Assignment 2:
Machine Learning Model as a Service

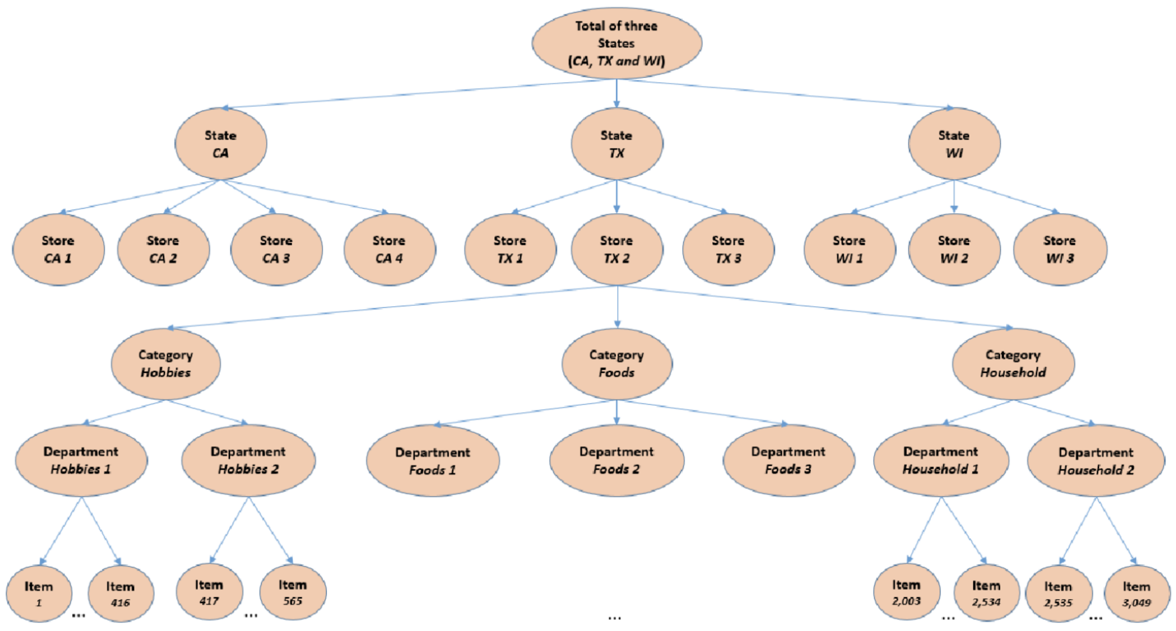
Name: Tarun Gupta
Student Id: 24607855
Date: 10/OCT/2023

36120 - Advanced Machine Learning Application
Master of Data Science and Innovation
University of Technology of Sydney

Table of Contents

Introduction:

We have given a dataset of an American retailer which has 10 stores in 3 different states of America which are California(CA), Texas(TX) and Wisconsin(WI). There are 3 category items available in each store which are Hobbies, Household and foods.
Below is the high level structure of the stores and items present.



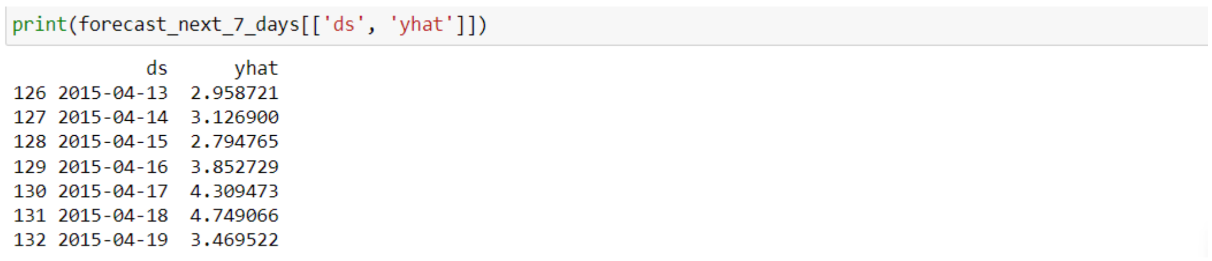
The dataset provided has been divided into 4 different csv files.

