Business Intelligence for Decision Support

Course ID - 32567

Assignment 3

By

Ankit Kapoor - 12982233

Shamsheer Verma - 12851248

Avinash Kumar Pakanati - 12723514

Table of Contents

Introduction

The grocery retailer Corporacion Favorita was founded in 1957 and is based in Quito, Ecuador. As one of the three biggest companies in the country with over 7000 employees and profits of 135 million USD a year , it operates stores at more than 300 locations in six countries across South America. The company intends to improve its stock management through the implementation of business intelligence.

The demand of any product sold in retail is subject to variation and meeting the customer's expectations is crucial to the success of any retailer. Retail of groceries, however, naturally comes with a series of special challenges that are unique to this particular class of products. The most important of the problematic aspects comes with perishable goods, that lose their value if not sold within a certain timespan. Therefore a prediction of upcoming changes in demand are an elementary need for these businesses. Was the estimation too high, perishable products that have been stocked and could not be sold, will expire and cause a loss to the business. In the contrary case, stores will run out of certain products, which not only means a loss in potential sales, but also a drop in customer satisfaction. The base of the predictions currently made at Corporacion Favorita, is mainly experience and the methods used are highly subjective.

To improve the predictions and optimize the stock, they look to implement a tool to use data analysis and possibly machine learning, which allows them to back up their decisions on actual data. business intelligence system in this process will be to mitigate this problem of poor .

The data analysis is based on results of prediction of unit sales of n number of product sold at different store locations. Analysis is performed using the two datasets Train and Test data set. Train dataset includes store number, date and item number, onpromotion and unit sales as its attribute. Test dataset includes number of products which are not included in training dataset, based on the similar products from training dataset, new product sales has to predicted. It uses onpromotion information along with store number, item number for the accurate prediction. (*Corporación Favorita Grocery Sales Forecasting* 2017)

Business Problem

The actual business problem faced by Corporacion Favorita is that they don’t have any accurate model for sales forecasting. The brick & mortar stores experience the problem with sales and purchasing forecasting. While predicting the product sales, if they predict more number of units, chances are that products gets overstocked leading to accumulation of perishable goods and lesser prediction of products will lead to items quickly selling out and being out of stock. This problem indirectly affects financial outcomes of the organizations as overstock would cause loss over the products stored unnecessarily and on the other hand, out of stock would cause customers returning empty handed. (*Corporación Favorita Grocery Sales Forecasting* 2017)

BI Project Objective

The main project objective is to create a model for Corporacion Favorita Grocery retailer. This model will allow the organization to forecast the product sales accurately. At present, they are dependent on forecasting which are based on subjective methods. These subjective methods do not provide required accuracy rate for the prediction. The model would be created with the help of machine learning techniques, ensuring that the customers get enough of products they require. (*Corporación Favorita Grocery Sales Forecasting* 2017)

Framework

1. Framework for Development:

   The framework for our development is shown in the following figure. The dataset is imported from kaggle and transferred to postgres. Using python as our backend we perform data extraction, data handling, perform pre-processing and exploration of the data and finally analyse to predict trends by the application of modelling techniques. In addition to this, we have used python as a Front-end for our Graphical User Interface where the management of the company can pull data about specific stores and products by date as desired.

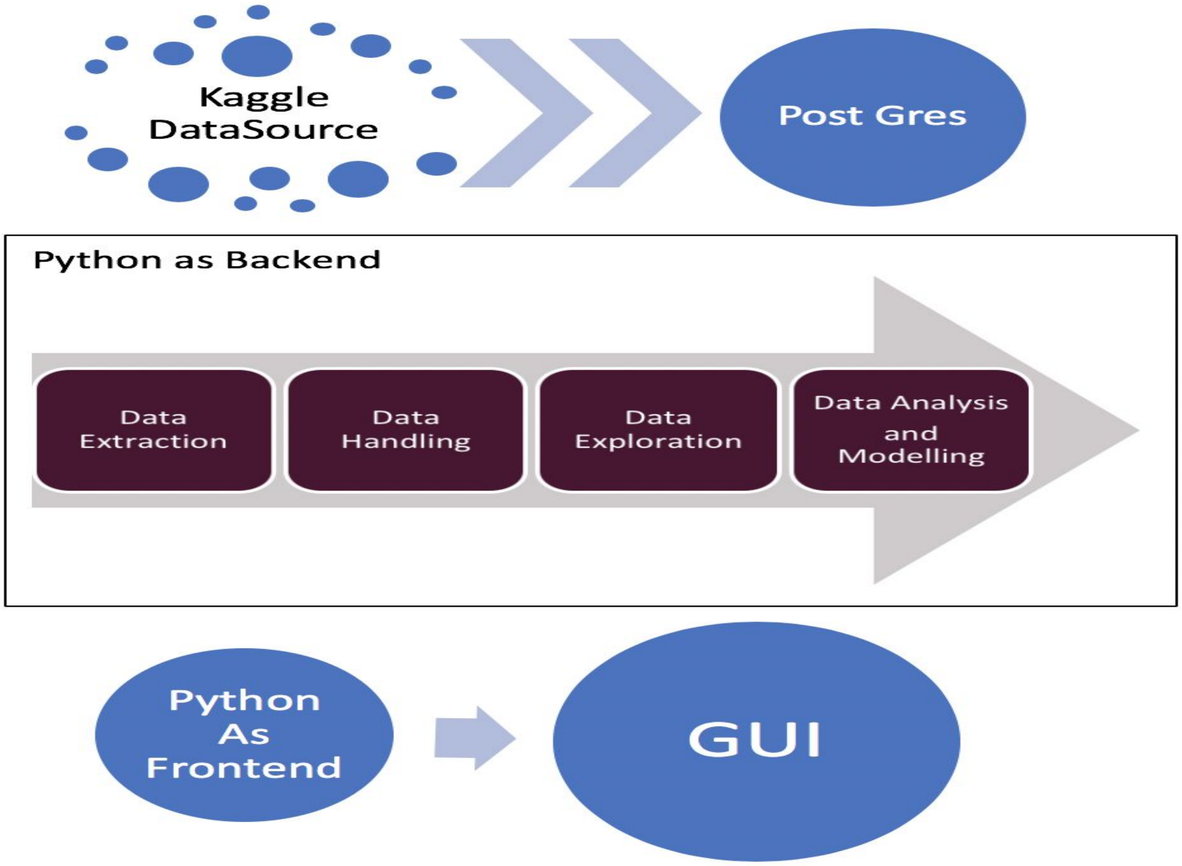


Fig 1. Development Framework

2. Framework for execution

In the execution framework as shown in figure 2, we have the GUI, Python as a backend providing application program interface to extract the data from the database and then run the model for analysis. The Graphical User Interface shown in figure 3, is used as the Sales Forecasting tool by the management of Corporation Favorita. Here the data can be filtered out according to the need by entering the Shop ID, Product ID, Date and promotion details. In the next step, Python is used in the backend as shown in figure 4, to extract the sales data from the database shown in figure 5, to further use for model training.

