Deloitte USI AI Academy

Capstone Project

Store Sales

Batch: B
Group: 5
Date: 24/03/2022

Members:
Sai Teja Burla
Vishnu S
Spandana Y R
Yashwanth K
Thathireddy Vishnuvardhan Reddy
Tilak Raj

01 Problem Statement

02 Data Analysis

03 Visualization

04 Data Pre-Processing

05 ML Model Development and Evaluation

06 PySpark and AWS

07 Model Deployment and Demonstration

Store Sales Forecast

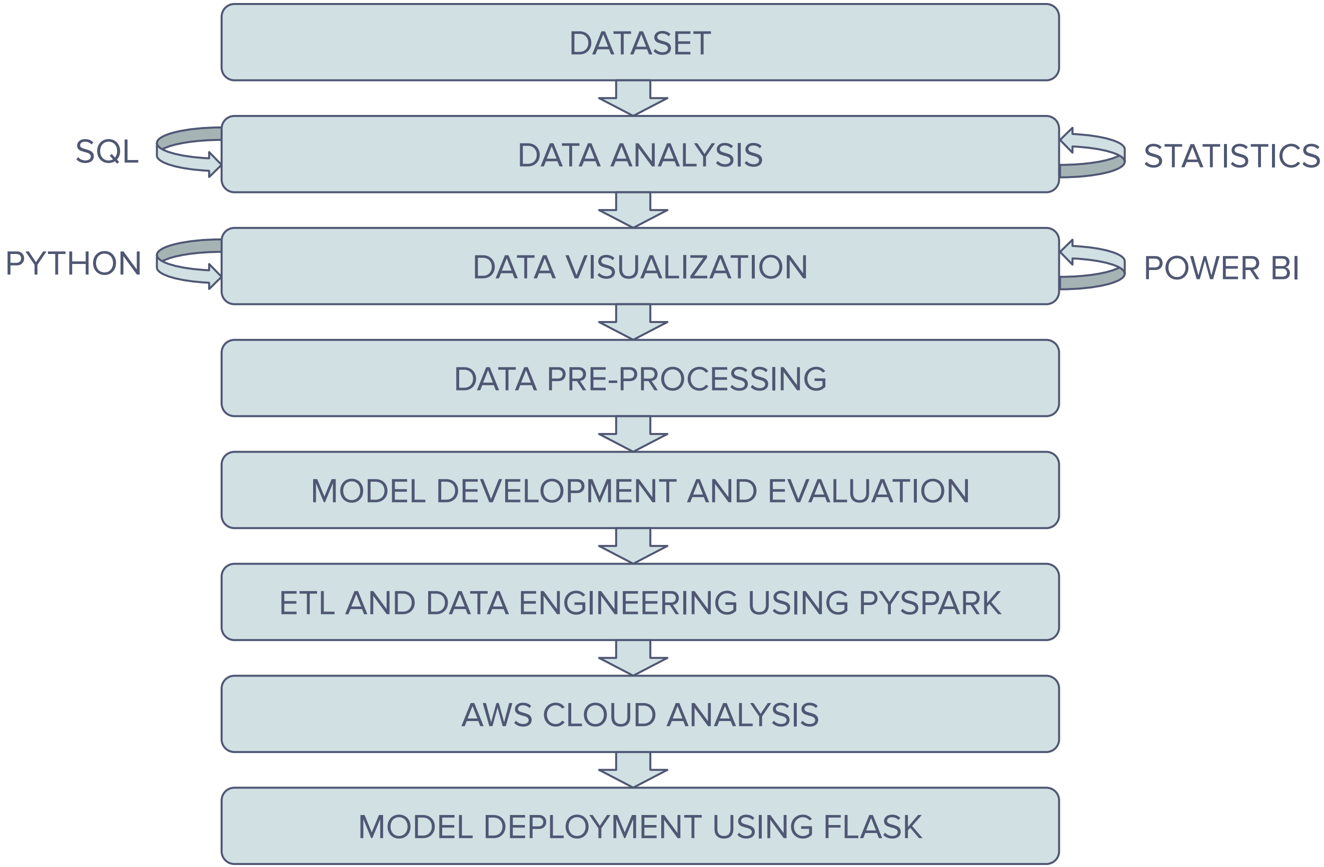
Problem Statement

Dean’s is a large web e-trailer chain which sells different types of products nationally. It has shopping services via different modes/types of transportation. It keeps track of order details, shipping details, discounts, provided to the customer, profit earned etc.

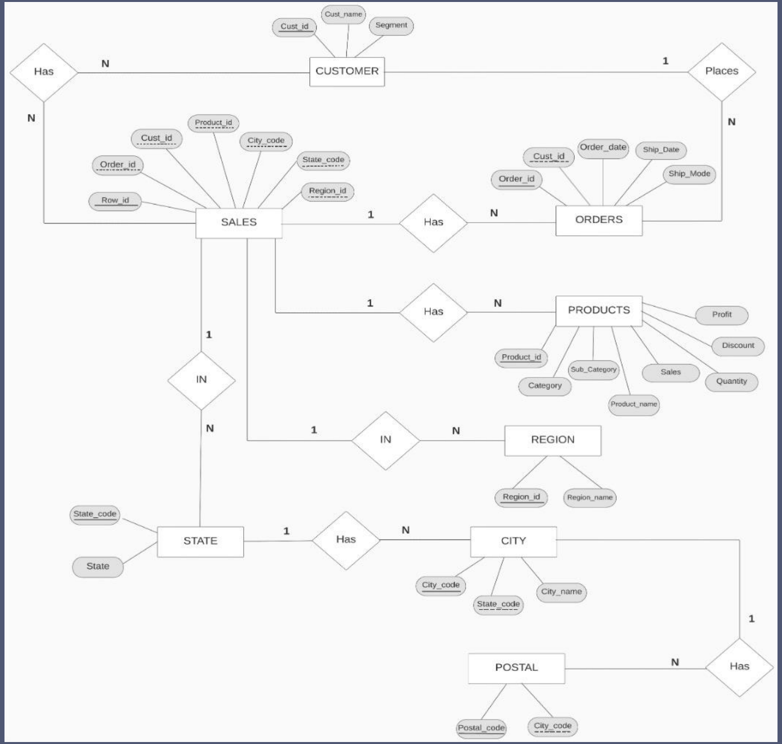
Solution

- To preprocess and aggregate the data into monthly sales
- To identify various factors in the data such as trend, seasonality and stationarity, etc
- To build various forecasting models for the time series data for forecasting sales in future
- Comparing all the forecasting models based on RMSE value to find the best one

