

TEAM_PAN

GitHub Link: -

<https://github.com/springyboiii/DataStorm2022SemiFinals.git>

Lowest Total MAPE score: - 37.98

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 dataframes, 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 visualize 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.

2. **Feature Engineering:** We created some new features and selected the 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 model, we use **Random Forest Regressor**. The process to select this model and reasoning is elaborated in the sections below.

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

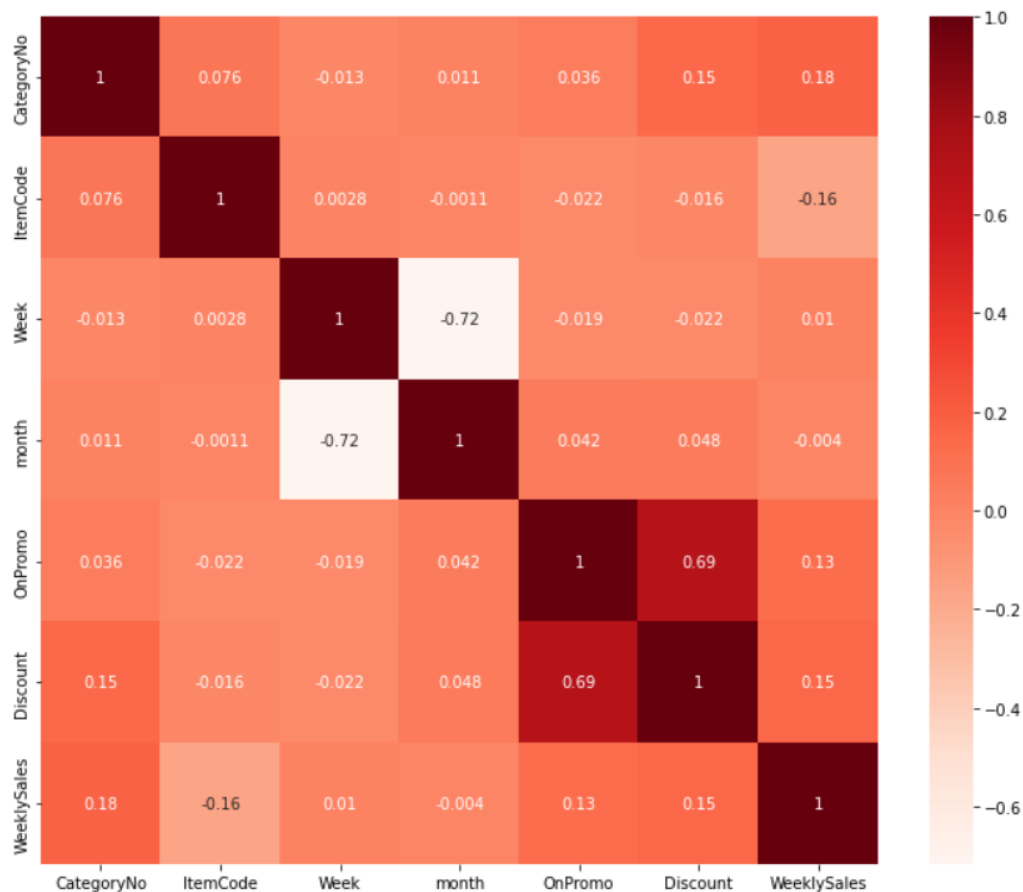
First, We label encoded the 'CategoryCode' feature as it was in object type. For example, 'category_1' was encoded as 1 in our dataset. Then we changed the 'DateID' feature to the DateTime feature to use it in our feature creation process. We created a new feature called, 'Week' by finding the number of weeks from the start date in the training dataset which is 1st October 2021. Since the validation and test file had a WeeklySales feature, we converted our training dataset's daily sale feature into a weekly sales feature by grouping the items by weeks and finding the sum of daily sales of the item for each week. We also extracted some new features like, 'year', and 'month' from the 'DateID' feature to use in our analysis process.

Then we did some feature engineering work in the promotion dataset. First, we changed the 'PromotionStartDate' and 'PromotionEndDate' features to the DateTime format. Using the 'PromotionStartDate' feature, we calculated the number of weeks from the 1st of October and named the new feature, 'Week'. Then, we merged the feature-engineered training dataset and promotion dataset using the left outer join on features, 'ItemCode' and 'Week'.