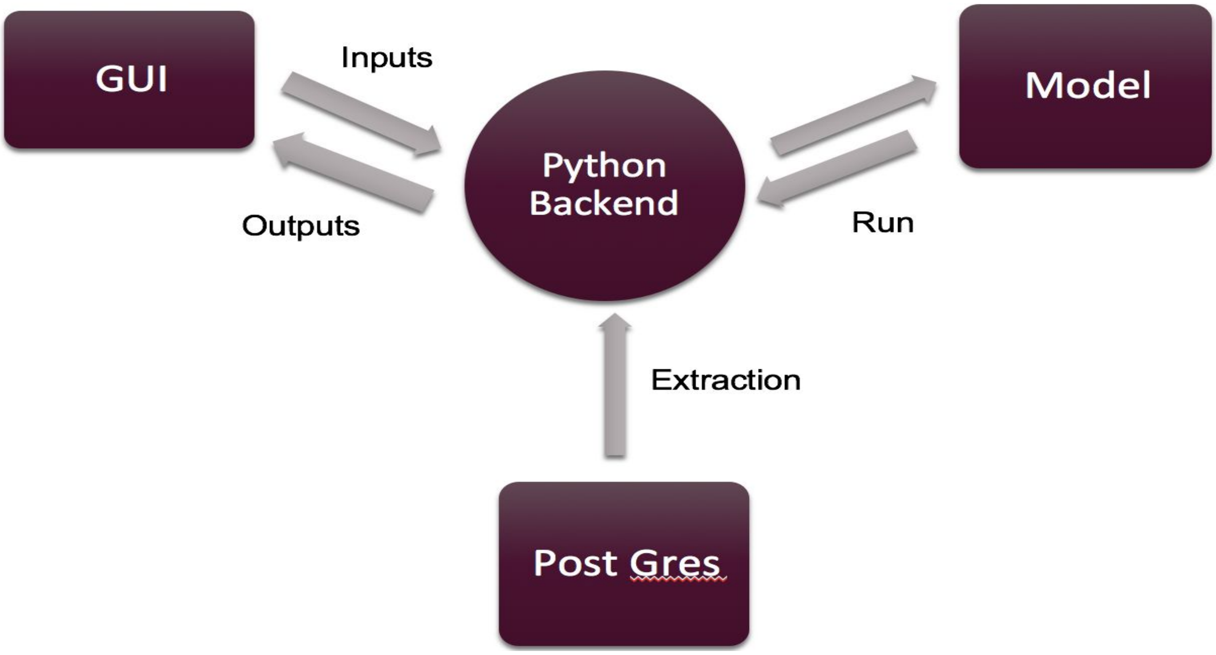
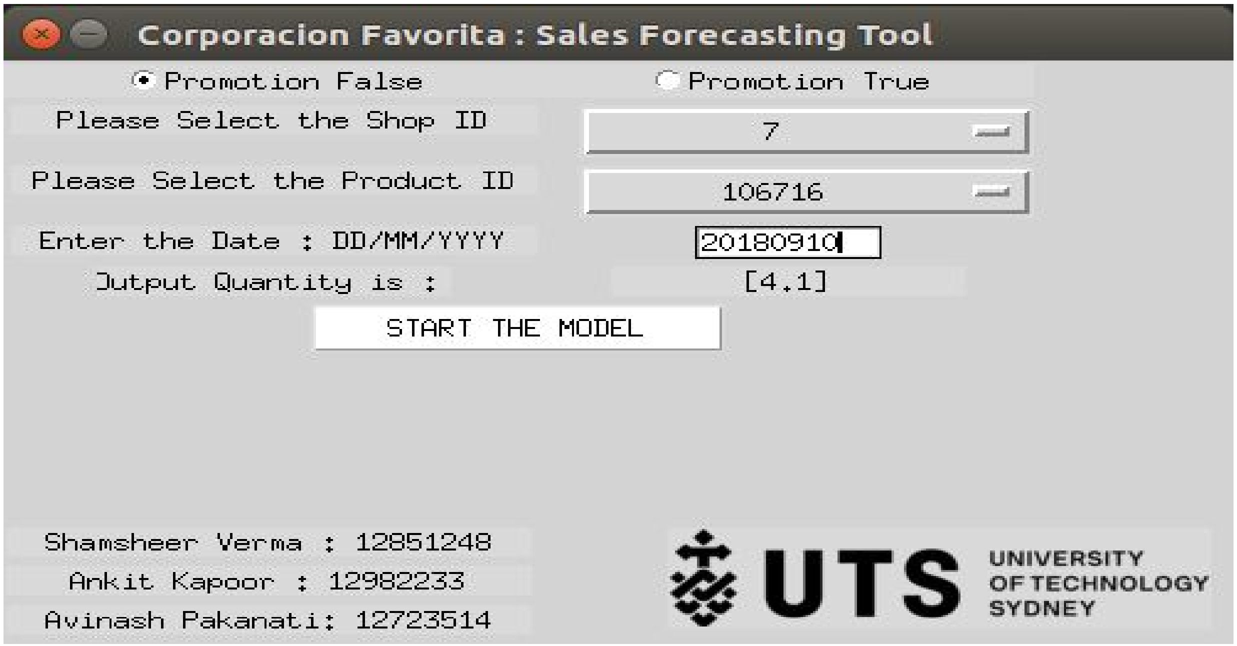


Fig 2. Execution Framework



● Promotion False ○ Promotion True

Please Select the Shop ID: 7

Please Select the Product ID: 106716

Enter the Date : DD/MM/YYYY: 20180910

Output Quantity is : [4.1]