IMPLEMENTATION



Data Analysis Using SQL



ER DIAGRAM

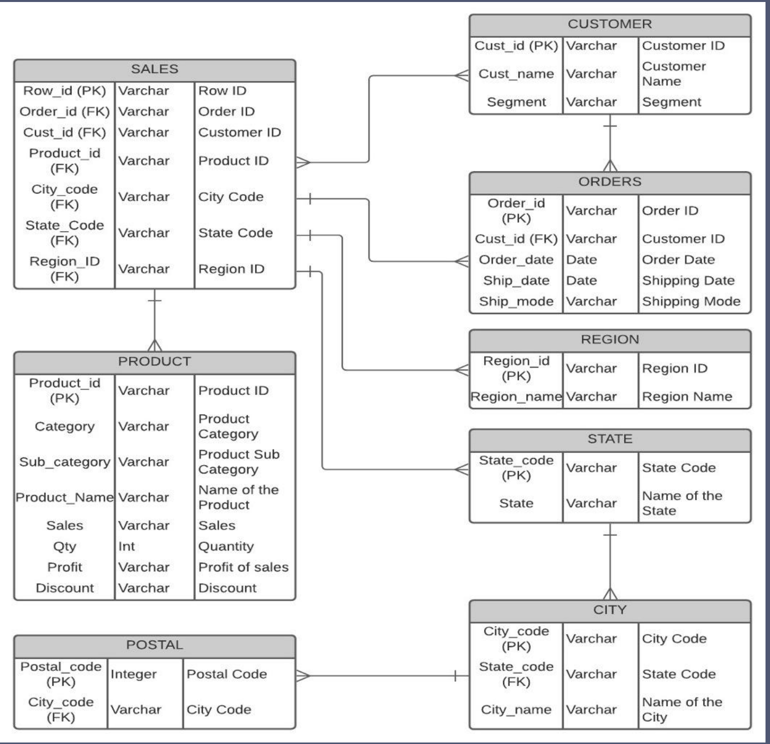


TABLE DESIGN

Data Analysis Using Python

```
Number of Rows in the Dataset:  9985
Out[5]:
```

	Row_ID	Postal_Code	Sales	Quantity	Discount	Profit	Duration	Order_Year
count	9985.000000	9985.00000	9985.000000	9985.000000	9985.000000	9985.000000	9985	9985.000000
mean	4993.896545	55182.79359	229.636100	3.789685	0.156183	28.587722	3 days 23:01:09.584376564	2012.723986
std	2883.894709	32060.89504	622.927104	2.225074	0.206477	234.183523	1 days 17:55:50.881956015	1.123722
min	1.000000	1040.00000	0.444000	1.000000	0.000000	-6599.978000	0 days 00:00:00	2011.000000
25%	2497.000000	23223.00000	17.280000	2.000000	0.000000	1.731000	3 days 00:00:00	2012.000000
50%	4993.000000	56301.00000	54.500000	3.000000	0.200000	8.671500	4 days 00:00:00	2013.000000
75%	7489.000000	90008.00000	209.940000	5.000000	0.200000	29.364000	5 days 00:00:00	2014.000000
max	9994.000000	99301.00000	22638.480000	14.000000	0.800000	8399.976000	7 days 00:00:00	2014.000000

```
dfm['Category'].value_counts()
```

```
Office Supplies     6020
Furniture           2120
Technology          1845
Name: Category, dtype: int64
```

```
dfm['Segment'].value_counts()
```

```
Consumer         5188
Corporate        3016
Home Office      1781
Name: Segment, dtype: int64
```

```
Day = dfm.groupby(['Ship_Mode','WeekDay']).count()['Row_ID']
Day
```

```
Ship_Mode       WeekDay
First Class     Friday        248
                Monday        135
                Saturday      290
                Sunday        274
                Thursday      256
                Tuesday       147
                Wednesday     184
Same Day        Friday         74
                Monday        110
                Saturday       47
                Sunday         13
                Thursday      118
                Tuesday       102
                Wednesday      76
Second Class    Friday        262
                Monday        274
                Saturday      332
                Sunday        392
                Thursday      208
                Tuesday       198
                Wednesday     272
Standard Class  Friday        711
                Monday        976
                Saturday      948
                Sunday        846
                Thursday      550
                Tuesday      1090
                Wednesday     852
Name: Row_ID, dtype: int64
```

Data Analysis Using Python

```
dfm['Region_Name'].value_counts()
```

```
West       3199
East       2845
Central    2321
South      1620
Name: Region_Name, dtype: int64
```

```
State_Count = dfm.loc[dfm['Region_Name'] == 'West', 'State']
State_Count.value_counts()
```