From the newly merged data frame, we found the items with promotions and without promotions for that particular week and created a new feature, 'OnPromo' where we label the ones with promotions as 1 and without promotions as 0. We created another feature,

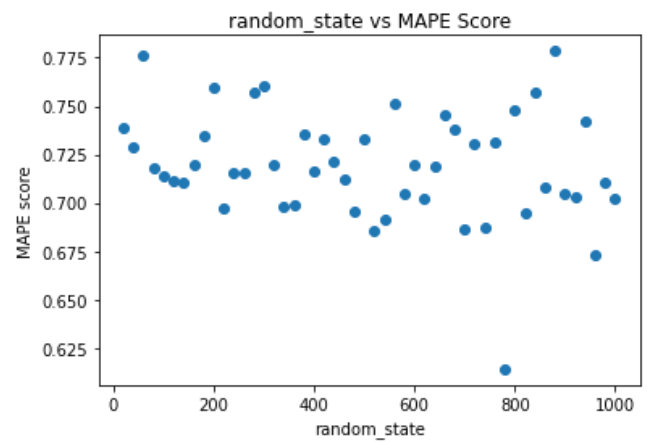
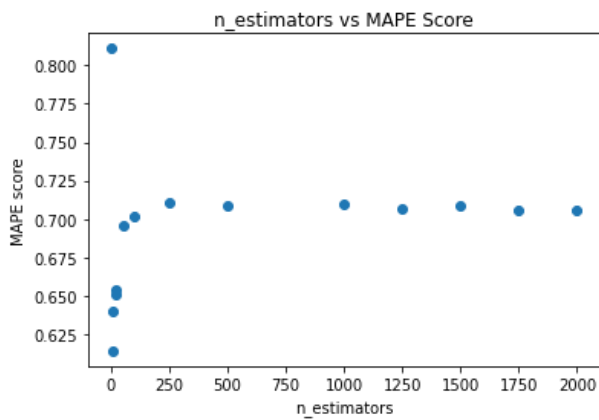
Discount' by calculating the amount of discount for items with promotions using the 'DiscoutValue', 'DiscountType', and 'SellingPriceFeatures'.

After plotting these features in a HeatMap, we selected some important features such as, 'CategoryCode', 'ItemCode', 'OnPromo', and 'Discount'.



Modeling Approaches

We used the same models for both items that are on promotion as well as the forecasting of items that are not on promotion. As the first step, we trained our training dataset using the basic model which is Linear Regression but the score for MAPE was too large. Then, We tried the XGB **Regressor** model to train our data and got a MAPE score of 77.91. From our past experience, Random Forest Regressor and **LightGBM** can be better models for time series forecasting problems. So next we tried using the **LightGBM** model to train our data and we got a MAPE score of 76.2 and then, we tried **Random Forrest Regressor** to train our data and our MAPE score was improved considerably to 71.47. Then we tried out different values for the hyperparameters n_estimators and random state. We got optimum values for n_estimators and random_state, which are 10 and 780 respectively. We got a MAPE score of 61.41 for this model.

