

CRISP DM Method - Project Report

Walmart Recruiting - Store Sales Forecasting

STEP 1 : Introduction and Initial Process

Introduction:

Walmart is one of the fortune 500 companies having been on No 1 spot for the last 8 years. So, for such a huge company to keep its position it becomes important to analyse the sales so that they remain in their position. So predicting the sales for future is important to keep up the business. We predict the sales of the future using the past data of 3 years from 45 stores , trying out various models to improve the accuracy up on the whole.

Problem Description:

In this project we have been provided with historical sales data for 99 products across 45 outlets. Each store contains a number of departments, and we have to predict the Weekly Sales. In addition, Walmart runs several promotional markdown events throughout the year. These markdowns precede prominent holidays, the four largest of which are the Super Bowl, Labor Day, Thanksgiving, and Christmas. The weeks including these holidays are weighted five times higher in the evaluation than non-holiday weeks. Part of the challenge presented by this competition is modeling the effects of markdowns on these holiday weeks in the absence of complete/ideal historical data.

Problem Statement:

To determine the sales in Walmart in each department during any day and how it is going to affect their sales and when by using prediction models with the given data.

Objective:

To predict the sales in a department provided few details like store, department, date , their type, fuel price, job status and markdown values.

Plan:

To create a workable model to determine the sales prediction with high accuracy possible

Resources used:

An interactive scripting environment(Google Colab/ Kaggle Notebook)

Anaconda Prompt

Postman

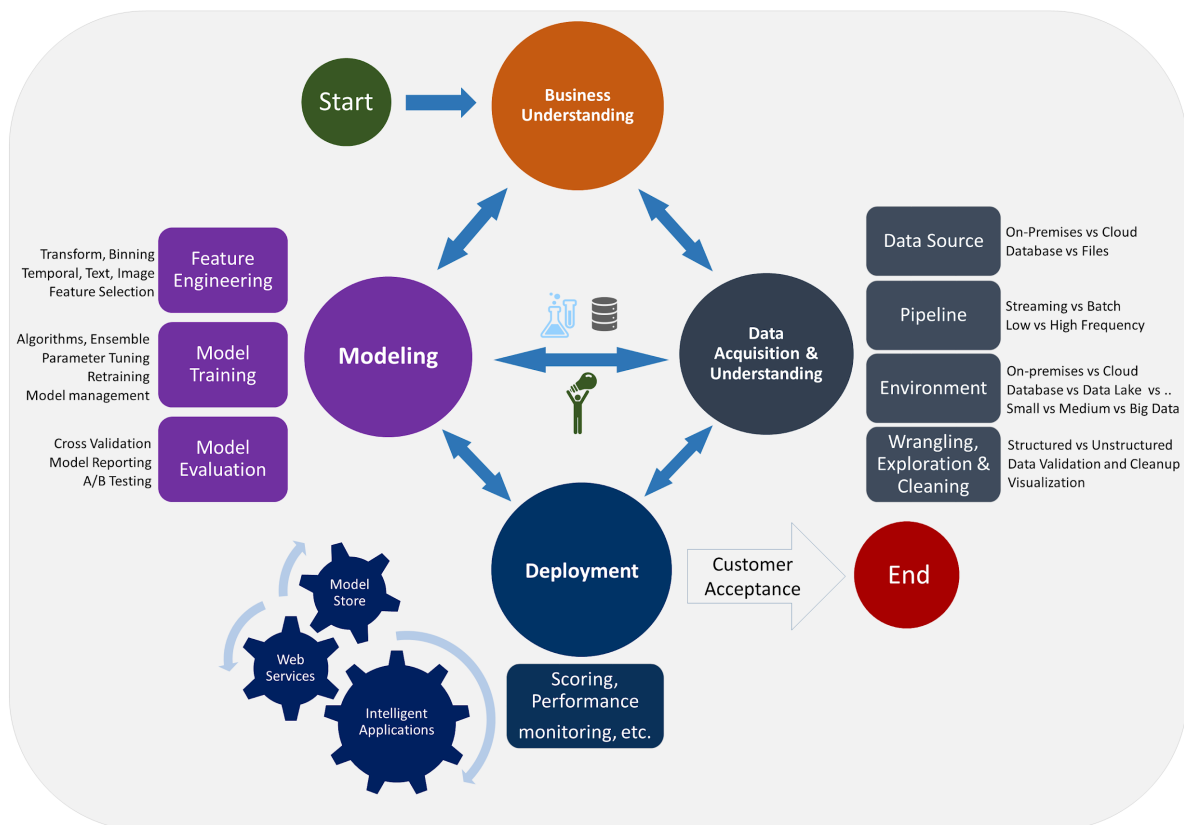
Various packages in python

Requirements:

All the requirements used is present in the requirements.txt

STEP 2: Project Cycle:

Data Science Lifecycle



In a machine learning life cycle the above steps are all used. Now, we determine how we do this with respect to our model.

Understanding the business of Walmart is important to predict the sales. Once we understand that obtaining the data is important. To analyse and create a model we need data. Processing the data is important to ensure that reliability is high. We have to preprocess, remove and do all data transformations suitable to ensure stability in model creation. Post this, we determine the most suitable model by trying out various models depending upon whether the problem is a regression or classification one. Walmart Sales prediction is a regression one. After determining the best model, we deploy it in the front end for customer use. That could be done using Flask. In our case, I generated a pickle file which was deployed using a flask model. Further, we can deploy it in a website and provide it to the customer. This is the overall life cycle process.