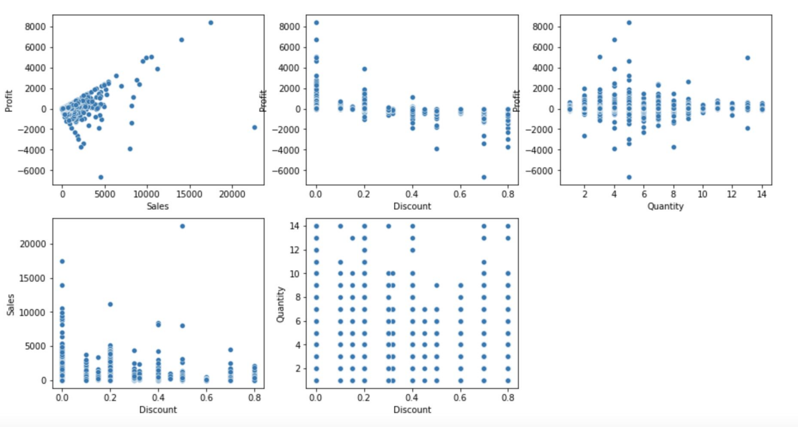
```
California    1998
Washington     506
Arizona        224
Colorado       182
Oregon         124
Utah            53
Nevada          39
New Mexico      36
Idaho           21
Montana         15
Wyoming          1
Name: State, dtype: int64
```

```
City_Count = dfm.loc[dfm['State'] == 'California', 'City_Name']
City_Count.value_counts().head(10)
```

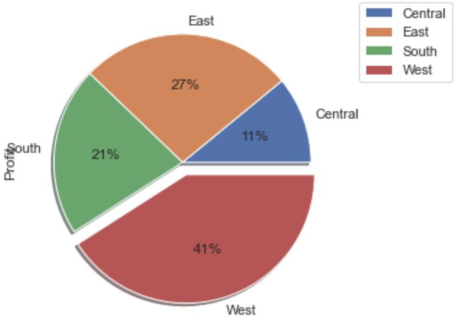
```
Los Angeles      747
San Francisco    509
San Diego        170
San Jose          42
Long Beach        27
Anaheim           26
Oakland           26
Fresno            25
Pasadena          25
Westminster       17
Name: City_Name, dtype: int64
```

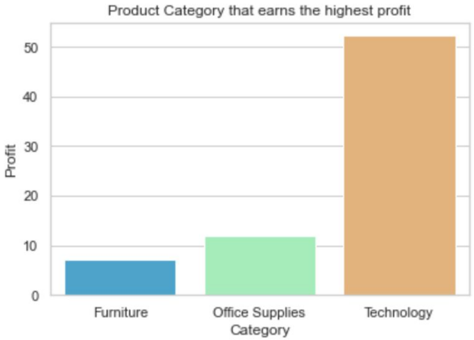
Hypothesis Testing:
Anova
Chi-Square
T-Test(One Sample)
T-Test(Two Sample)
T-Test(Paired Sample)
Correlation Test

```
<AxesSubplot:xlabel='Discount', ylabel='Quantity'>
```

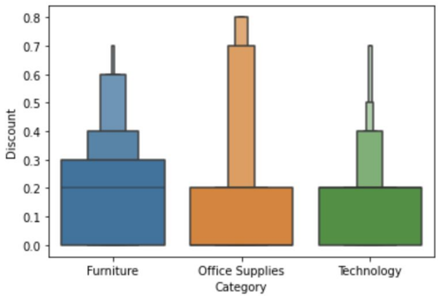


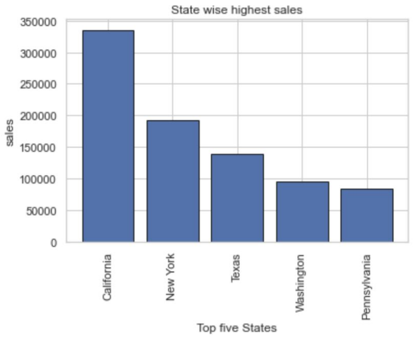
Data Visualization Using Python



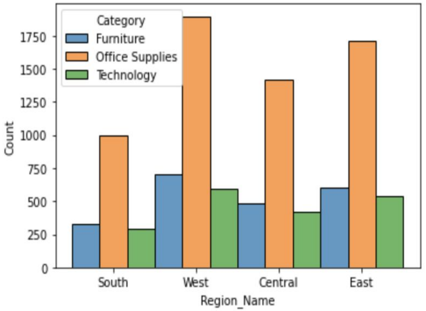


Out[26]: <AxesSubplot:xlabel='Category', ylabel='Discount'>

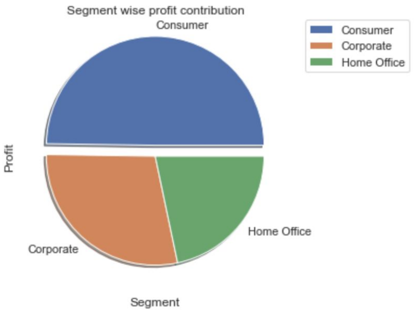




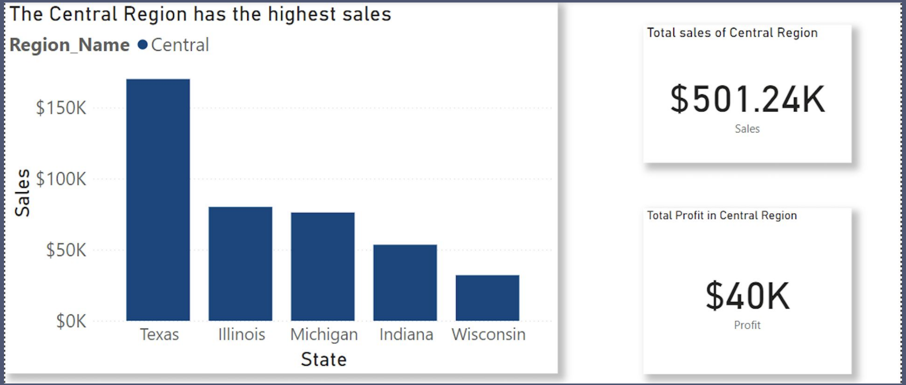
Out[17]: <AxesSubplot:xlabel='Region_Name', ylabel='Count'>