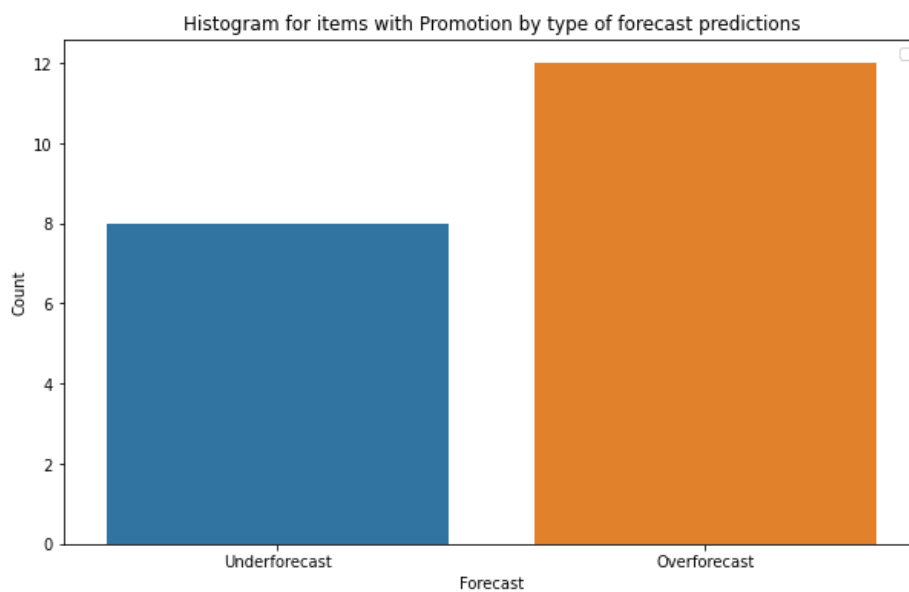


Evaluation Metrics

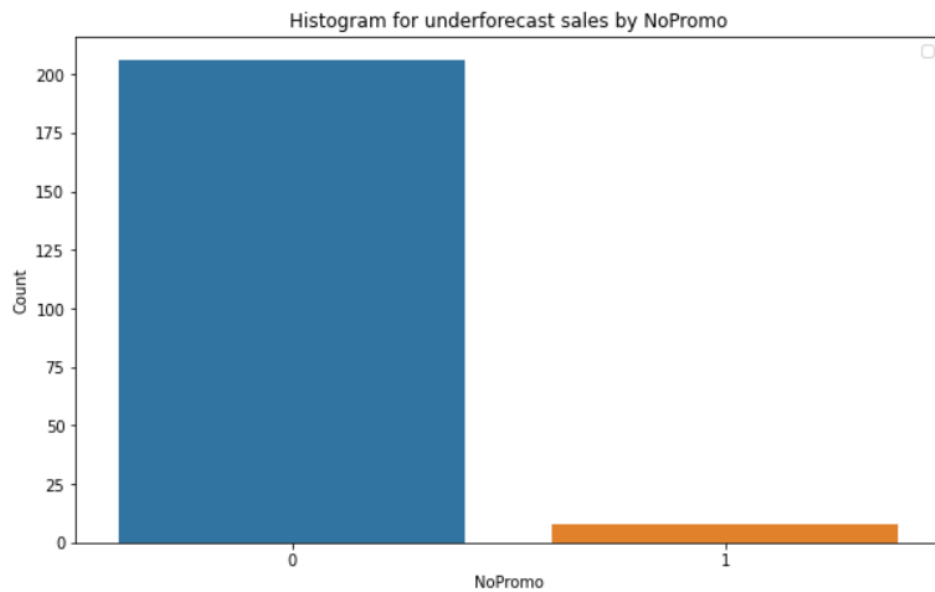
We used Mean Absolute Percentage Error(MAPE), Total Mean Absolute Percentage Error (Total Mape) for overall errors, and MAPE (UnderForecast) score for under forecast sale errors to evaluate our models.

Business Findings

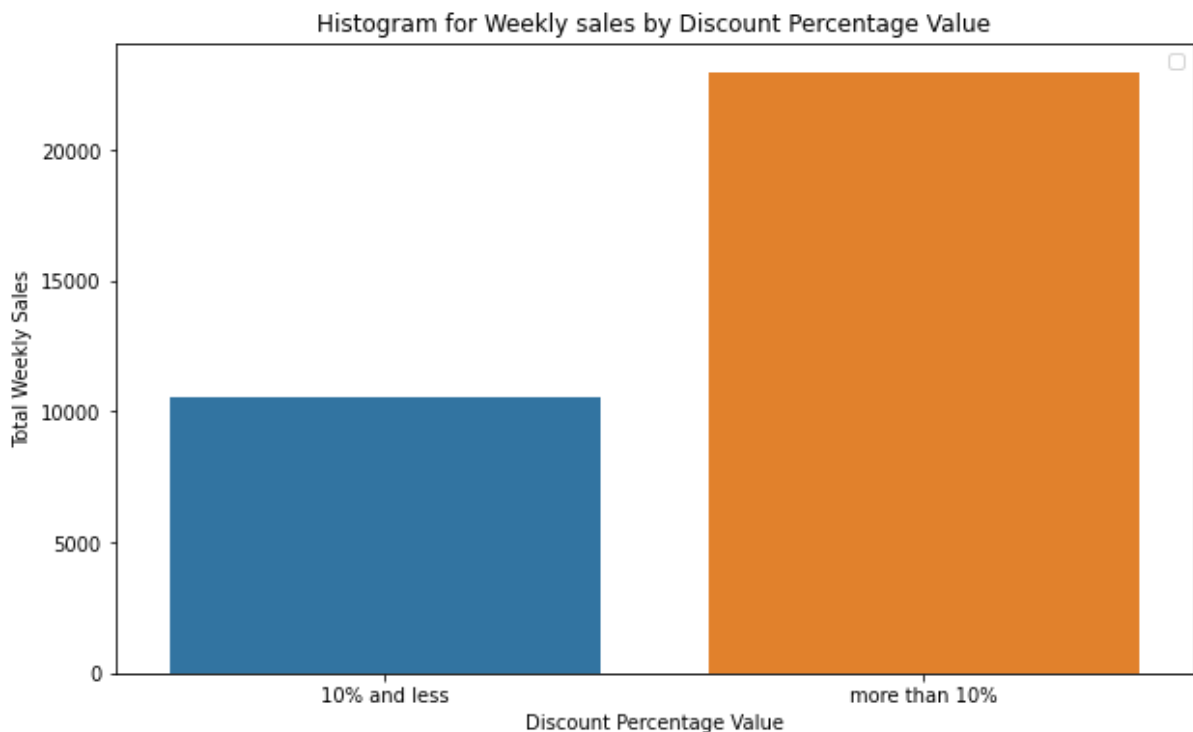
- 1) Predicted sales of the promoted items are higher than the actual sales for the majority of those items. From this, we can convey that items with promotions will have higher sales.