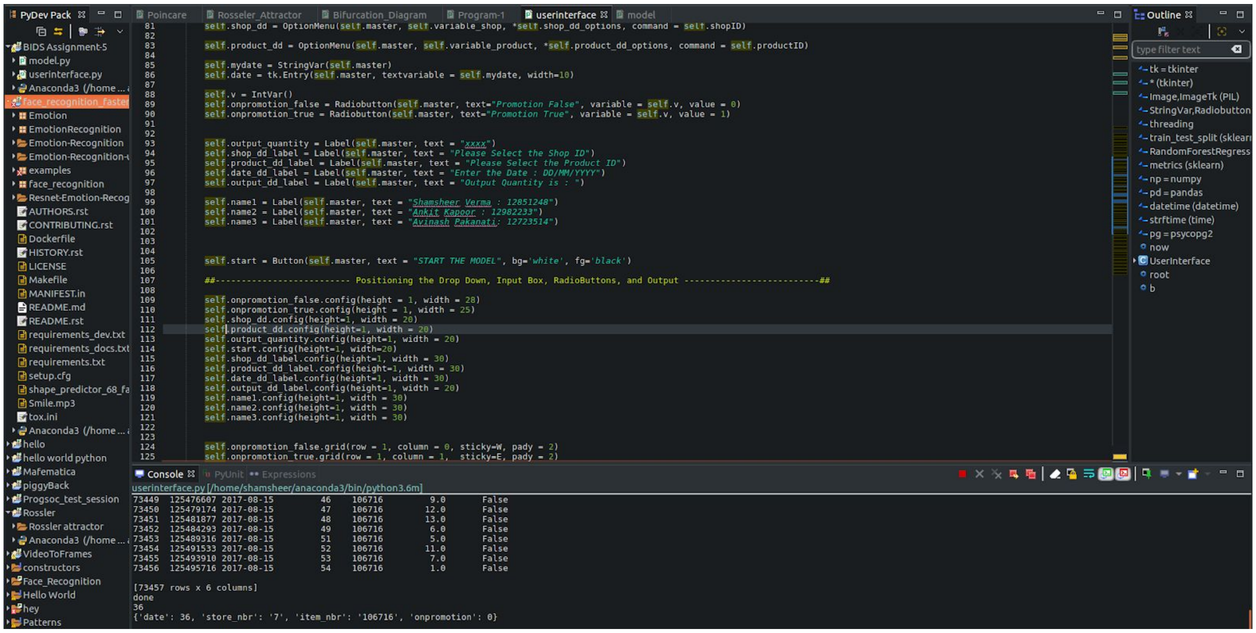
Shamsheer Verma : 12851248

Ankit Kapoor : 12982233

Avinash Pakanati: 12723514

UTS UNIVERSITY OF TECHNOLOGY SYDNEY

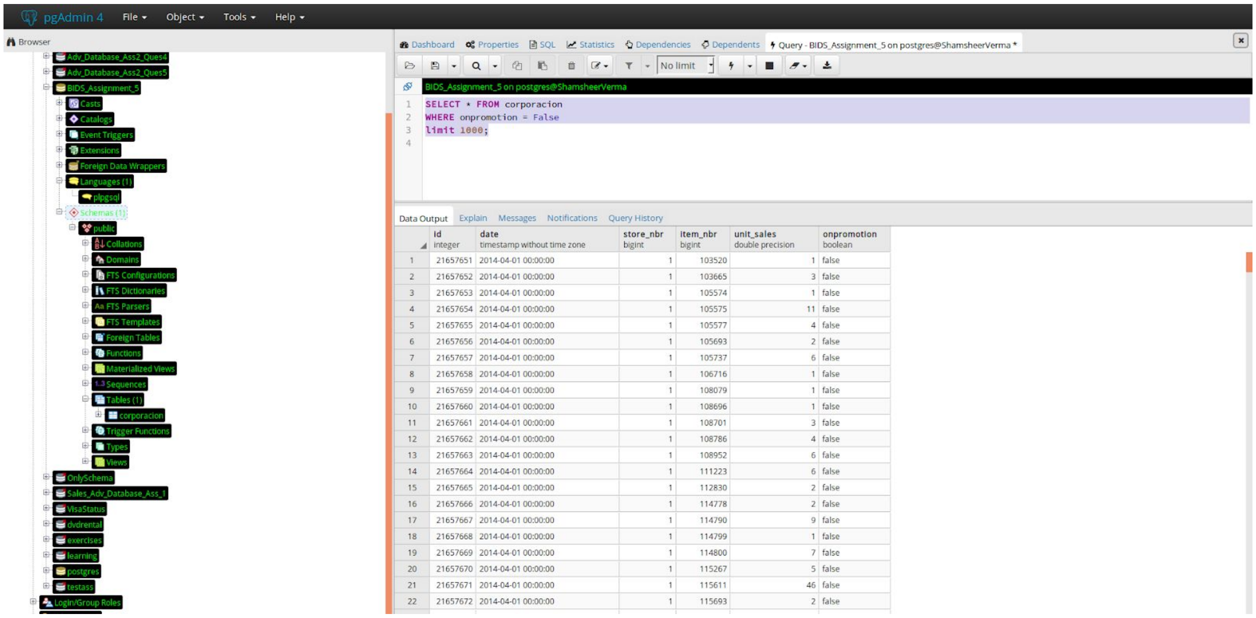
Fig 3. Graphical User Interface



```
        self.shop_dd = OptionMenu(self.master, self.variable_shop, *self.shop_dd_options, command = self.shopID)

        self.product_dd = OptionMenu(self.master, self.variable_product, *self.product_dd_options, command = self.productID)

        self.mydate = StringVar(self.master)
        self.date = tk.Entry(self.master, textvariable = self.mydate, width=10)

        self.v = IntVar()
        self.onpromotion_false = Radiobutton(self.master, text="Promotion False", variable = self.v, value = 0)
        self.onpromotion_true = Radiobutton(self.master, text="Promotion True", variable = self.v, value = 1)


        self.output_quantity = Label(self.master, text = "xxxx")
        self.shop_dd_label = Label(self.master, text = "Please Select the Shop ID")
        self.product_dd_label = Label(self.master, text = "Please Select the Product ID")
        self.date_dd_label = Label(self.master, text = "Enter the Date : DD/MM/YYYY")
        self.output_dd_label = Label(self.master, text = "Output Quantity is : ")

        self.name1 = Label(self.master, text = "Shamsheer Verma : 12851248")
        self.name2 = Label(self.master, text = "Ankit Kapoor : 12982233")
        self.name3 = Label(self.master, text = "Avinash Pakanati: 12723514")



        self.start = Button(self.master, text = "START THE MODEL", bg='white', fg='black')

        ##------------------------ Positioning the Drop Down, Input Box, RadioButtons, and Output ------------------------##

        self.onpromotion_false.config(height = 1, width = 28)
        self.onpromotion_true.config(height = 1, width = 25)
        self.shop_dd.config(height=1, width = 20)
        self.product_dd.config(height=1, width = 20)
        self.output_quantity.config(height=1, width = 20)
        self.start.config(height=1, width=20)
        self.shop_dd_label.config(height=1, width = 30)
        self.product_dd_label.config(height=1, width = 30)
        self.date_dd_label.config(height=1, width = 30)
        self.output_dd_label.config(height=1, width = 20)
        self.name1.config(height=1, width = 30)
        self.name2.config(height=1, width = 30)
        self.name3.config(height=1, width = 30)


        self.onpromotion_false.grid(row = 1, column = 0, sticky=W, pady = 2)
        self.onpromotion_true.grid(row = 1, column = 1,  sticky=E, pady = 2)
```

```
userinterface.py [/home/shamsheer/anaconda3/bin/python3.6m]
73449  125476607 2017-08-15         46     106716          9.0        False
73450  125479174 2017-08-15         47     106716         12.0        False
73451  125481877 2017-08-15         48     106716         13.0        False
73452  125484293 2017-08-15         49     106716          6.0        False
73453  125489316 2017-08-15         51     106716          5.0        False
73454  125491533 2017-08-15         52     106716         11.0        False
73455  125493910 2017-08-15         53     106716          7.0        False
73456  125495716 2017-08-15         54     106716          1.0        False

[73457 rows x 6 columns]
done
36
{'date': 36, 'store_nbr': '7', 'item_nbr': '106716', 'onpromotion': 0}
```