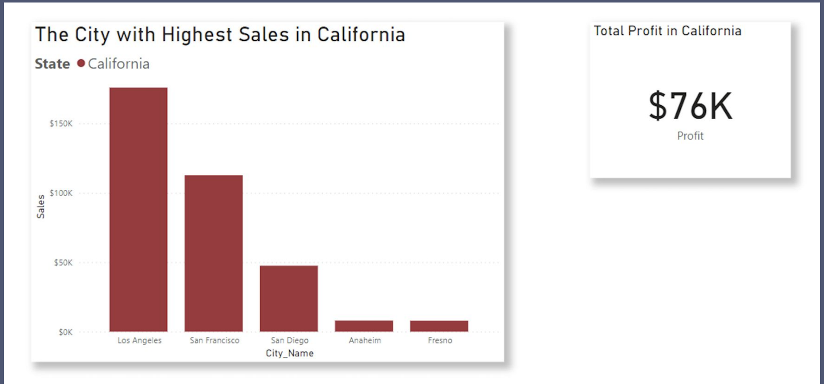


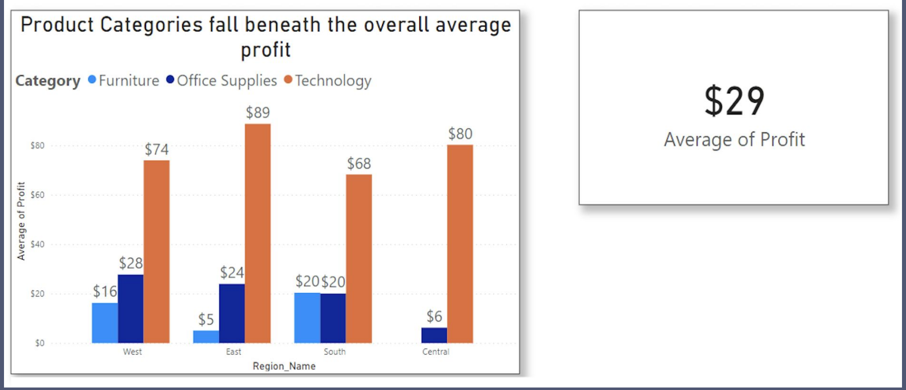
Out[9]: <matplotlib.legend.Legend at 0x2655bd187f0>

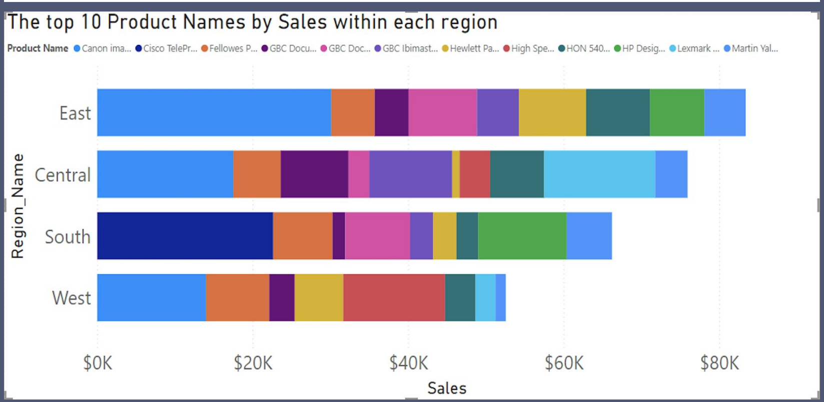


Data Visualization Using PowerBI

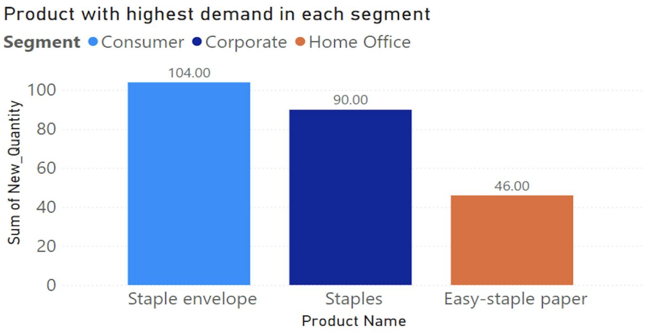


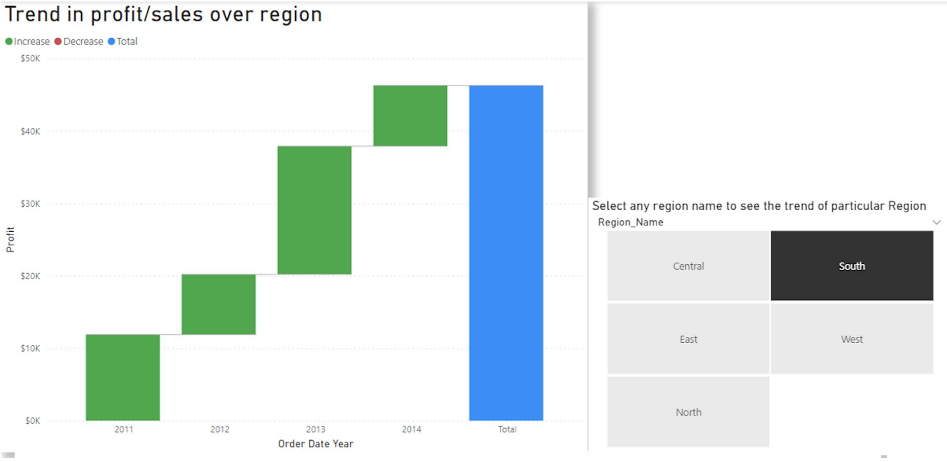






Data Visualization Using PowerBI





Data Preprocessing

```
Row ID              0
Order ID            0
Order Date          0
Ship Date           0
Ship Mode          22
Customer ID         0
Customer Name       0
Segment             9
Country             6
City_Code           0
State_Code          0
Postal Code         0
Product ID          0
Category            0
Sub-Category        0
Product Name        0
Sales               0
Quantity            0
Discount            0
Profit              0
dtype: int64
```

```
Row ID              0
Order ID            0
Order Date          0
Ship Date           0
Ship Mode           0
Customer ID         0
Customer Name       0
Segment             0
Country             0
City_Code           0
State_Code          0
Postal Code         0
Product ID          0
Category            0
Sub-Category        0
Product Name        0
Sales               0
Quantity            0
Discount            0
Profit              0
dtype: int64
```

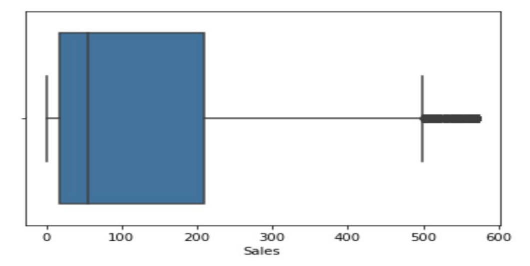
Data pre-processing steps are normally used to convert the raw data into clean data set that can be used for training the model.