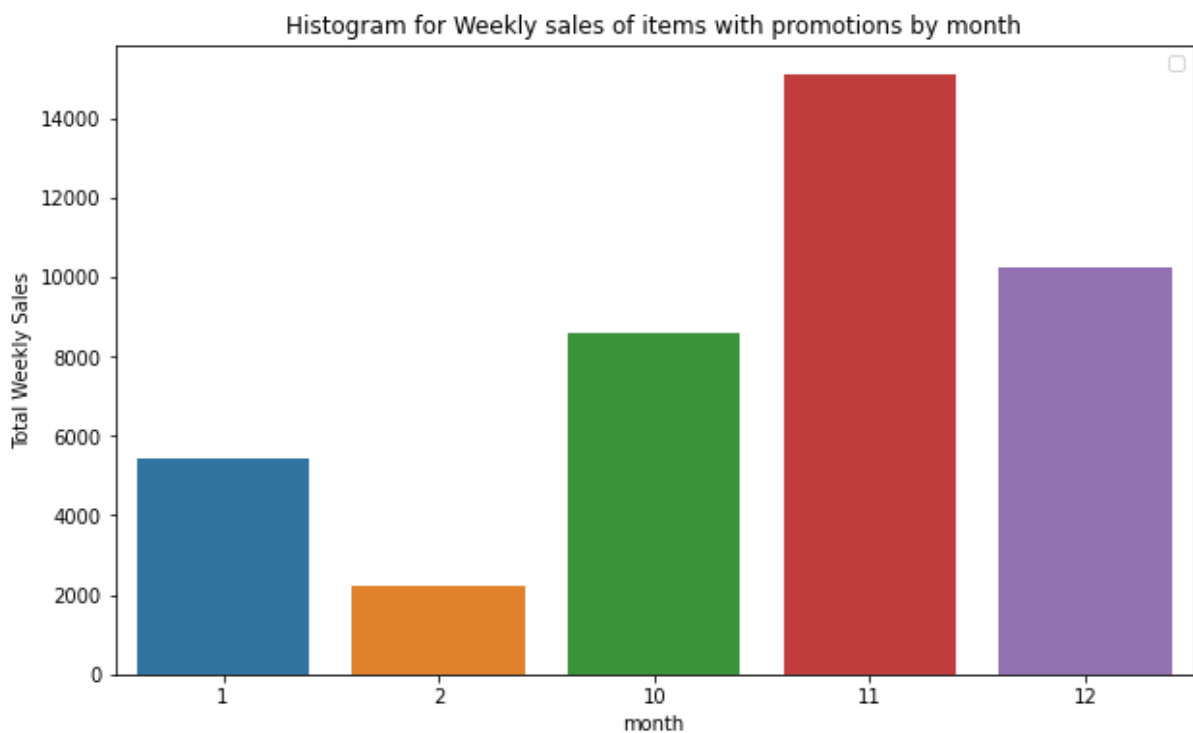
- 2) Most of the items with under forecast sales have no promotions. From this, we can see that promotions for items are necessary to have more sales.



