and least relevant features using the Correlation Matrix. The details of feature engineering are described in the coming section.

3. **Model Selection**: In our best we used **Light Gradient Boosted Machine (Light GBM)**. The process to select this model and reasoning is elaborated in coming sections.

4. **Train, Validation and Test sets**: We trained our model using the train set and modified and upgraded our model using the validation set to achieve more accuracy and then we tested our model in the test set.

Feature Engineering

This has been a tough task since only dates and the corresponding sales data were provided. We tried to extract as many features as we could. Week of the corresponding month, Week of the corresponding year, month, day are some of them. Apparently, Only the month feature helped us improve our models, therefore we had to drop the others. We used Total MAPE score as a benchmark to select features. Dimensionality reduction was done using backward feature extraction.

Since the test file had a WeeklySales feature, we converted our training dataset's daily sale feature into weekly sales feature by grouping the items by weeks and finding the sum of daily sales of the item for each week.

We swapped the dateID column with datetime objects and extracted the features mentioned above and tested their correlation and feature importance and found only the month feature amplifying our model's performance. Then we further label encoded the month feature from mapping October(to 1) to March(to 6). We also label encoded the category code feature to be able to use it in our analysis. We encoded "category_1" as 1 and so on.

We carried out a different process for our validation and test datasest to match the feature set of our feature engineered train dataset. Since week 1, 2 will be in February and week 3,4 will be in March, we created(label encoded) a new month feature to accommodate the same feature set as our training dataset.

Final Model

Light Gradient Boosted Machine (Light GBM) is the best model we used, but before coming to this model, we trained our dataset using different models. As usual, we started with the basic model for regression, which is Linear Regression, but we got a high Total Mean Absolute Percentage Error (Total MAPE) score. Then we observed that there is no linear relationship between the features we selected and the target variable with the help of plot diagrams. Next, we used the **XGBoost Regressor** model to train our data and we got 67.26 for Total MAPE then by tuning the hyperparameters of the XGBoost Regressor model, we were able to improve the accuracy and decrease the Total MAPE score to 66.25. Later, we learned that Random Forest regressor and Light GBM models can be better machine learning models for time series forecasting problems.

First, we tried the **Random Forest Regressor** model as it can improve performance and reduce overfitting. We first tried the Random Forest Regressor model with the default values and then we tried out different values for the hyperparameter n_estimator, which is the number of decision trees that build in the forest, and we got an optimum value for n_estimator, which is 2000. We got a Total MAPE score of 55.91 for this model.