- Null Value Detection and Treatment
- Outlier Analysis and Treatment
- Data Encoding
- Data Scaling and Transformations
- Feature Engineering
- Splitting into Train and Test

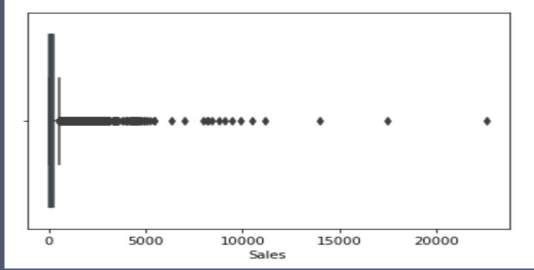
```
timeSeriesDF = df[['Order Date','Sales']]
timeSeriesDF.columns = timeSeriesDF.columns.str.replace(' ','_')
timeSeriesDF['Order_Date'] = timeSeriesDF['Order_Date'].str.replace('-','/')
timeSeriesDF['Order_Date'] = pd.to_datetime(timeSeriesDF['Order_Date'])
timeSeriesDF.head()
```

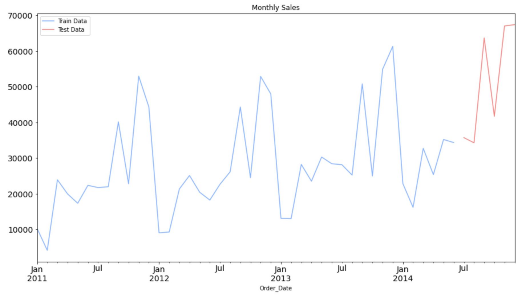
	Order_Date	Sales
0	2013-11-09	261.9600
1	2013-11-09	731.9400
2	2013-06-13	14.6200
3	2012-10-11	957.5775
4	2012-10-11	22.3680

<AxesSubplot:xlabel='Sales'>



<AxesSubplot:xlabel='Sales'>





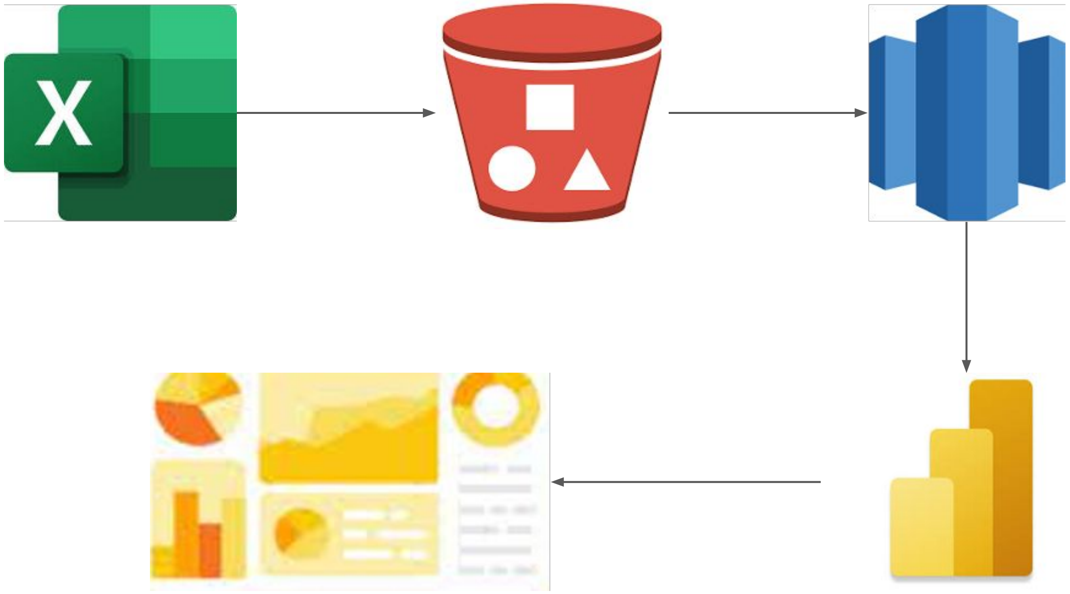
ML Part

ETL and Data Engineering Using PySpark

- Load data from local file system to Hadoop Cluster using Hive Table
- Using Spark Session, load the data from Hive table to the Spark DataFrame
- Null Value Identification and Outlier Analysis and treatment of both using Spark
- Data Profiling using Spark
- Execute SQL Commands using Spark
- Save the data in parquet form for later use, using Spark