Fig 4. Python Development Environment



```
SELECT * FROM corporacion
WHERE onpromotion = False
limit 1000;
```

	id integer	date timestamp without time zone	store_nbr bigint	item_nbr bigint	unit_sales double precision	onpromotion boolean
1	21657651	2014-04-01 00:00:00	1	103520	1	false
2	21657652	2014-04-01 00:00:00	1	103665	3	false
3	21657653	2014-04-01 00:00:00	1	105574	1	false
4	21657654	2014-04-01 00:00:00	1	105575	11	false
5	21657655	2014-04-01 00:00:00	1	105577	4	false
6	21657656	2014-04-01 00:00:00	1	105693	2	false
7	21657657	2014-04-01 00:00:00	1	105737	6	false
8	21657658	2014-04-01 00:00:00	1	106716	1	false
9	21657659	2014-04-01 00:00:00	1	108079	1	false
10	21657660	2014-04-01 00:00:00	1	108696	1	false
11	21657661	2014-04-01 00:00:00	1	108701	3	false
12	21657662	2014-04-01 00:00:00	1	108786	4	false
13	21657663	2014-04-01 00:00:00	1	108952	6	false
14	21657664	2014-04-01 00:00:00	1	111223	6	false
15	21657665	2014-04-01 00:00:00	1	112830	2	false
16	21657666	2014-04-01 00:00:00	1	114778	2	false
17	21657667	2014-04-01 00:00:00	1	114790	9	false
18	21657668	2014-04-01 00:00:00	1	114799	1	false
19	21657669	2014-04-01 00:00:00	1	114800	7	false
20	21657670	2014-04-01 00:00:00	1	115267	5	false
21	21657671	2014-04-01 00:00:00	1	115611	46	false
22	21657672	2014-04-01 00:00:00	1	115693	2	false

Fig 5. Postgres SQL Database

Data quality and preparation

Data preparation is a time intensive process which takes about 60-90 percent of the whole procedure. Most of the time, data have problems and inconsistencies which need to be rectified before analysing and applying any predictive, prescriptive or descriptive model. In our case, the data is pretty consistent and clean except a few parameters.

Data Quality and Assessment Report

As this problem involves multiple datasets, there is a need to assess the quality of each dataset. All the datasets have consistent and non-null values except the train dataset which has about 16% of null values in onpromotion column. Moreover, the type values given for unit_sales are inconsistent. It has both float and int type of values because either the items are sold in integer values (a bag of chips) or in float values (1.5 kg of cheese). This needs to be standardized, otherwise, model will generate an error regarding multiple type of values involved.

Data Preparation Procedure

The datasets used for this problem has been taken from kaggle. These datasets were first loaded into Postgres SQL database from where it is fetched to python using database connection. Then, they were converted to dataframe, so that further operations can be applied. First of all, all the rows with null values in "onpromotion" column were dropped because only 16% of the records have null values. Moreover, whenever the data frame is loaded into the jupyter notebook, it cannot read columns having multiple data types.

Unit_sales have values ranging from -15,372 to 89,440 covering a huge spectrum. These values have been normalized using minmax scalar standardization.

In addition to normalization, factorization of onpromotion attribute has been done, which creates the column consisting of only 1 and 0. It becomes easy for a model to get trained using numerical values instead of categorical values. Some models do not support categorical values, so to be supported for various models it is recommended to convert these categorical values to numerical values.

The machine learning models do not take date time stamp values. Hence date attribute values has been transformed to number of days from the present date so that it can be used for the prediction process. Predicting unit_sales is not a classification problem where data is to be classified such as True or False. The whole problem is based on the prediction of continuous variables, i.e, the output data is a variable where it can range in any order.

Predictive Analysis

There are four regression methods which we have used for predicting the sales. Each of the regression techniques have been discussed below.

Linear Regression

Linear regression is a linear approach used to model the relationship between a scalar response and one or more independent variables by calculating the bias and coefficient terms. The regression plot for linear regression has been shown below. It can easily be deduced from the plot that there seems to be no linear relationship between predicted sales and actual sales or in other words predicted sales are not matching with the actual sales.