- **Sales** - the dataset containing the information about the stores, items, categories and the sales on the basis of days.
- **Calendar -** This dataset contains the date, id of the week and the mapping of days.
- **Calendar_Events:** This dataset contains the information about events happening on the basis of date. This dataset has the features as date, event_name and event_type.
- **Item_pricing:** This dataset has the records of selling information containing sell_price, item-id, id of the week and store_id.

On the basis of given dataset, We have been asked to perform the machine learning model
- Which will predict the sales revenue on the basis of item, store and the date.
- Which will forecast the total sales revenue across all the stores for the upcoming days.

Also, the machine learning model needs to be served as a **Production API using FastAPI and Heroku** and the models will be provided with the production api endpoints.

With the following problem statement, I have performed the machine learning models and been able to predict the sales revenue.
For the first statement, I have performed the **Random Forest Regressor Model** which is able to predict the sales revenue precisely. The accuracy achieved by the model is 46% and the MSE is 51.14.
For the second statement, I have performed the **Prophet model** for forecasting using time-series algorithm and received the result as:



```
print(forecast_next_7_days[['ds', 'yhat']])

             ds      yhat
126 2015-04-13  2.958721
127 2015-04-14  3.126900
128 2015-04-15  2.794765
129 2015-04-16  3.852729
130 2015-04-17  4.309473
131 2015-04-18  4.749066
132 2015-04-19  3.469522
```

Business Understanding:

a. Business Use Case:

On the basis of the given dataset and the problem statement given, the business team can predict the sales revenue for any item and store. Also, the business team can forecast the sales revenue for the next 7 days. The accuracy of the model will play a significant role as to predict the sales_revenue is the key point for the business team to analyse and consider the future business predictions and plan management.

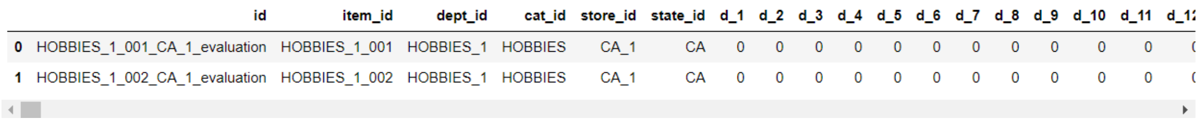
b. Key Objective:

The key objective of this experiment is to analyse and predict the sales revenue. Stakeholders will use these models as a Production API which can help to analyse the sales on the basis of items and stores. Also, this will help stakeholders forecasting the sales revenue for next 7 days.

Data Understanding:

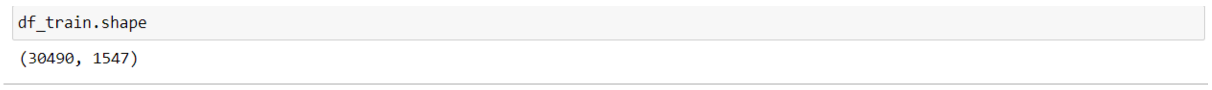
We have given the dataset in 5 different csv files containing the information of stores, items, sales, dates and events.
Provided 5 different datasets are:

- **Sales_training dataset:** This dataset has the information of the sales with respect to the days. Here is the sample of the dataset:



	id	item_id	dept_id	cat_id	store_id	state_id	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_10	d_11
0	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1_001	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0
1	HOBBIES_1_002_CA_1_evaluation	HOBBIES_1_002	HOBBIES_1	HOBBIES	CA_1	CA	0	0	0	0	0	0	0	0	0	0	0

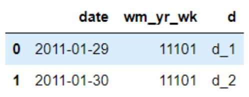
Here in the dataset, item_id is the id of item, store_id is the id of the store, cat_id is the id of category and there are features from d_1 to d_1541 which are the sales on the basis of days corresponding to item_id and store_id.
Here is the shape of the sales dataset.



```
df_train.shape
```

```
(30490, 1547)
```

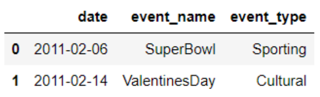
- **Calendar dataset:** This dataset contains the information about the dates with respect to the days and also the id of the week.
  Here is the sample of the dataset.



	date	wm_yr_wk	d
0	2011-01-29	11101	d_1
1	2011-01-30	11101	d_2

In the dataset, d_1, d_2 are the days mapping and the wm_yr_wk is the id of the week. We have 1969 records with 3 features as mentioned in the sample.

