DATA STORM 3.0

Sales forecasting Project report

Team Name- XGBoosters

Github repo Link - https://github.com/Vipooshan-1998/Data-Storm-3.0---Storming-Round

Username: ds22-45

Display Name : ds22-45

Team members:

V. Vipooshan

T.Thuvaaragan

S.Sarangan

Lowest total MAPE: 89.95782

Approach

After going through the datasets and the problem statement, it was clear that this problem can be seen as a Time series problem. In our dataset we only have sales numbers with date_id,category_code and item_code. Also, it is impossible to create new features from the given features. So it is inappropriate to consider this as a traditional prediction problem.

So we sorted the data as weekwise and categorywise and then itemwise. Then we visualised the dataset (Fig 1) along with the time(ie date) and we were able to see a pattern in the data which led us to consider Time series.

Then ,although the whole dataset has a pattern, predictions tend to deviate. So we considered an alternate way. Since the predictions for each item are expected individually, we selected some items randomly and obtained the data related to that item only, and did the Dickey-Fuller Test. The p value was relatively low as expected.As data looked stationary, we proceeded with the time series analysis for every individual item.(Fig 2)

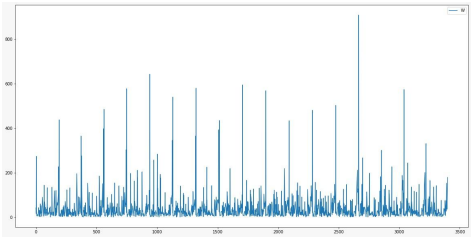

Fig (1): Visualisation of whole data in the validation Dataset along time

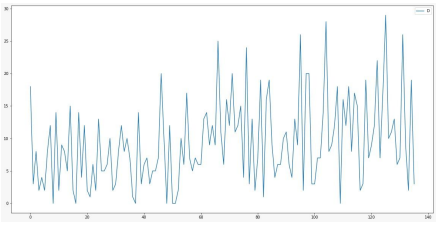

Fig (2): Visualisation of one item along with time item id =3418

```
Results of Dickey-Fuller Test:
Test Statistic                -1.719249e+01
p-value                        6.538477e-30
#Lags Used                     3.000000e+00
Number of Observations Used    1.669000e+03
Critical Value (1%)           -3.434274e+00
Critical Value (5%)           -2.863273e+00
Critical Value (10%)          -2.567693e+00
dtype: float64
```

Fig (3): Dickey-Fuller Test for data given by fig(1)

```
Results of Dickey-Fuller Test:
Test Statistic                  -4.999904
p-value                          0.000022
#Lags Used                       2.000000
Number of Observations Used    116.000000
Critical Value (1%)             -3.488022
Critical Value (5%)             -2.886797
Critical Value (10%)            -2.580241
dtype: float64
```

Fig (4): Dickey-Fuller Test for data given by fig(2)

Data organisation and preprocessing

When analysing the dataset we found that Training data has 194 items.Validation data has 95 items in it and testing data has 97 items. We found that validation data and test data are mutually exclusive.

We divided the Training data as two parts, so that we get the items as in the validation data in one part (Train A) and items as in the testing data (Train B) as another part. With our models We predicted the values for validation data(ie Train A) and compared them with the actual data given to us in the validation dataset. We chose the models performed well in this prediction using the Total Mean absolute percentage error value.Those models will perform well on testing data(Train B).

We have considered 2 different approaches.

- we can do forecasting for dates and then convert those days as weeks
- we can convert the dates as weeks and then do the forecasting for weeks

Tools and Libraries

- Pandas for dataframe handling
- Numpy for array manipulations
- Matplotlib for visualisations purposes
- Keras and Tensorflow
- Adfuller from statsmodel
- Time series models
    1. ARIMA
    2. Prophet
    3. Neural Prophet
    4. LSTM

Model selection

For models we have considered several models.

- ARIMA

    It is one of the most commonly used models for Time series analysis. but we weren't able to generate the expected output from the model. The Total MAPE value was high and even after doing a lot of tuning, we weren't able to attain the expected results.

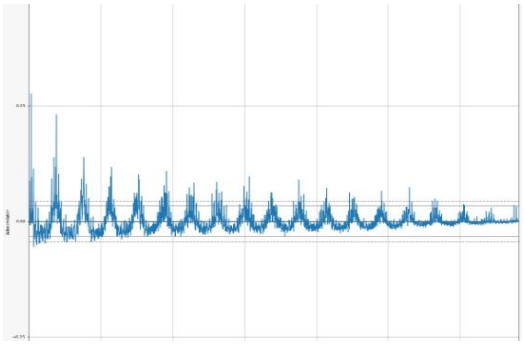

Fig (5): Autocorrelation plot

- Prophet and Neural-Prophet

To improve accuracy we switched to the Prophet model. The model demands inputs as ds and y. So we assigned date to ds and daily sales for y. Then using the data we forecasted the next 28 days and converted it as weekly sales for every item. This model performed better than ARIMA for us with the given dataset But still the results were unsatisfying.