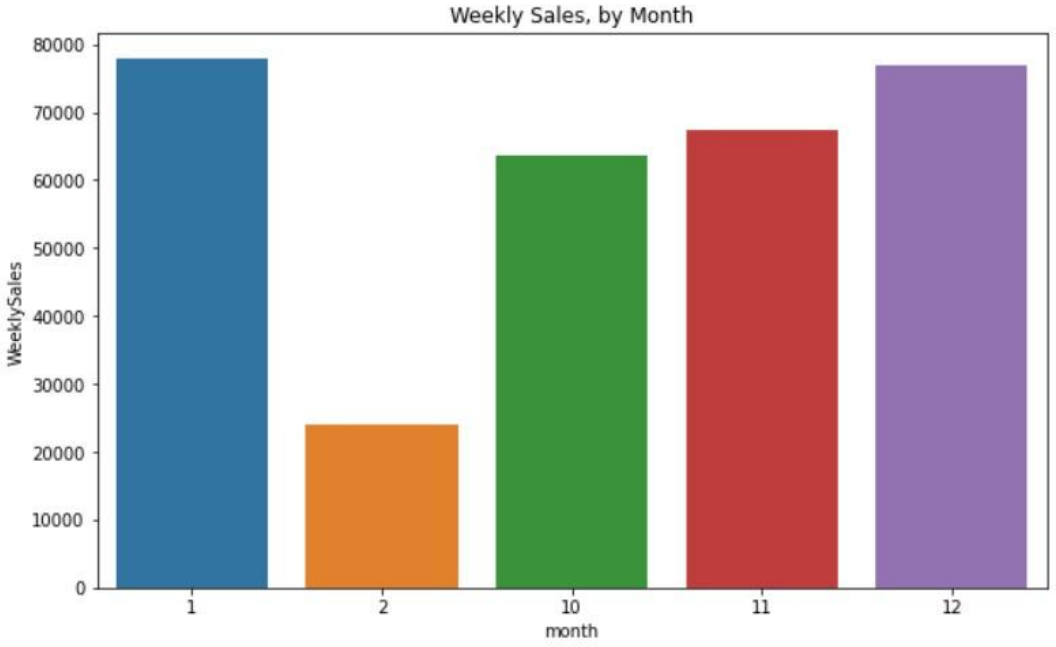
After some point of hyperparameter tuning of the Random Forest Regressor model, we couldn't improve the performance, so we thought of using the Light GBM model to train our data. Light GBM is an extension of the gradient boosting algorithm, but it focuses on boosting examples with larger gradients so it can speed up training and improve performance for forecasting problems. We trained our data with the Light GBM model by changing different sets of values for the hyperparameters, n_estimator, and learning_rate and we obtained the optimum values for n_estimator and learning_rate, which are 2000 and 0.3 respectively. We got a Total MAPE score of 55.27. Then, we again used some feature importance and feature selection techniques on our dataset and we trained using the Light GBM model with the above-mentioned hyperparameter values and we got our best Total MAPE score which is **54.48**.

Business Insights and Predictions

- Correlation matrix picked the month of sales as the most correlated feature.

	CategoryNo	ItemCode	Week	WeeklySales	year	month	week_of_year
CategoryNo	1.000000	0.084967	-0.019253	0.182079	-0.017613	0.015787	0.018280
ItemCode	0.084967	1.000000	-0.000878	-0.164303	-0.003941	0.004193	0.005327
Week	-0.019253	-0.000878	1.000000	0.003113	0.800788	-0.706233	-0.666577
WeeklySales	0.182079	-0.164303	0.003113	1.000000	0.002097	-0.001981	0.001040
year	-0.017613	-0.003941	0.800788	0.002097	1.000000	-0.987838	-0.917410
month	0.015787	0.004193	-0.706233	-0.001981	-0.987838	1.000000	0.926327
week_of_year	0.018280	0.005327	-0.666577	0.001040	-0.917410	0.926327	1.000000

- Simple graphical analysis backed this up as December and January marked the highest number of weekly sales. Christmas and New year must have played an important part in this scenario



- Item ‘169504’ had the highest number of sales during the given time period and the demand kept increasing as time went by. They should make sure their supply of 169504 doesn’t dry out in the coming months.

	CategoryNo	ItemCode	Week	WeeklySales	year	month	week_of_year	monthsales	itemsales
1783	3	169504	3	669	2021	10	42	63513	25556
2791	3	169504	13	653	2021	12	51	77028	25556
3303	3	169504	18	360	2022	1	4	77884	25556
1617	3	169504	6	716	2021	11	45	67418	25556
4194	3	169504	8	676	2021	11	46	67418	25556
...	...	...	...	...	...	...	...	...	...
6379	2	64978	16	2	2022	1	2	77884	184
4022	2	64978	13	4	2021	12	52	77028	184
6438	2	64978	9	16	2021	11	47	67418	184
6466	2	64978	14	4	2022	1	52	77884	184
7184	2	64978	5	7	2021	11	44	67418	184

- They should start to collect more data such as corresponding item price on a particular date to anticipate demand before price hikes, Number of items bought by a customer to manage footfall traffic and so on.

- Rest of the items bought together with a specific item can be used to classify and group items which can prove to be a vital feature to predict sales.

- Ambient temperature and weather can also come in handy when predicting sales of specific items such as beverages and food.