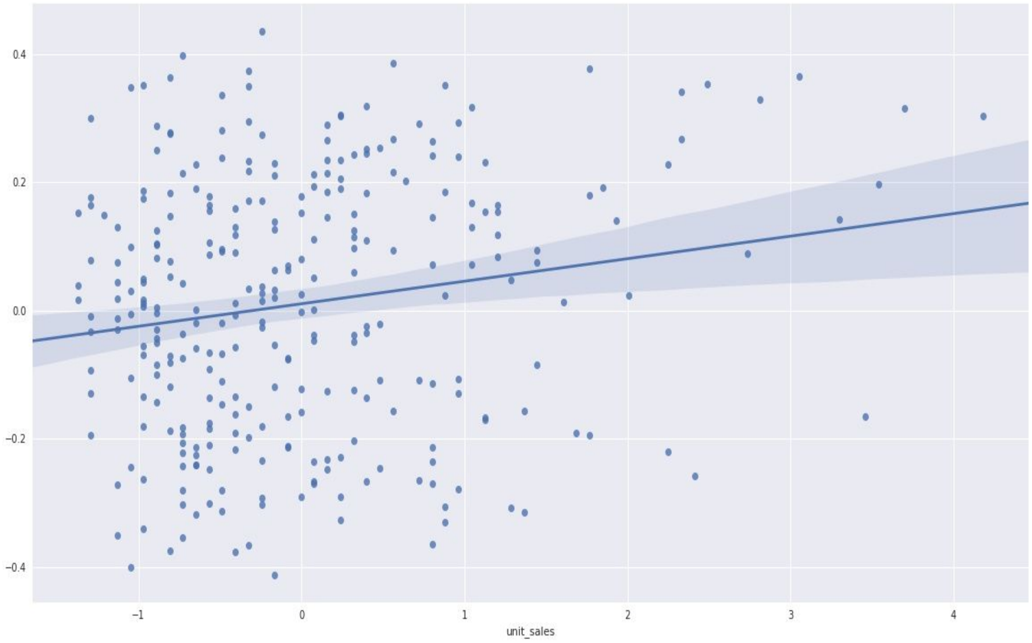
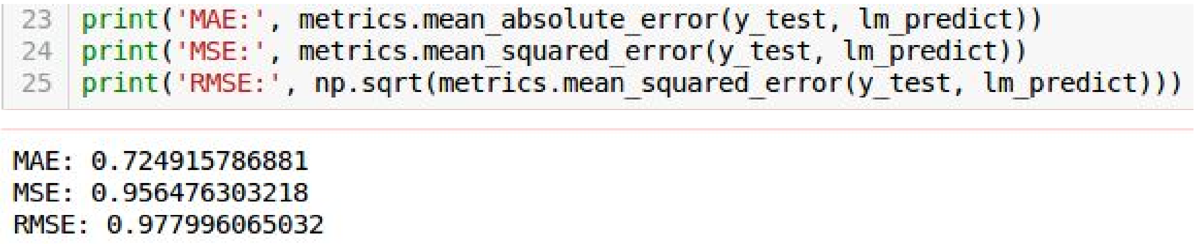


Fig 6. Regression Plot for Linear Regression

Also to estimate the accuracy of the model, different error metrics have been evaluated using sklearn metrics. Although magnitude does not reflect how the model is, but from the plot it can be estimated that this linear approach is not suitable for sales prediction.

MAE	MSE	RMSE
0.7249	0.9564	0.9779

Fig 7. Error Metrics for Linear Regression



```
print('MAE:', metrics.mean_absolute_error(y_test, lm_predict))
print('MSE:', metrics.mean_squared_error(y_test, lm_predict))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, lm_predict)))
```

```
MAE: 0.724915786881
MSE: 0.956476303218
RMSE: 0.977996065032
```

Fig 8. Python code for error metric calculation

Polynomial Regression

Polynomial regression is an another type of regression technique which is similar to linear regression except it finds the n-degree polynomial to find the relationship between dependent and independent variables. The degree of this polynomial can be adjusted while initializing the model. The regression plot for this technique has been shown below. This technique is performing pretty well as compared to linear regression but this result can further be improved using other techniques available.

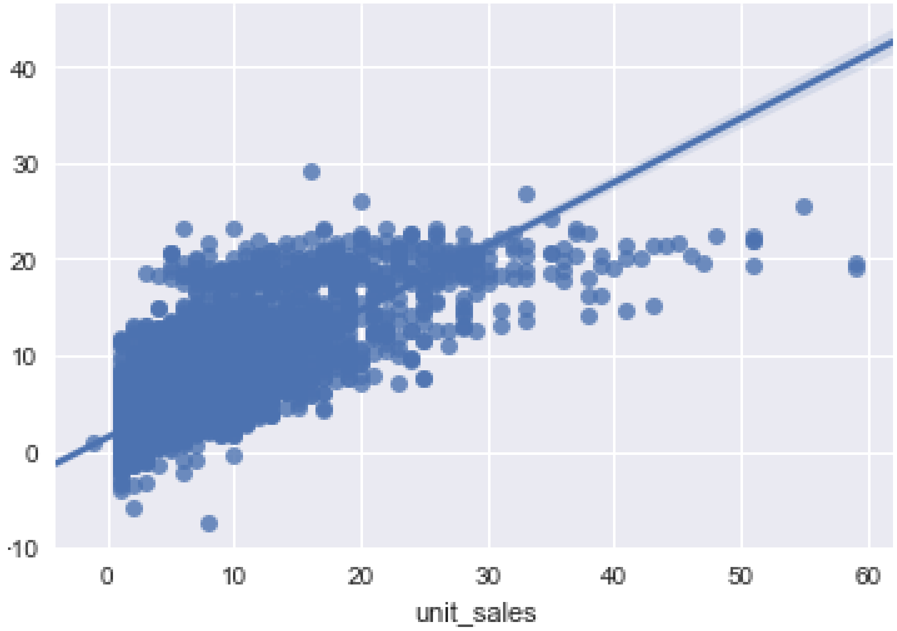
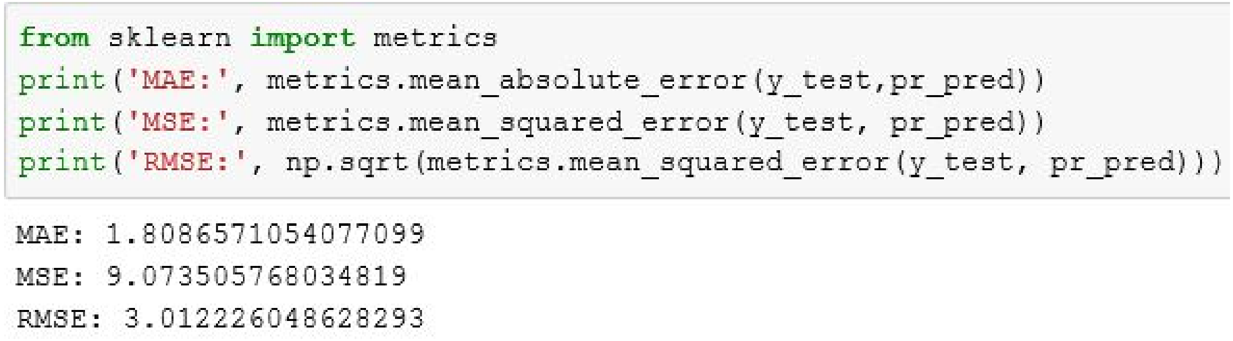


Fig 9. Regression plot for Polynomial Regression.