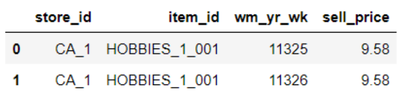
- **Calendar_events dataset:** This dataset holds the information about events happening on the basis of date in the stores.
  Here is the sample of the dataset.



	date	event_name	event_type
0	2011-02-06	SuperBowl	Sporting
1	2011-02-14	ValentinesDay	Cultural

In the given dataset, event_name and event_type are the information of events running in the store. In this dataset we have 167 records.

- **Item_pricing dataset:** This dataset has the information about the price of item, item_id, store_id and the id of the week. This dataset can be considered as the base dataset which holds the information about the selling item.
  Here is the sample of the dataset.



	store_id	item_id	wm_yr_wk	sell_price
0	CA_1	HOBBIES_1_001	11325	9.58
1	CA_1	HOBBIES_1_001	11326	9.58

This dataset has **6841121 records**. It is a huge dataset and takes time to merge and load.

- **Sales_test dataset:** This dataset is similar to the train dataset and contains the information about the sales on the basis of days only. This dataset is for the evaluation and for the understanding of the dataset.

Data Preparation:

Step 1: As we have given different datasets, we first need to prepare the dataset and merge the dataset into one dataset for further analysis.
I have learned about the dataset and finalised the item_pricing dataset as the base dataset on which I need to merge other datasets.
Also, I have checked the unique values of the store using store_id. Here is the result achieved for the store_id.

```
store_unique
```

```
array(['CA_1', 'CA_2', 'CA_3', 'CA_4', 'TX_1', 'TX_2', 'TX_3', 'WI_1',
       'WI_2', 'WI_3'], dtype=object)
```

Step 2: As the dataset is too big and leads to the out of memory error while merging the dataset with others. I have decided to split the dataset on the basis of store id.
After splitting the dataset on the basis of store_id, Now, I have **10 different store datasets**.
I have considered each dataset on by one to merge with other datasets.
Here is the command to get the dataset on the basis of store_id.
df_store = df_item_prices.loc[df_item_prices['store_id'] == 'CA_4']

Step 3: Now, the store dataset is ready, I would need to perform the merge command again and again. So, Instead of creating the command again and again, I have created the function which will accept the datasets to merge and the column on which I need to merge the dataset.

```
def merge_datasets(df1, df2, onColumn):
    df_datasets_merge = pd.merge(df1, df2, on=onColumn, how="inner")
    return df_datasets_merge
```

This merge_datasets method will accept two datasets and the column on which I need to merge as parameters. It will then perform the merge command. This command will merge the datasets on the basis of inner join.

Step 4: As the dataset is still huge and difficult to merge with other datasets, I have again splitted the dataset on the basis of **item category**(hobbies, household and food category).

```
filtered_df_hobbies = df_store[df_store['item_id'].str.contains('hobbies', case=False)]
filtered_df_foods = df_store[df_store['item_id'].str.contains('FOODS', case=False)]
filtered_df_household = df_store[df_store['item_id'].str.contains('household', case=False)]
```

```
filtered_df_hobbies.head(2)
```