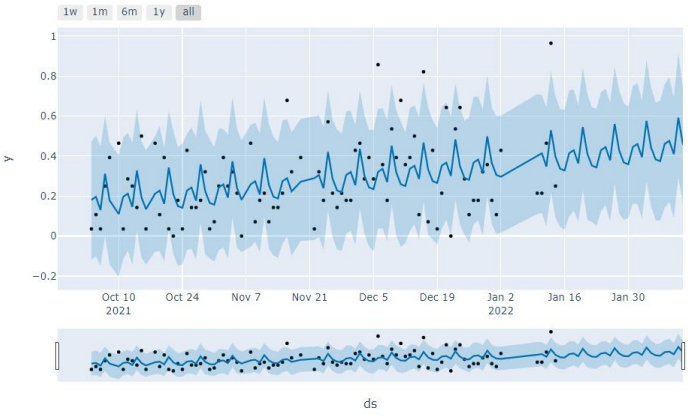

Fig (6): plot showing the Forecast of a single item using the prophet model

- LSTM

Since we have been using only time series models we wanted to try deep learning based models also. And LSTM models are best fit to handle sequential data.So we needed to convert the time series problem as a supervised machine learning model.

	DateID	WeeklySales	diff	lag_1	lag_2	lag_3	lag_4	lag_5	lag_6	lag_7	lag_8	lag_9
0	2021-12-18	28.0	24.0	-3.0	-19.0	5.0	-8.0	-36.0	38.0	13.0	3.0	0.0
1	2021-12-26	8.0	-20.0	24.0	-3.0	-19.0	5.0	-8.0	-36.0	38.0	13.0	3.0
2	2021-12-30	11.0	3.0	-20.0	24.0	-3.0	-19.0	5.0	-8.0	-36.0	38.0	13.0
3	2022-01-11	11.0	0.0	3.0	-20.0	24.0	-3.0	-19.0	5.0	-8.0	-36.0	38.0
4	2022-01-22	6.0	-5.0	0.0	3.0	-20.0	24.0	-3.0	-19.0	5.0	-8.0	-36.0
5	2022-01-28	7.0	1.0	-5.0	0.0	3.0	-20.0	24.0	-3.0	-19.0	5.0	-8.0
6	2022-02-03	10.0	3.0	1.0	-5.0	0.0	3.0	-20.0	24.0	-3.0	-19.0	5.0
7	2022-02-08	7.0	-3.0	3.0	1.0	-5.0	0.0	3.0	-20.0	24.0	-3.0	-19.0
8	2022-02-14	0.0	-7.0	-3.0	3.0	1.0	-5.0	0.0	3.0	-20.0	24.0	-3.0
9	2022-02-21	0.0	0.0	-7.0	-3.0	3.0	1.0	-5.0	0.0	3.0	-20.0	24.0
10	2022-02-28	0.0	0.0	0.0	-7.0	-3.0	3.0	1.0	-5.0	0.0	3.0	-20.0
11	2022-03-07	0.0	0.0	0.0	0.0	-7.0	-3.0	3.0	1.0	-5.0	0.0	3.0

Fig (7): Created features from difference between adjacent weeks(lags) for item=117610

The method is to get the difference in sales compared to the previous weeks and build the model on it.We used previous weekly sales data to forecast the next ones.

This model performed very well for our dataset compared to all other models we tried. So we chose this model and tuned it and generated the forecast.

Business insights

- Insights from model and intervention that can be made by management

		Weekly Sales			
		sum	mean	median	std
CategoryCode	DateID				
category_1	2022-02-14	1207	38.935484	18.0	67.499104
	2022-02-21	1652	55.066667	23.5	96.185859
	2022-02-28	2115	64.090909	37.0	123.676333
	2022-03-07	1859	56.333333	29.0	130.509498
category_2	2022-02-14	1189	25.297872	17.0	27.929971
	2022-02-21	1572	33.446809	20.0	38.919543
	2022-02-28	1655	35.212766	18.0	54.398664
	2022-03-07	1737	36.957447	19.0	49.186848
category_3	2022-02-14	358	35.800000	39.5	23.327618
	2022-02-21	446	44.600000	41.5	40.699986
	2022-02-28	321	35.666667	42.0	22.079402
	2022-03-07	376	37.600000	28.5	27.166973
category_4	2022-02-14	250	83.333333	66.0	70.613974
	2022-02-21	360	72.000000	53.0	47.639270
	2022-02-28	378	126.000000	84.0	92.455395
	2022-03-07	346	69.200000	48.0	61.112192

Fig (8): Analysis that shows the forecasted items' weekly sales category wise.

This table shows that category 4 & 2 show higher liquidity than category 3 & 1. So the management can take necessary actions to maintain the availability of those high demanding items and also may limit stocks of less moving categories. And also we can do the same analysis for each item individually and get the insight about each item and maintain stock according to that.

- Some useful attributes that can make the forecast even more reliable

  ➔ Management can collect the data related to the holidays and festival days and use those data for the analysis as sales are very much correlated with those attributes.

  ➔ Management can introduce a review system for the items. So that those reviews can be used in the future predictions which can be a strong deciding factor.