Stores Sales Prediction

-iNeuron Internship

BY DERRICK T

Problem Statement:

- Nowadays, shopping malls and Big Marts keep track of individual item sales data in order to forecast future client demand and adjust inventory management. In a data warehouse, these data stores hold a significant amount of consumer information and particular item details. By mining the data store from the data warehouse, more anomalies and common patterns can be discovered.

Approach: The classical machine learning tasks like Data Exploration, Feature Engineering, Model Building and Model Testing. Try out different machine learning algorithms that's best fit for the above case.

Results: To build a solution that should able to predict the sales of the different stores of Big Mart according to the provided dataset.

Dataset:

- We have train (8523) and test (5681) data set, train data set has both input and output variable(s). We need to predict the sales for test data set. >>Item Identifier: Unique product ID >>Item_Outlet_Sales: Sales of the product in the particular store. This is the outcome variable to be predicted.<<

Item_Weight: Weight of product

Item_Fat_Content: Whether the product is low fat or not

Item_Visibility: The % of total display area of all products in a store allocated to the particular product

Item_Type: The category to which the product belongs

Item_MRP: Maximum Retail Price (list price) of the product

Outlet_Identifier: Unique store ID

Outlet_Establishment_Year: The year in which store was established

Outlet_Size: The size of the store in terms of ground area covered

Outlet_Location_Type: The type of city in which the store is located

Outlet_Type: Whether the outlet is just a grocery store or some sort of supermarket

Details

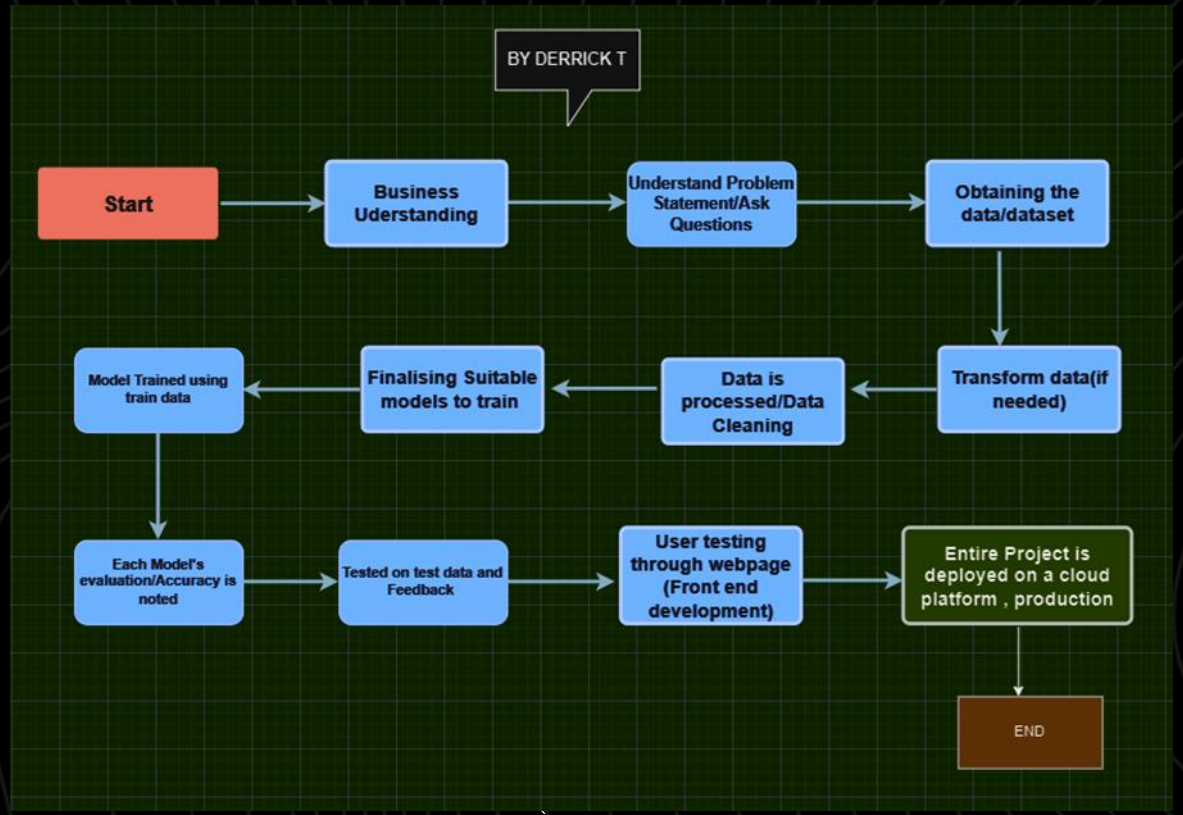
- The proposed solution is building a machine model that is trained using past data and evaluated, and the final evaluated model is fed in the unseen data to make predictions, in this case the sales amount is predicted.
- There were no technical requirements for developing this solution, the default requirements were a Personal Computer with good computing power.
- The data required was based on the problem statement and the variables or the dependencies provided. The client provides the dataset with relevant columns as mentioned in the problem statement. After the solution is developed, user data of the user's choice will be required.