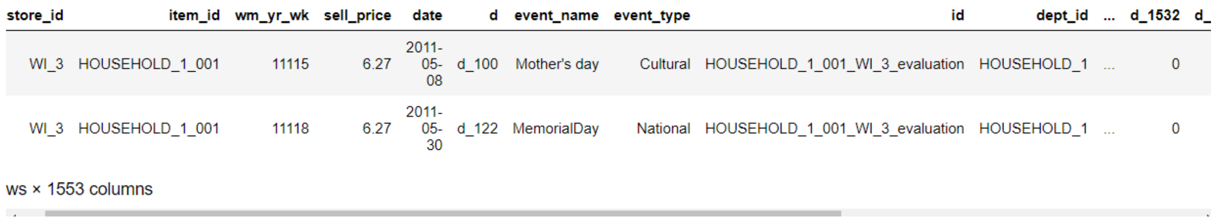
	store_id	item_id	wm_yr_wk	sell_price
698412	CA_2	HOBBIES_1_001	11327	8.26
698413	CA_2	HOBBIES_1_001	11328	8.26

Now, the individual dataset contains the information of individual stores and item categories. The dataset is now ready to merge.

Step 5: I have merged the dataset with the calendar and calendar_events dataset by using the above merge_dataset method.

Step 6: I have noticed that some of the records contain the information of the dates after than **2015-04-18** which is the last date record present in the sales_train dataset. So, before merging with the sales_train dataset, I have filtered the records which are in the range of the train dataset. By using the following command, I have filtered the records of the dataset.
filtered_df_foods = filtered_df_foods[filtered_df_foods['d'] <= 'd_1541']

Step 7: Now, merge the updated dataset with the sales_train dataset. Here is the sample of the merged dataset.



store_id	item_id	wm_yr_wk	sell_price	date	d	event_name	event_type	id	dept_id	...	d_1532	d_
WI_3	HOUSEHOLD_1_001	11115	6.27	2011-05-08	d_100	Mother's day	Cultural	HOUSEHOLD_1_001_WI_3_evaluation	HOUSEHOLD_1	...	0	
WI_3	HOUSEHOLD_1_001	11118	6.27	2011-05-30	d_122	MemorialDay	National	HOUSEHOLD_1_001_WI_3_evaluation	HOUSEHOLD_1	...	0	

ws × 1553 columns

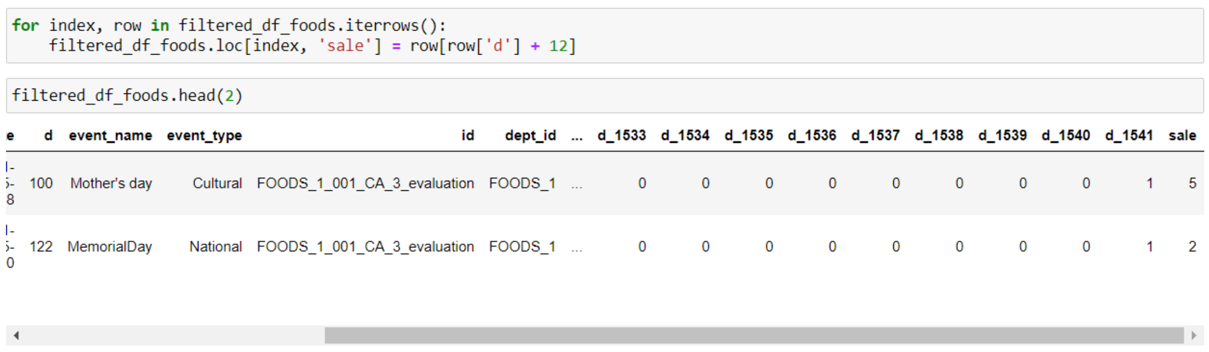
Now, all the merging process is done for individual dataset having individual store and item id, I then proceed to fetch the days sale on the basis of dates.

Step 8: To fetch the day's sale on the basis of dates, we first need to understand a few of the things from the dataset.

- Day column present on which position. Basically, the days column starts from 13th position in the merged dataset.
- The 'd' column in the merged dataset represents which days column.