The error metrics have also been calculated for this regression technique and results have been shown below. The magnitude of error metrics comes to be really high which makes this technique inappropriate for predicting sales

MAE	MSE	RMSE
1.808	9.073	3.0122

Fig 10. Error metrics for Polynomial Regression



```
from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test,pr_pred))
print('MSE:', metrics.mean_squared_error(y_test, pr_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, pr_pred)))
```

```
MAE: 1.8086571054077099
MSE: 9.073505768034819
RMSE: 3.012226048628293
```

Fig 11. Python code for error metric calculation

Kernel Ridge Regression

Another popular method Kernel Ridge Regression was applied to see how the results stand to other used techniques. The regression plot for this technique has been shown below. Again, this technique is not suitable for forecasting sales as trend between predicted and actual sales is not conforming a linear relationship.

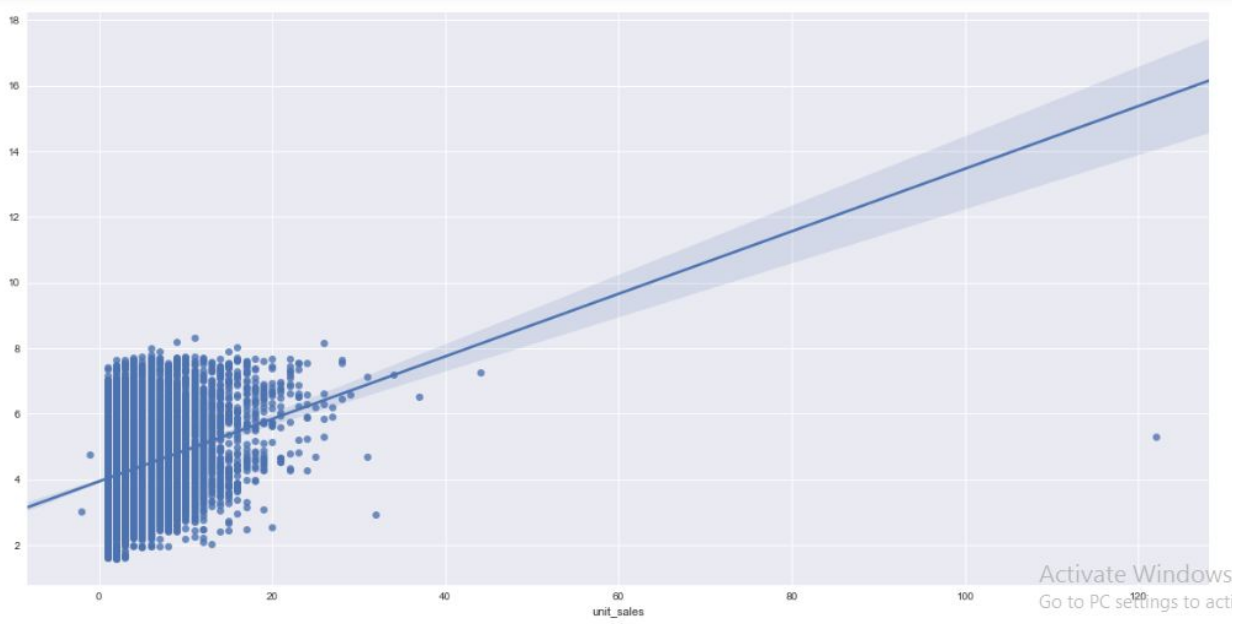
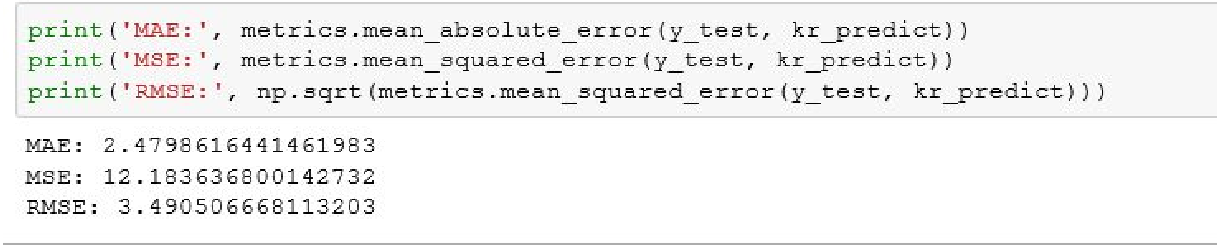


Fig 12. Regression Plot for Kernel Ridge regression

The error metrics for kernel ridge regression has been calculated using the following python code. The magnitude of these error metrics is found to be very high which makes it unsuitable for this project.

MAE	MSE	RMSE
2.479	12.1836	3.4905

Fig 13. Error metrics for Kernel Ridge regression



```
print('MAE:', metrics.mean_absolute_error(y_test, kr_predict))
print('MSE:', metrics.mean_squared_error(y_test, kr_predict))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, kr_predict)))
```

```
MAE: 2.4798616441461983
MSE: 12.183636800142732
RMSE: 3.490506668113203
```

Fig 14. Python code for error metric calculation

Random Forest Regression

It is the fourth regression technique which we used for forecasting sales. This is an ensemble learning method which uses number of decision trees to predict the sales value. Then the average of these values is taken as final output as shown in image below.