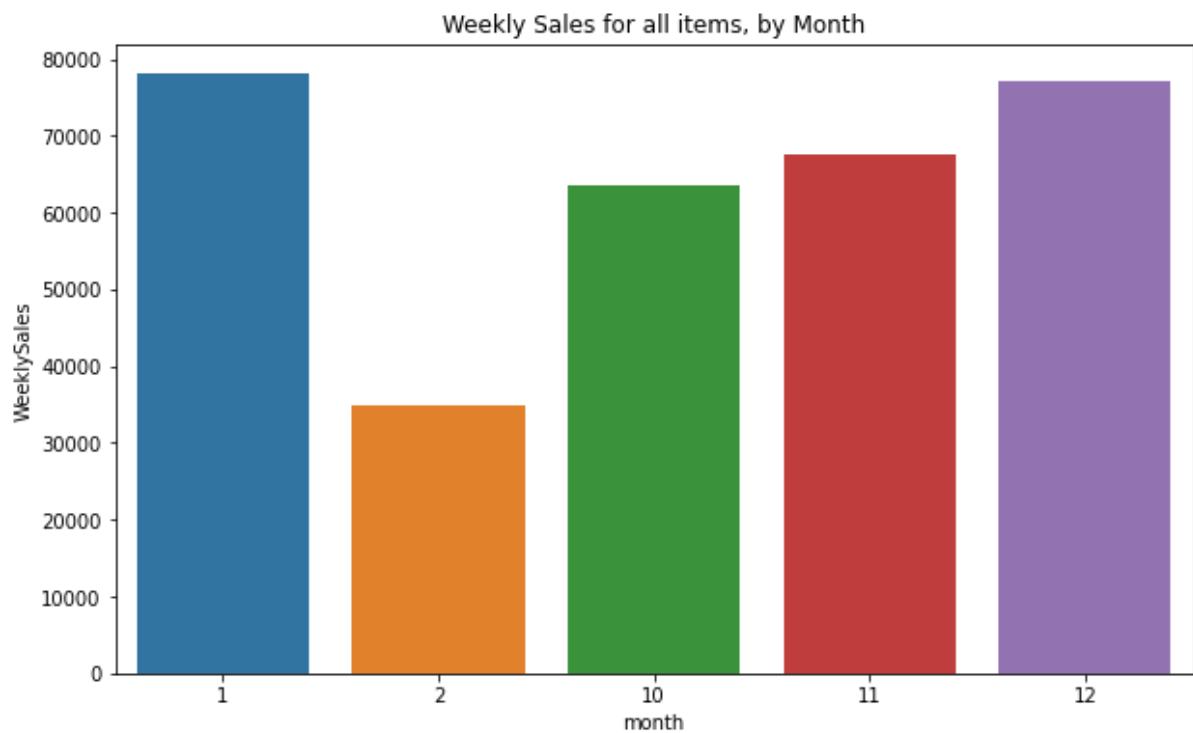
- 3) Total weekly sales of items with higher discount values is greater than those with lower discount values. Therefore, to attract more customers and to increase the sales of items, it is better to increase the discount percentage by a reasonable value.



4) November and December have higher total weekly sales for items with promotions than other months because there will be more promotions for items at year-end.



5) 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.



6) 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.