After analysing this, I have first filtered the 'd' column and extracted the value present.
For instance, if the 'd' column has a value as d_193, the result will be 193.
To perform this, I have written a command as:
filtered_df_foods['d'] = filtered_df_foods['d'].str.extract(r'd_(\d+)')
Where filtered_df_foods is the merged dataset of the food category.
After this, I have transformed the d column from string data type to int datatype.

Step 9: I have created the new column as 'sale' which will hold the number of sales done on the specific day. The sale of the day has been calculated on the basis of the 'd' column. I have fetched the value of **d+12 column** as in my case, the day's column started from 13th position. This has done by:



```
for index, row in filtered_df_foods.iterrows():
    filtered_df_foods.loc[index, 'sale'] = row[row['d'] + 12]
```

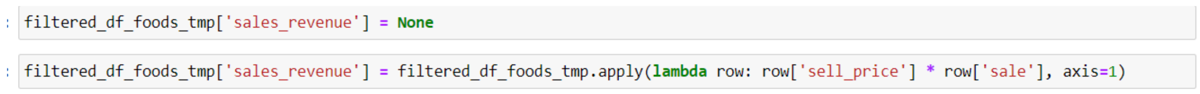
```
filtered_df_foods.head(2)
```

e	d	event_name	event_type	id	dept_id	...	d_1533	d_1534	d_1535	d_1536	d_1537	d_1538	d_1539	d_1540	d_1541	sale
l- 5- 8	100	Mother's day	Cultural	FOODS_1_001_CA_3_evaluation	FOODS_1	...	0	0	0	0	0	0	0	0	1	5
l- 5- 0	122	MemorialDay	National	FOODS_1_001_CA_3_evaluation	FOODS_1	...	0	0	0	0	0	0	0	0	1	2

In the result, we can see that there is a new column named sale having the sale of a specific day.

Step 10: As the data is fetched from the days column, so Now I have dropped all the columns starting from ‘d_’.

Step 11: I have created one more column as sales_revenue which will show the revenue on the basis of sell_price and the sale.

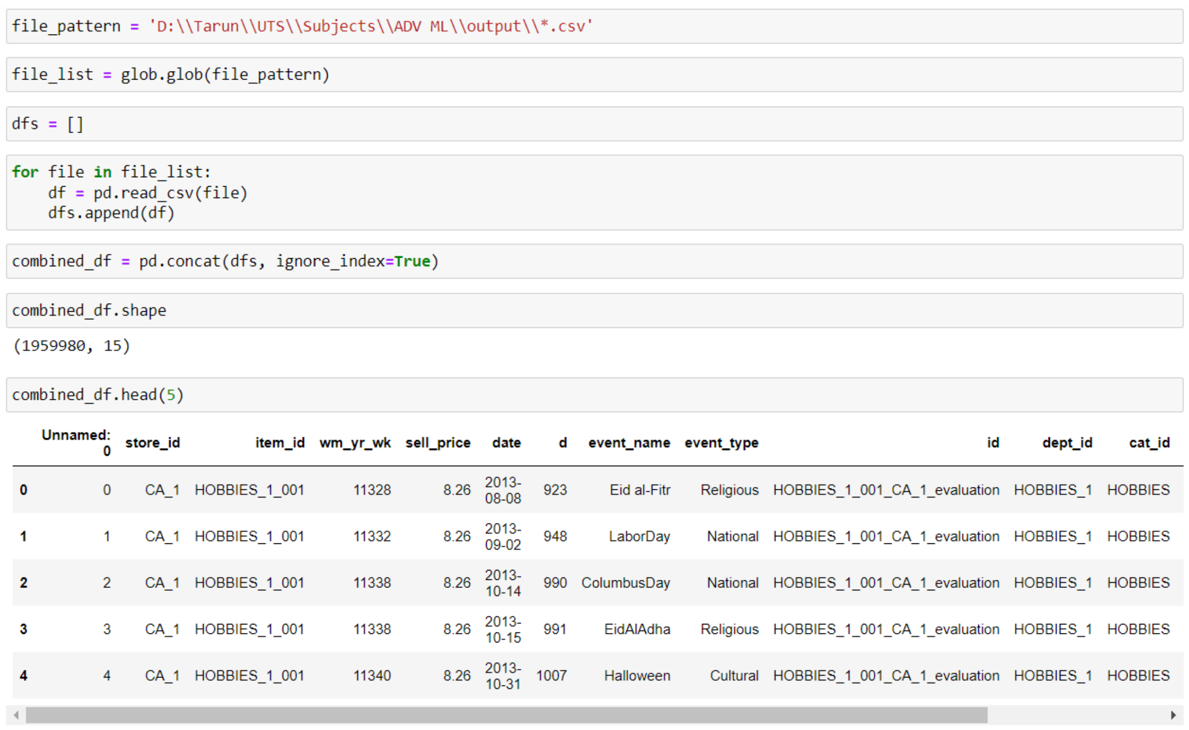


```
filtered_df_foods_tmp['sales_revenue'] = None
```

```
filtered_df_foods_tmp['sales_revenue'] = filtered_df_foods_tmp.apply(lambda row: row['sell_price'] * row['sale'], axis=1)
```