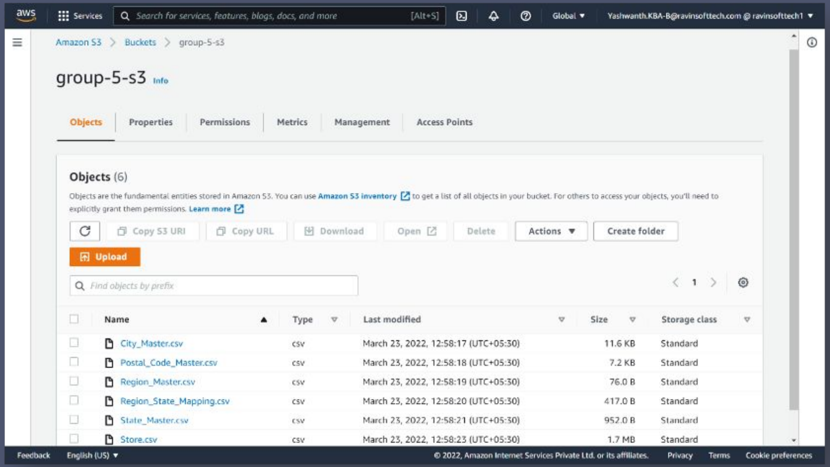
AWS Cloud Analysis

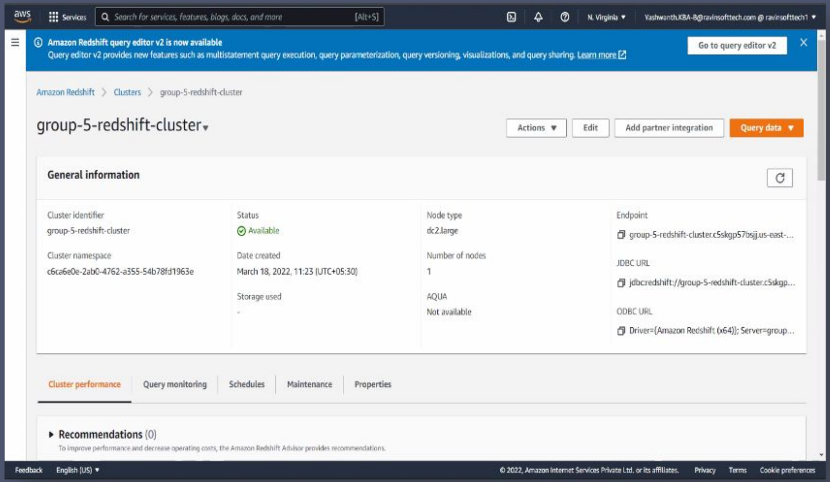
- Moving the datasets to AWS S3
- Creating Redshift Instance
- Creating tables in Redshift
- Creating a pipeline/copy command to move the data from S3(storage) to Redshift Table
- Connecting Redshift data to PowerBI
- Visualization in PowerBI



AWS Cloud Analysis

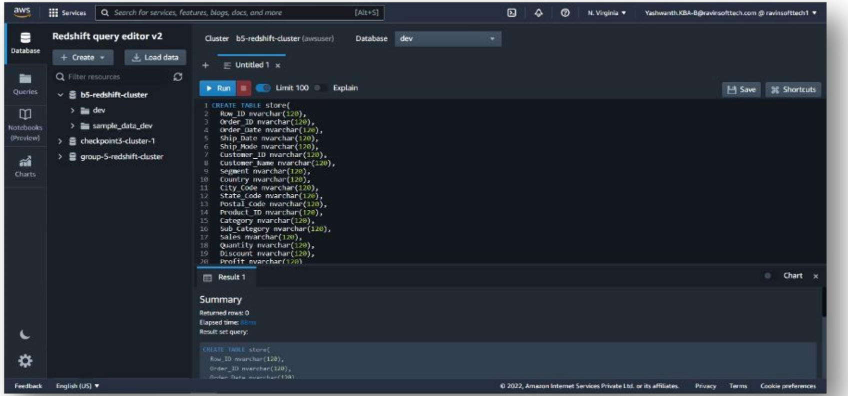
We made an AWS S3 Bucket and moved all the datasets into the same and we also created a Redshift Instance.

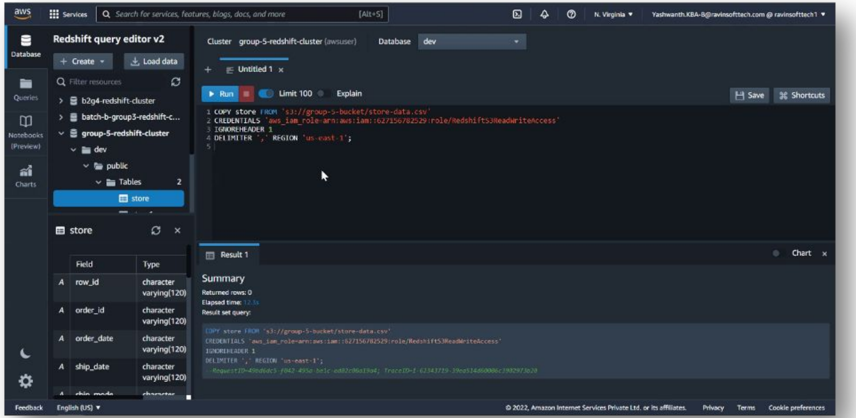




AWS Cloud Analysis

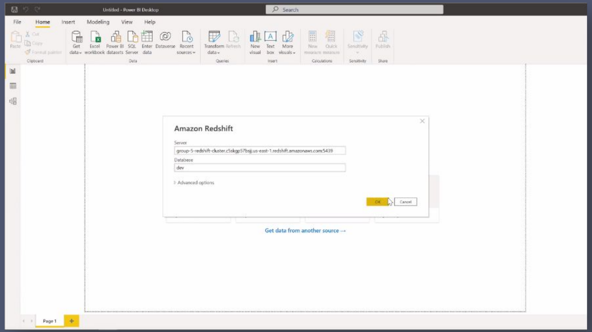
We then made tables in Redshift and moved the data from S3 into the created tables using COPY command.