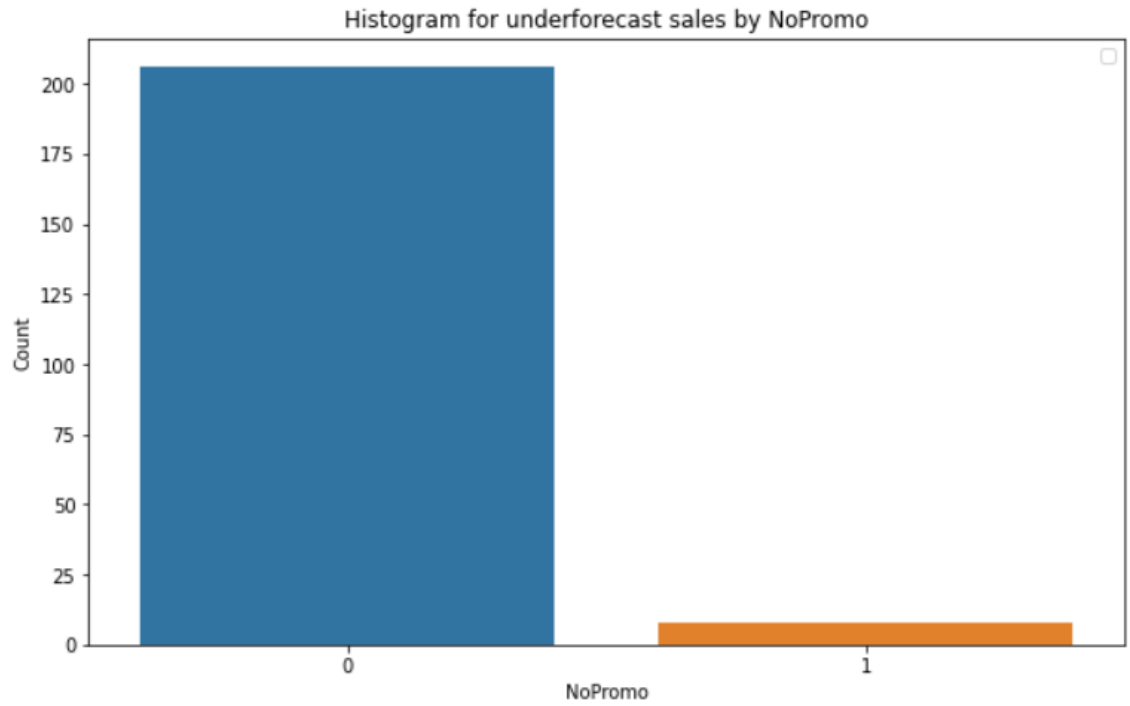
	CategoryNo	ItemCode	Week	month	WeeklySales	monthsales	itemsales
1829	3	169504	18	1	360	78090	26303
2081	3	169504	8	11	676	67604	26303
4129	3	169504	5	11	727	67604	26303
1661	3	169504	16	1	487	78090	26303
6561	3	169504	2	10	627	63648	26303
...
4674	2	64978	3	10	3	63648	194
154	1	1068883	12	12	4	77193	194
5160	2	64978	14	1	4	78090	194
4814	2	64978	1	10	5	63648	194
7773	1	1068883	3	10	2	63648	194

Descriptive approach to assess the under forecasting error

After predicting the sales from the model, we merged the predicted sales values with the validation dataframe, and filtered out the rows where the predicted sales value is less than the actual weekly sales values and then by using the MAPE(under forecast) evaluation metric equation, we measured the error rate of our prediction for UnderForecast sales.

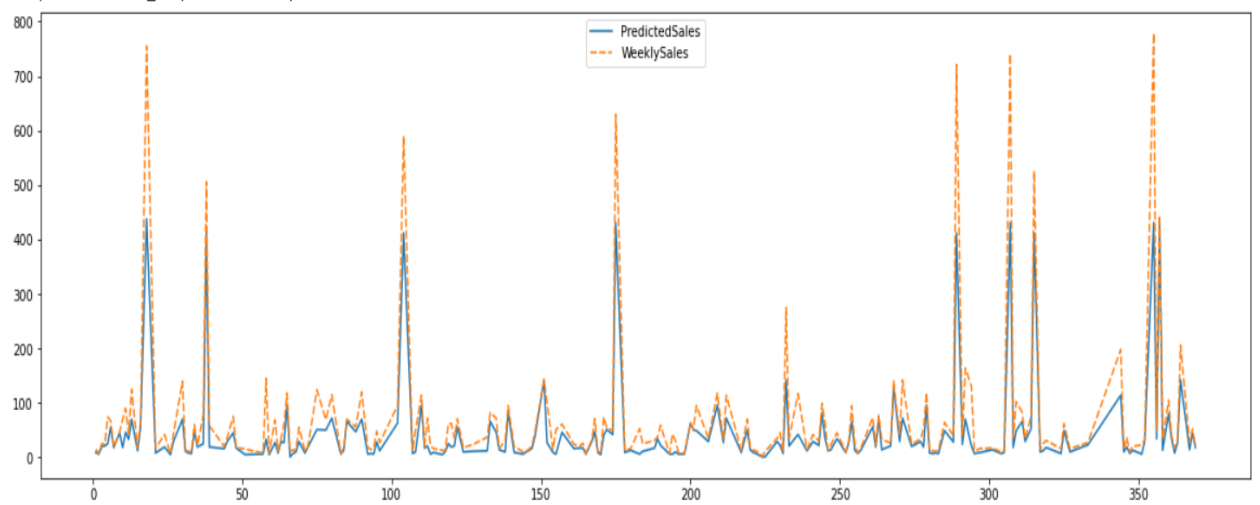
To assess the under forecast errors and to find ways to minimize them we did some descriptive analysis.