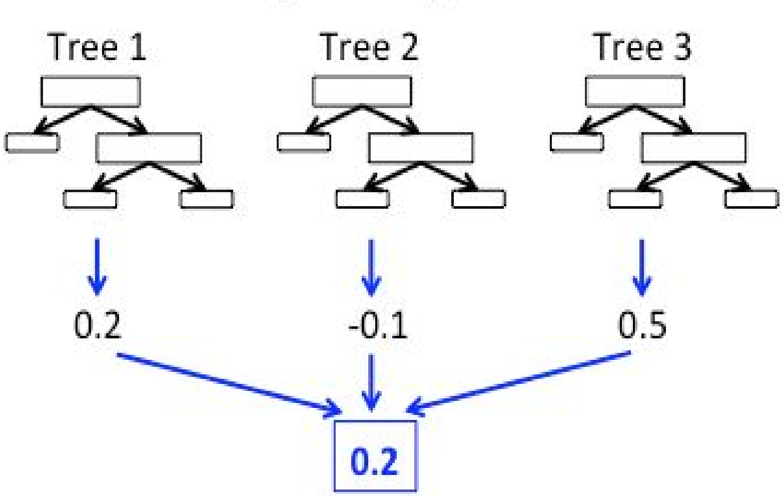


Fig 15. Random Forest Regression

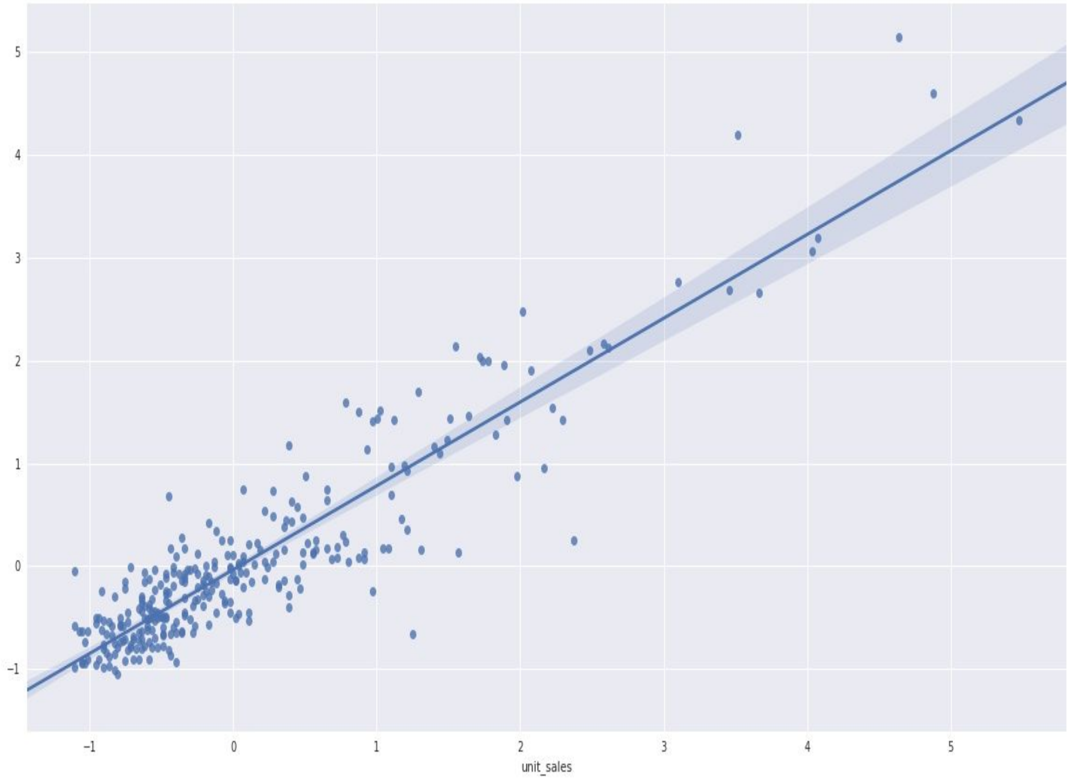
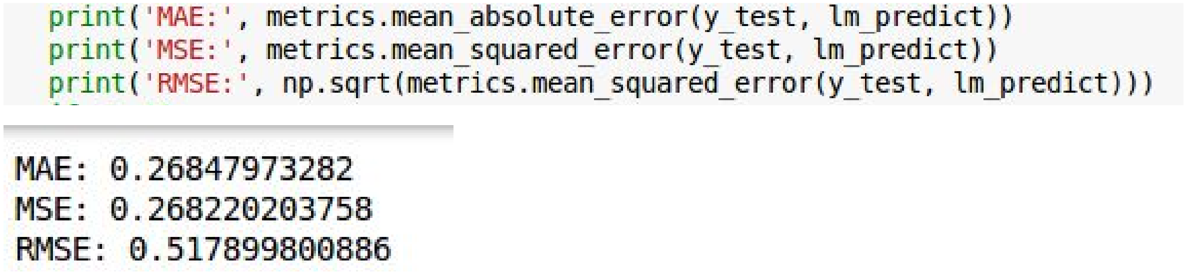


Fig 16. Regression Plot for Random Forest Regression

The regression plot for the random forest regression has been shown above. It can be seen from the regression plot that the predicted sales and actual sales form an almost linear relationship which makes it suitable for sales forecasting. Also the error metric calculated using python code gave much lesser value in terms of magnitude.

MAE	MSE	RMSE
0.2684	0.26822	0.5178

Fig 17. Error metrics for Random Forest Regression



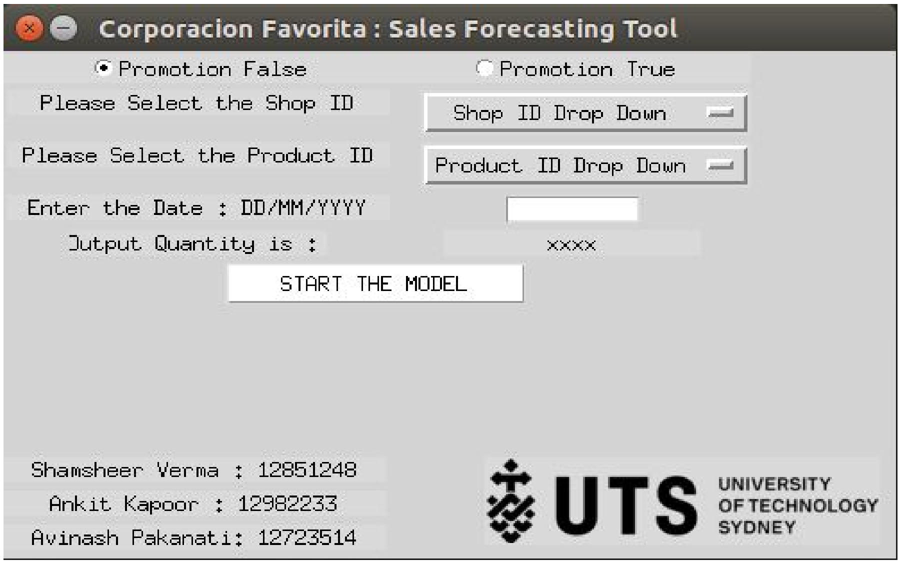
```
print('MAE:', metrics.mean_absolute_error(y_test, lm_predict))
print('MSE:', metrics.mean_squared_error(y_test, lm_predict))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, lm_predict)))
```

```
MAE: 0.26847973282
MSE: 0.268220203758
RMSE: 0.517899800886
```

Fig 18. Python code for error metric calculation

Graphical User Interface (GUI)

Graphical User Interface is one of the most important part of a product. This is where managers and customers interact with the data without directly engaging with the back end of the code. A good GUI allows the customer to understand, interact, and manipulate the inputs according to the business requirements and in exchange provide the meaningful output to the user at the end. The below figure shows the GUI for this project.