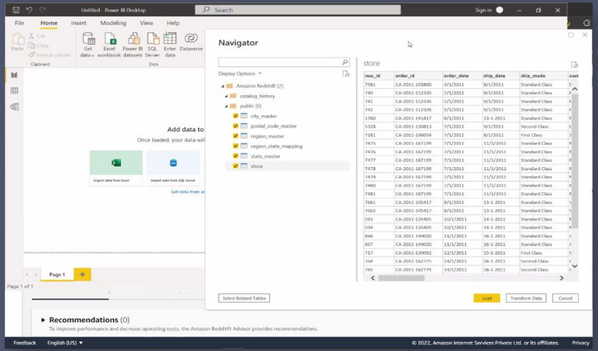


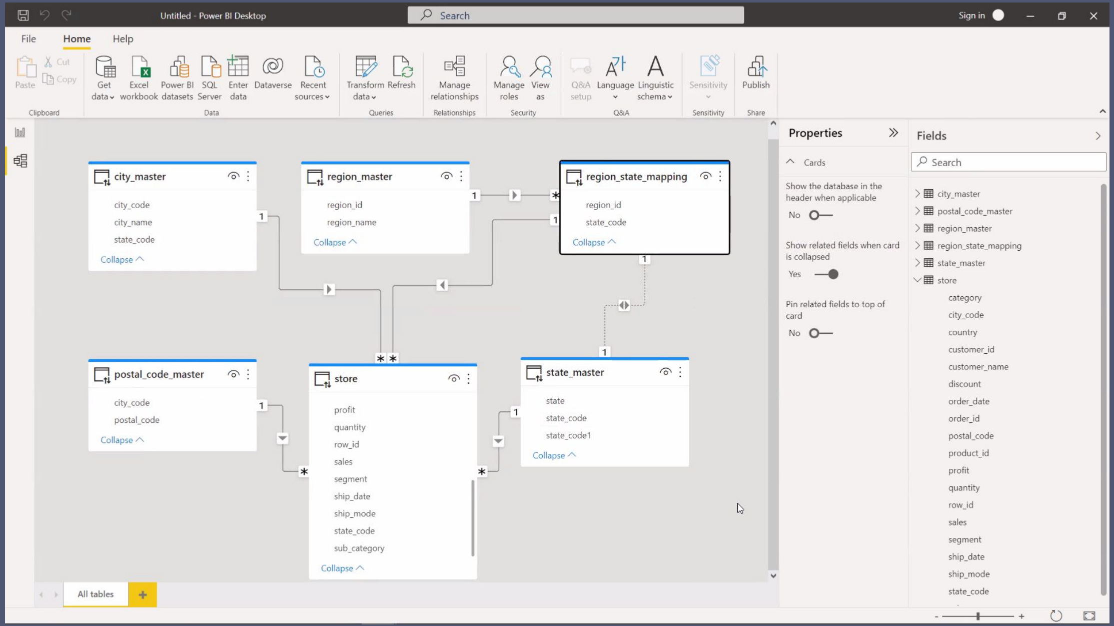


AWS Cloud Analysis

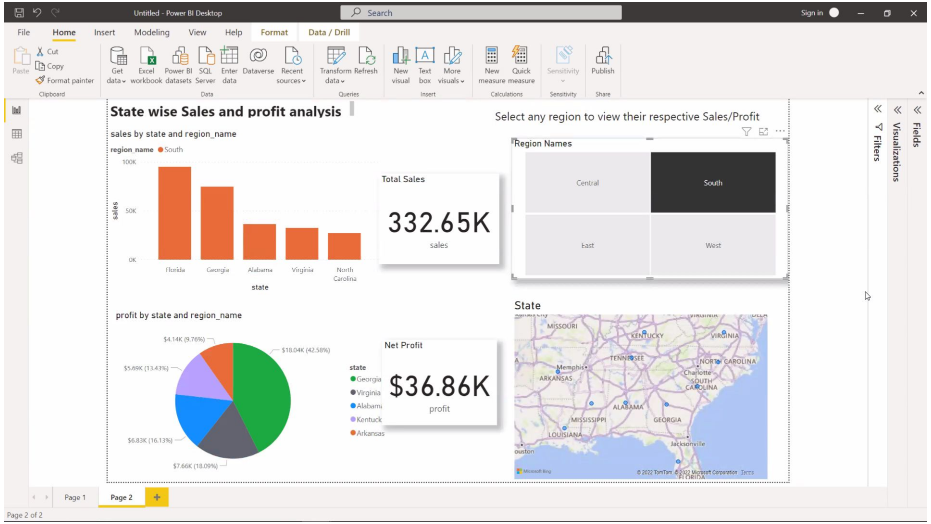
We then connected Redshift to PowerBI and checked the relationship between all the tables created