Tools Used

Contd.

- Python was used as the programming language – to load, pre-process, scale, build ML model and for saving it.
- For visualization, data wrangling etc, frameworks such as NumPy, pandas, matplotlib, scikit learn were used
- PyCharm was used as the IDE for developing a flask application.
- Jupyter notebook as the IDE for developing the solution.
- Visual Studio Code to develop a webpage using HTML and CSS.

Process Flow



Model:

```
In [133]:
1 y_train = y_train.fillna(y_train.mean())

In [153]:
1 lr.fit(x_train_std,y_train)

Out[153]: LinearRegression()

In [154]:
1 y_pred_lr=lr.predict(x_test_std)

In [71]:
1 y_test

Out[71]: 8179     904.8222
         8355    2795.6942
         3411    1947.4650
         7089     872.8638
         6954    2450.1440
                    ...
         1317    1721.0930
         4996     914.8092
         531      370.1848
         3891    1358.2320
         6629    2418.1856
         Name: Item_Outlet_Sales, Length: 1705, dtype: float64
```

```
In [158]:
1 from sklearn.ensemble import RandomForestRegressor
2 rf=RandomForestRegressor()

In [159]:
1 rf.fit(x_train,y_train)

Out[159]: RandomForestRegressor()

In [160]:
1 y_pred_rf=rf.predict(x_test)

In [161]:
1 print(r2_score(y_test,y_pred_rf))
2 print(mean_absolute_error(y_test,y_pred_rf))
3 print(np.sqrt(mean_squared_error(y_test,y_pred_rf)))

0.536975573184493
787.945689415224
1123.3478725508144
```

Model:

The model was trained using the train data and accuracy was observed on:

For linear regression model:

```
print(mean_absolute_error(y_test,y_pred_lr))
print(np.sqrt(mean_squared_error(y_test,y_pred_lr)))
```

```
888.3515689944207
1174.8379645681518
```

For random forest model:

```
print(r2_score(y_test,y_pred_rf))
print(mean_absolute_error(y_test,y_pred_rf))
print(np.sqrt(mean_squared_error(y_test,y_pred_rf)))
```

```
0.536975573184493
787.945689415224
1123.3478725508144
```

Performance

The solution developed should be very accurate as the sales might differ each time, all the factors when entered correctly by the user will be a step to provide accurate predictions.

Performance can be improved by training the model on new data and evaluating it often.

Reusability

The code written and the components used should have the ability to be reused with no problems.

Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

Conclusion

- Random Forest model provided more accuracy than linear regression model.

- From this , we can conclude that this model can be used as a sales improving technique by forecasting sales for different product aspects and plan a business model accordingly.