- 1) The number of items which are under forecasted according to whether there is a promotion or not.



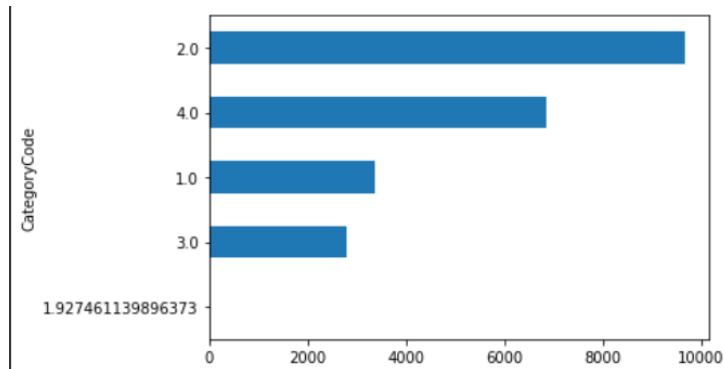
2) To analyze how far the prediction has under forecasted.

<matplotlib.axes._subplots.AxesSubplot at 0x7f8765264b90>



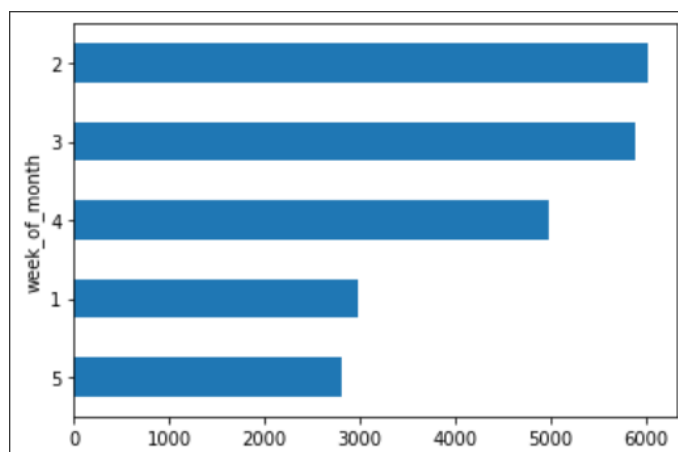
Descriptive Analysis

CategoryCode-DiscoutAmount



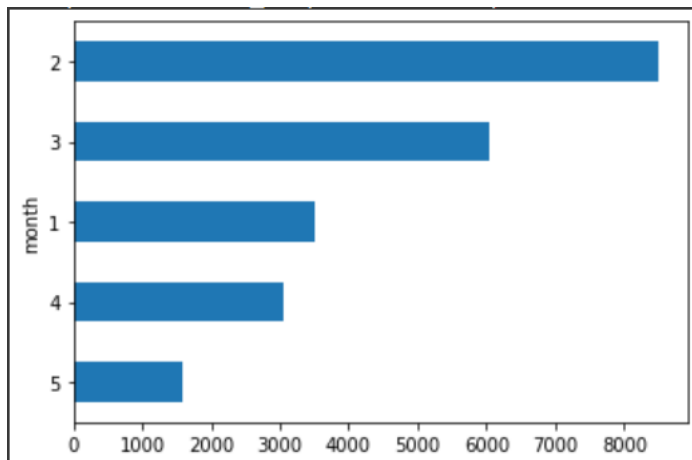
- The shop has discounted category_2 items the most in terms of value. Discounting category_1,3 items will improve the sales more.
- **Intervention - Discounting category_1,3 items will improve the sales more of those respective categories.**