Now, the merged dataset is ready for one category item and one store. I have performed all the steps for each store and category data and saved the merged file in the internal storage.

Step 12: As the merged data is ready in multiple csv files, we need to merge all and create one combined dataset. I have performed this operation as:



```
file_pattern = 'D:\\Tarun\\UTS\\Subjects\\ADV ML\\output\\*.csv'
```

```
file_list = glob.glob(file_pattern)
```

```
dfs = []
```

```
for file in file_list:
    df = pd.read_csv(file)
    dfs.append(df)
```

```
combined_df = pd.concat(dfs, ignore_index=True)
```

```
combined_df.shape
```

```
(1959980, 15)
```

```
combined_df.head(5)
```

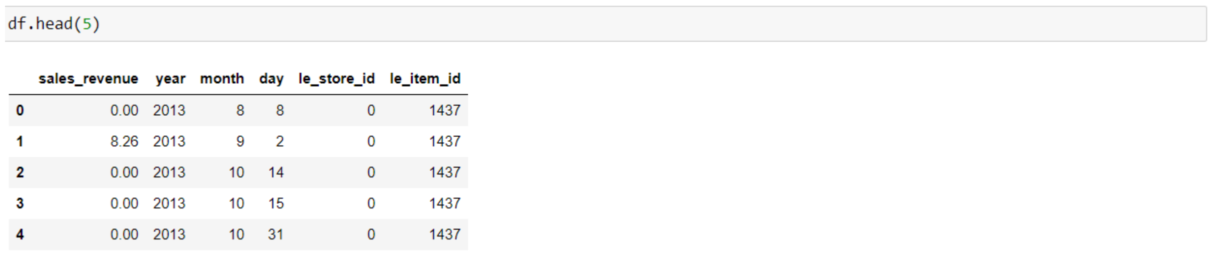
	Unnamed: 0	store_id	item_id	wm_yr_wk	sell_price	date	d	event_name	event_type	id	dept_id	cat_id
0	0	CA_1	HOBBIES_1_001	11328	8.26	2013-08-08	923	Eid al-Fitr	Religious	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1	HOBBIES
1	1	CA_1	HOBBIES_1_001	11332	8.26	2013-09-02	948	LaborDay	National	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1	HOBBIES
2	2	CA_1	HOBBIES_1_001	11338	8.26	2013-10-14	990	ColumbusDay	National	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1	HOBBIES
3	3	CA_1	HOBBIES_1_001	11338	8.26	2013-10-15	991	EidAlAdha	Religious	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1	HOBBIES
4	4	CA_1	HOBBIES_1_001	11340	8.26	2013-10-31	1007	Halloween	Cultural	HOBBIES_1_001_CA_1_evaluation	HOBBIES_1	HOBBIES

As we can see in the combined dataset, all the dataset is merged and now we have a total 1959980 records present in the combined dataset.

Step 13: I have created a new dataset as ‘df’ which will contain the relevant information from the combined dataset such as store_id, item_id, date and sales_revenue.