Corporacion Favorita : Sales Forecasting Tool

● Promotion False ○ Promotion True

Please Select the Shop ID: Shop ID Drop Down

Please Select the Product ID: Product ID Drop Down

Enter the Date : DD/MM/YYYY

Output Quantity is : xxxx

START THE MODEL

Shamsheer Verma : 12851248

Ankit Kapoor : 12982233

Avinash Pakanati: 12723514

Fig 19. GUI for the Sales Forecasting Tool

Input and Output Prompts

The Input and Output prompts in this User Interface are :-

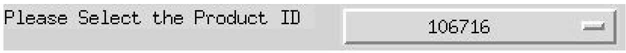
1. Promotion (True/False) - The promotion radio button indicates whether the product is in Promotion/Discounted or not. If the user clicks on the False button then the input is harnessed as No Promotion as shown in below .

● Promotion False ○ Promotion True

2. Shop ID - It is a drop down menu for the user to select a particular Shop ID. Once the user selects a Shop ID, the program automatically inputs the values of Shop ID in the system. Currently, there are 10 Shop ID’s present, however, this can be increased as per requirements as shown below .

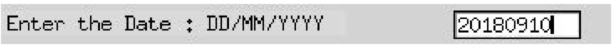
Please Select the Shop ID: 6

3. Product ID - It is similar to Shop ID’s drop down menu. However, instead of shops, this prompt allows the user to choose and select a particular product. There are currently 10 Products to choose from as an example, nevertheless, this can be increased as per the requirements as shown below .



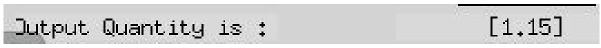
Please Select the Product ID 106716

4. Date - This involves the user to enter a particular date in the YYYYMMDD format. The date entered here represents the time at when the quantity of item number to be predicted as shown in below .



Enter the Date : DD/MM/YYYY 20180910

5. Output Quantity - This is a model generated output where the back end code inputs all the above data and performs calculations to provide an appropriate result. The output is in the float format with upto two decimal points as shown below .



Output Quantity is : [1.15]

6. Start the model - This prompt is in the click button form where once a user clicks on the START THE MODEL, it triggers the model to start it work and perform calculations at the back end as shown below .

Process for Using User Interface (UI)

Below is the procedure to use the developed user interface.

Step 1

The first step is to select whether the product that is to be predicted is on promotion or not. After clicking on the Promotion as shown in Fig 16 as the Step 1, it will trigger the python code to receive the string values as True or False.

Step 2

In this step, the user will select the appropriate Shop ID from the drop down menu which will reflect the product to be inside that shop as shown in Fig 16. This will help the model to predict the output according to Shop that the user is interested in.

Step 3

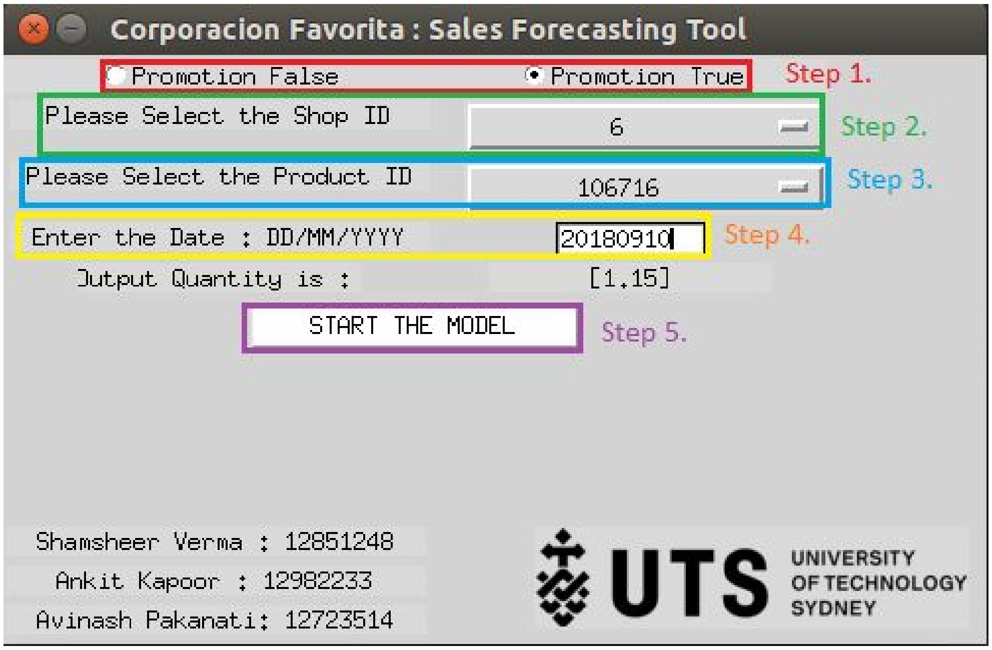
The third step is to select the Product ID from the drop down menu as shown in Fig 16. This will allow the user to focus on the particular product ID so that the output will be associated to a particular item.

Step 4

Second last step is to enter the date in the YYYYMMDD format as shown in Fig 16. This will help the user to predict the output for a particular date.

Step 5

The last step is to click on the "START THE MODEL" prompt as shown in Fig 16. This will trigger the model to start working and provide the appropriate result at the end.



Corporacion Favorita : Sales Forecasting Tool

○ Promotion False ● Promotion True

Please Select the Shop ID: 6

Please Select the Product ID: 106716

Enter the Date : DD/MM/YYYY 20180910

Output Quantity is : [1,15]

Shamsheer Verma : 12851248
Ankit Kapoor : 12982233
Avinash Pakanati: 12723514

Fig 20. Steps to use the Forecasting Tool

Business Recommendation

Currently, managers are using experience and word of mouth to forecast the number of items to be purchased to fill the inventory. This is subjective in nature results in accuracy in forecasting. This accuracy can increased by the use of different data handling methods and deep learning or machine learning algorithms that can further enhance the accuracy which in turn could optimize the product unit sales.

As shown in the Predictive Analysis section, Model Preparation and selection has been an iterative process where different models are implemented according to the business requirements and accuracy. The linear regression was the first model to initiate with and afterwards we moved towards complex techniques such as Poly Regression, Kernel Ridge, and Random Forests Regression successively. Out of all the machine learning models, Random Forest Regression technique have performed very well and has given the minimum error of 0.2684 compared to other models.

Conclusion

In a nutshell, an accurate forecasting too has been developed which can be used for forecasting sales instead of subjected that are being used by Corporacion Favorita. The Random Forest Regression is running at the background to predict the sales. Various models have been tested before deploying random forest for this tool. Random Forest Regression technique is used to achieve good accuracy which can be obtained with a single click and few selections as Input. The user friendly and responsive interface has been created for Store managers who might not have any prior knowledge of Python programming.

References

Corporación Favorita Grocery Sales Forecasting 2017, viewed 1 October 2018, <https://www.kaggle.com/c/favorita-grocery-sales-forecasting>.