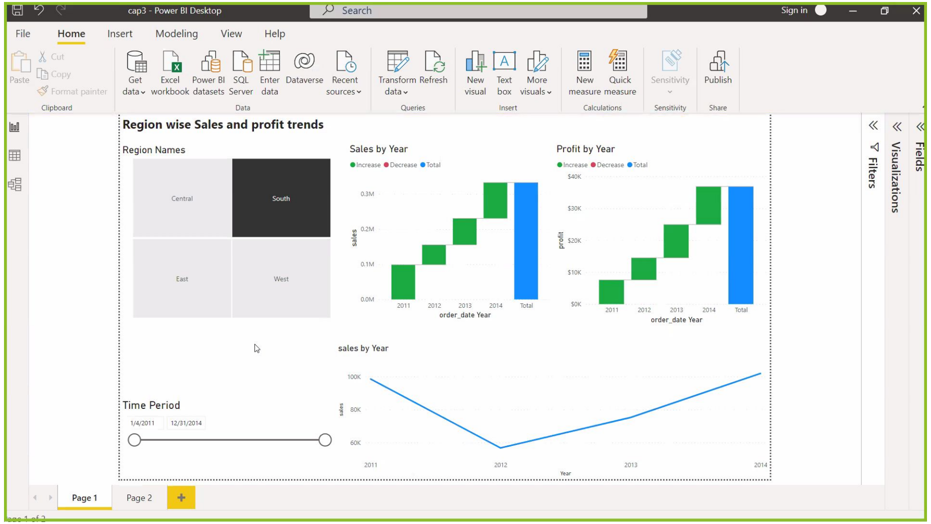




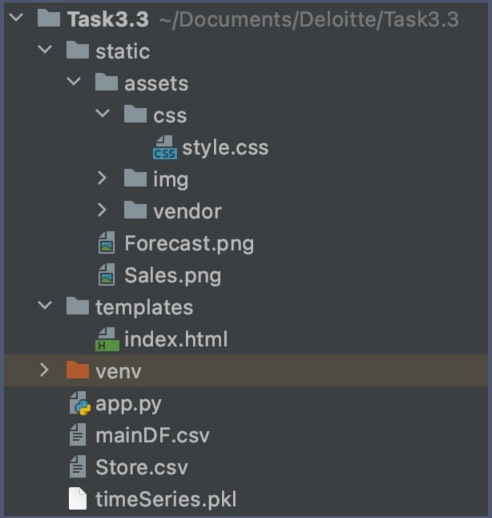


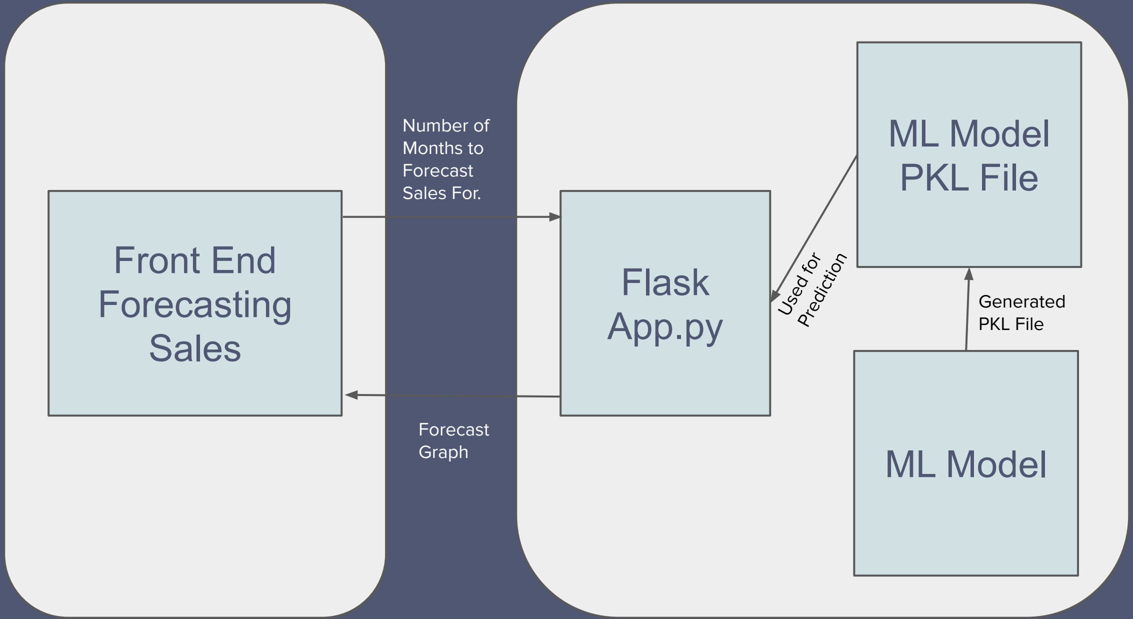
AWS Cloud Analysis Power BI Dashboard



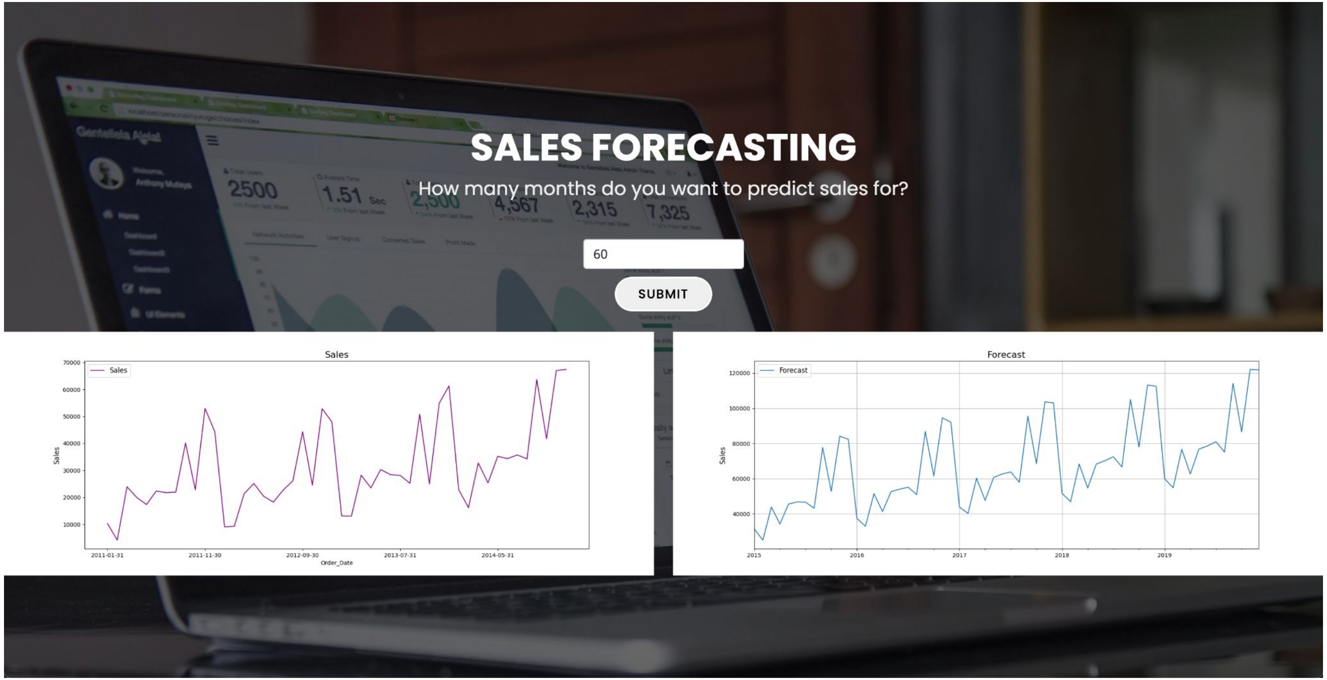


ML Model Deployment Using Flask Architecture





Deployment and DEMO



Thank You!!!