Step 14: Now, the dataset is ready for preprocessing and modelling, I have first performed the Label Encoding on the categorical dataset and converted the date into day, month and year format.

The final dataset is in the form as:



```
df.head(5)
```

	sales_revenue	year	month	day	le_store_id	le_item_id
0	0.00	2013	8	8	0	1437
1	8.26	2013	9	2	0	1437
2	0.00	2013	10	14	0	1437
3	0.00	2013	10	15	0	1437
4	0.00	2013	10	31	0	1437

Now, the data preprocessing is done and is ready for data modelling.

Modelling:

Approaches performed on the Predictive Model

Approach 1: Linear Regression Model

After the data preparation, I have performed the Linear Regression model on the final dataset. I have chosen Linear Regression because the final dataset is in continuous data format, so I can't choose the Classification model. I have performed the Linear Regression Model but its accuracy is 0.008 and the MSE which I have recorded is 86.28 which shows that the Linear Regression model does not fit the dataset.

Approach 2: Random Forest Regressor Model

I have then performed a Random Forest Regressor model on the final dataset which gives me quite good results. I am able to achieve 0.46 accuracy from the model and the MSE which I have recorded was quite low which was 51.14.

Approach 3: XGBoost Model

While there was a chance of improvement, So, I thought of implementing one more machine learning model as XGBoost. The result achieved from this model was not quite good as compared to the Random Forest model. I have achieved the model accuracy as 0.04 and MSE value as 90.97.

Approach 4: K-Nearest-Regressor Model

I have performed one more model on the given dataset as K-nearest-neighbour(KNN) model which gave me a better result than XGBoost but as compared to the Random Forest model, it was quite low. I have achieved the model accuracy as 0.15 and MSE value as 80.34 which was not suitable as per the business requirement.

Approach 5: Random Forest Regressor with Outliers Removal

After performing 4 models, I have chosen the Random Forest model as my final model and planned to perform some other preprocessing steps on the model.
I have removed the outliers from the dataset by using the IsolationForest model which helps to identify the outliers in the dataset and can help to remove the outliers. Following the

removal of outliers in the dataset, I have again performed the Random Forest model and found the MSE value has reduced considerably and now the MSE value is 17.12 only. However, the model accuracy is still low as compared to the previous Random Forest model accuracy which is 0.24.

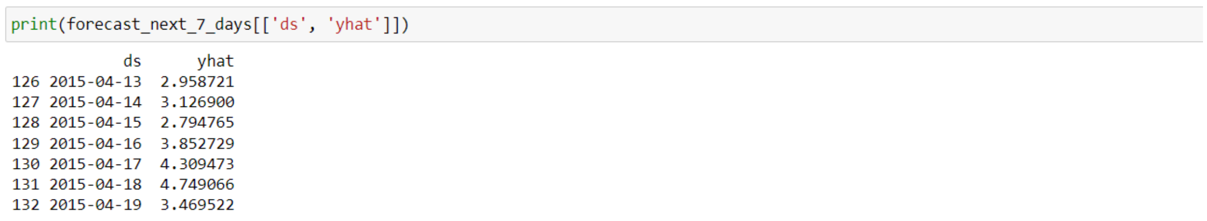
Final Predictive Model: Random Forest Regressor Model with Outliers Value

After following all these approaches, I have finalised the Random Forest model by considering the outliers in the dataset as it gave me the best result in the model experiments.

Random Forest model with the hyperparameters used:
I have used the n_estimators as 100 and random_state as 42 for the models to perform. Here is the command which I have used for the modelling.
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

Forecasting Model: Prophet Time-Series Model

For forecasting the sales revenue of all the stores for the next 7 days, I have used a time-series machine learning model as Prophet which is able to predict the sales revenue for next 7 days. The results achieved by the model is:



```
print(forecast_next_7_days[['ds', 'yhat']])
```

```
            ds      yhat
126 2015-04-13  2.958721
127 2015-04-14  3.126900
128 2015-04-15  2.794765
129 2015-04-16  3.852729
130 2015-04-17  4.309473
131 2015-04-18  4.749066
132 2015-04-19  3.469522
```

Evaluation:

- ### Evaluation Metrics:
  I have performed MSE to validate the model performance and also I have used model score to validate the accuracy of the model. These are the two key factors to determine the model performance.