CategoryCode-Week of month

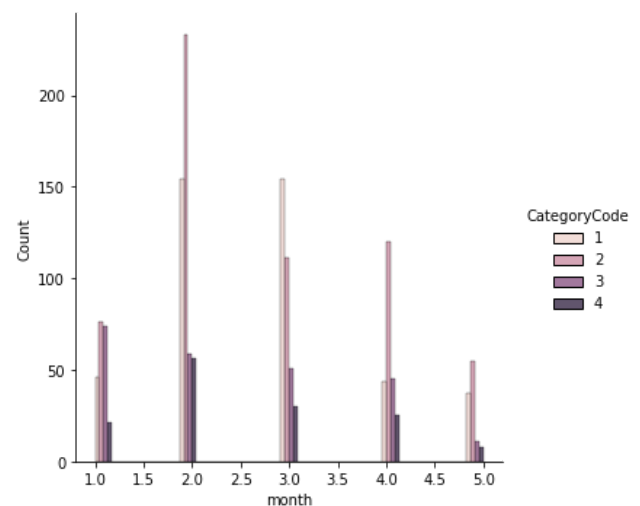
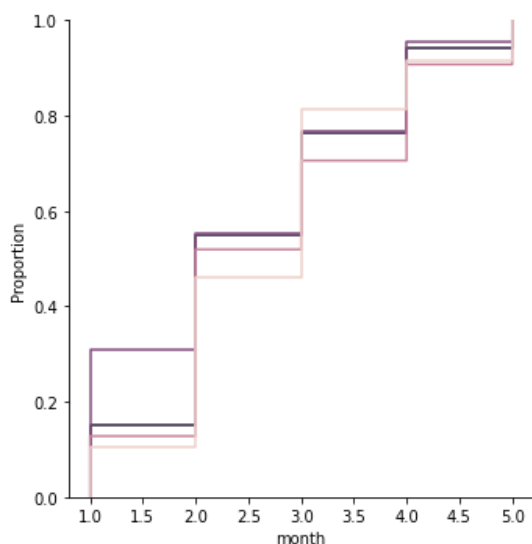


- They have always preferred the latter weeks of a month to provide discounts. Since the last week(5th) usually ends up with 2-3 days, multiplying that by 2 proves this stake further.
- **Intervention - Discounting right at the start of each month will bring in more sales and avoid clearance sales at the month ends. Also, this approach will ease the business process even more.**

DiscountAmount-Month



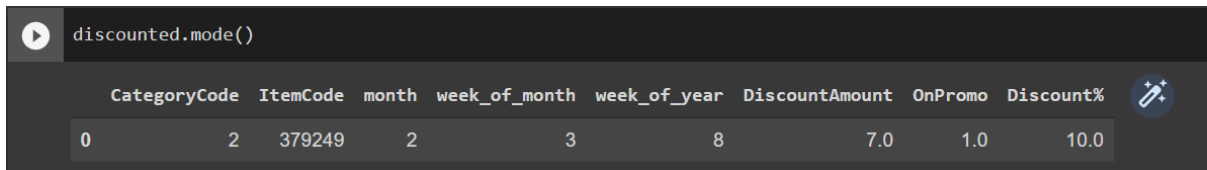
- November(2) and December(3) recorded the most number of promotions beating other months quite easily. End-of-the-year promotions must have played a bigger role in this.
- **Intervention - Discounting right at the start of each month will bring in more sales and avoid clearance sales at the month-ends.**



- The 2 graphs above show the major influence of category_3 items being discounted in October(1), November(2), and January(4) while the trend shifts as category_1 items get discounted mostly in December and February(5)

Mode

- ItemCode 379249 has been the most discounted product while category_2 items were often discounted. 3rd week of month was the most popular month for discounts and the 8th week from the starting date recorded the most discounts. 10% discount was the go-to in this shop



```
discounted.mode()
```

	CategoryCode	ItemCode	month	week_of_month	week_of_year	DiscountAmount	OnPromo	Discount%
0	2	379249	2	3	8	7.0	1.0	10.0

Median

- DiscountAmount 8.0
- Discount% 10.0

Mean

- DiscountAmount 16.086170
- Discount% 12.748227

IQR

- DiscountAmount 13.0
- Discount% 5.0

Range

- DiscountAmount 82.5
- Discount% 25.0

Standard Deviation

- DiscountAmount 16.107022
- Discount% 5.406951

Variance

- DiscountAmount 2.594361e+02
- Discount% 2.923512e+01

Skewness

- Mean and Median of DiscountAmount(Rs) and Discount% is higher than the mode.
Therefore DiscountAmount and Discount% are positively skewed.

Pairplot