- ### Results and Analysis:
  I have finalised the Random Forest model to predict the sales revenue for the given item, store and date and has given me the accuracy as 0.46 and MSE value as 51.14. This model is able to predict the sales revenue for the given inputs. Also, this model is able to predict the test dataset given. Also, I have finalised the Prophet model which is able to forecast the sales revenue for the next 7 days.
  While performing this experiment, I gained the insights on how to work on a big dataset and can perform data preparation in chunks of data. Also, I have learnt how different models can vary the result of the model. I have performed 4 models and all are giving a different range of results. For the first time, I have worked on the time

series model and I have learnt how the world predicts the future analysis on the basis of current data.

- **Business Impact and Benefits:**
  As the model is able to predict the sales revenue, the business team can analyse the performance of stores along with the item categories present in the store. This experiment can help the business team plan their future blueprints on the sales and predict the revenue from the store. By using the machine learning model as an api service, it is accessible and easy to use. All they need is to provide the input to the model for the dates, store and item they are looking for and they will receive the result from the api.

- **Data Privacy and Ethical Concerns:**
  The data has been provided for educational purposes and has nothing to do with the real-time ethical issues. The experiment performed and deployed on the git and heroku which are private to use and secured with the authentication. The api service provided is public and does not contain any kind of data information.

Deployment:

This machine learning model is deployed on the Heroku server by using the FastApi. This will help the business team to use the models in a more efficient way. All they need to do is to pass the input parameters into the api and the team will get the result.

There are two apis created:
- **Predict_revenue:** which will predict the sales revenue on the basis of store_id, item_id and the date. These three will be the input parameters and the predicted sales_revenue will be received as the result.
- **Forecast_revenue_next_7_days:** This method will forecast the sales revenue for the next 7 days predicted from the model.

I have published the experiment model on the github and the heroku for the deployment.
Below are the details to access the model.
Git link: https://github.com/tarungupta293/Sales-Revenue-Prediction
API link for Predict_revenue:
https://demo-testing-1999.herokuapp.com/models/predictive/predicting_sales_revenue.py
https://demo-testing-1999.herokuapp.com/models/forecasting/forecasting_next_7_days.py

Conclusion:

While working on the experiment, I have found many interesting insights to learn. I have worked on a time-series model for the first time and learnt how it works and how forecasting works in real-time dataset. I have worked on the real-time big dataset and merged the dataset into chunks. In this experiment, I have performed many models and based on their performance, I have learnt which parameters should be considered while deciding the best model. I have also learnt about how machine learning models can be used as a service api and can be deployed for production in order to use for the business team. For the first time, I have worked on the service apis and got an insight how business teams who don't have so much technical knowledge generally work on the machine learning models.

References:

1. https://medium.com/@ravikumar10593/end-to-end-machine-learning-model-deployment-using-fast-api-and-heroku-part-1-ace8657d7719#:~:text=Here%20in%20this%20article%2C%20we%20will%20discuss%20end-to-end,applications%20entirely%20in%20the%20cloud%29%20Code%20links%3A%20https%3A%2F%2Fgithub.com%2FRavikumar10593-hub%2FFastAPI_depoloyment_to_heroku
2. https://fastapi.tiangolo.com/
3. https://medium.com/@akhandsingh1010/fastapi-installation-and-setup-on-windows-machine-4721046740a6
4. https://blog.back4app.com/deploy-fastapi/
5. https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/
6. https://www.kaggle.com/code/prashant111/tutorial-time-series-forecasting-with-prophet
7. https://practicaldatascience.co.uk/machine-learning/how-to-use-the-isolation-forest-model-for-outlier-detection