STEP 3: Data Understanding

We have been provided with 5 datasets in '.csv' format. The description is as follows:

About the DATA:

Stores.csv

This file contains anonymized information about the 45 stores, indicating the type and size of store.

train.csv

This is the historical training data, which covers from 2010-02-05 to 2012-11-01.

Within this file you will find the following fields:

Store — the store number

Dept — the department number

Date — the week

Weekly_Sales — sales for the given department in the given store

IsHoliday — whether the week is a special holiday week

test.csv

This file is identical to train.csv, except we have withheld the weekly sales. You must predict the sales for each triplet of store, department, and date in this file.

Features.csv

This file contains additional data related to the store, department, and regional activity for the given dates. It contains the following fields:

Store — the store number

Date — the week

Temperature — average temperature in the region

Fuel_Price — cost of fuel in the region

Markdown 1-5 — anonymized data related to promotional markdowns that Walmart is running. Markdown data is only available after Nov 2011, and is not available for all stores all the time. Any missing value is marked with an NA.

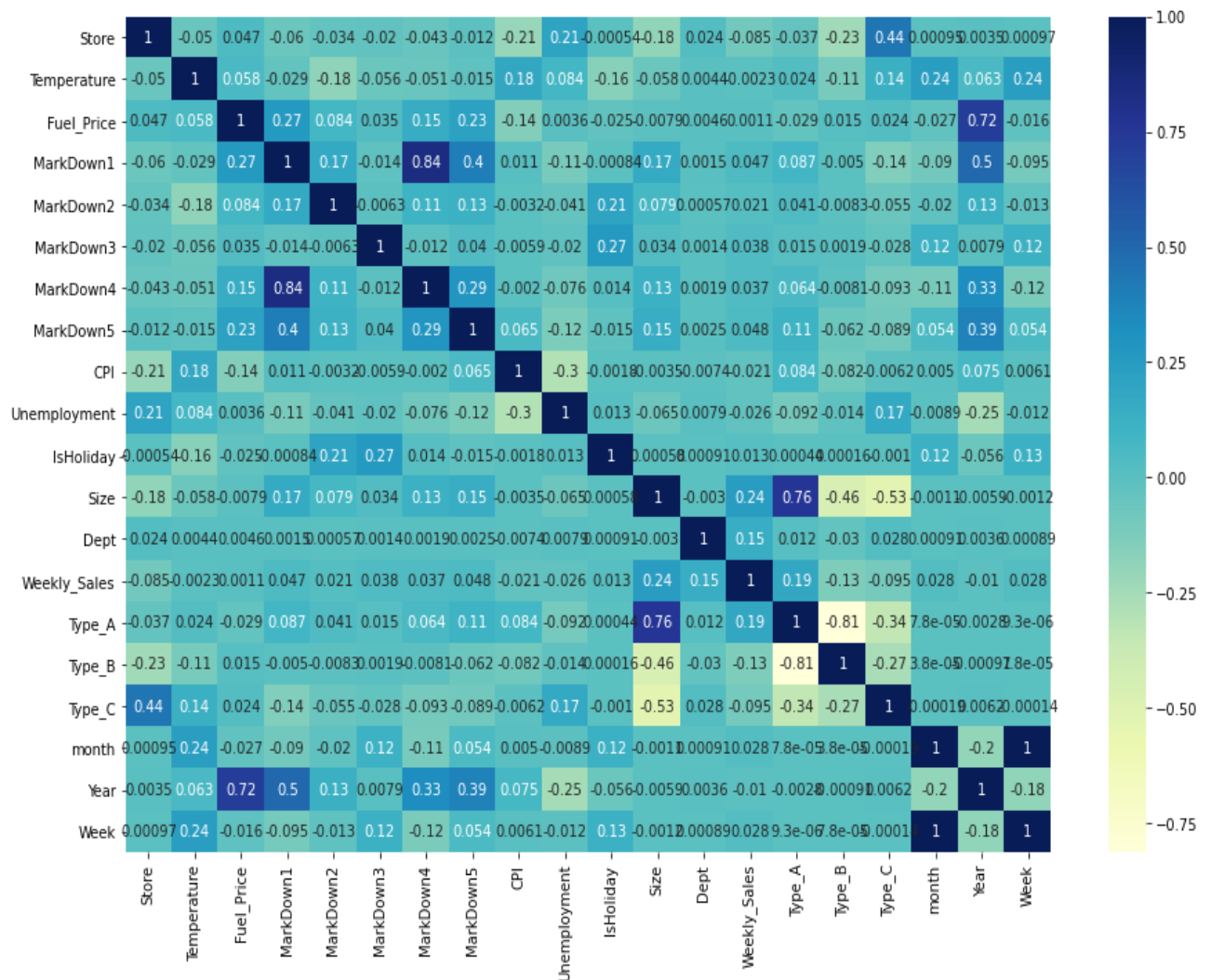
CPI — the consumer price index

Unemployment — the unemployment rate

IsHoliday — whether the week is a special holiday week.

STEP 4: Exploratory Data Analysis & Data Preparation

The correlation heatmap for the combined data (Train + Stores + Features) is as follows:



Missing Values:

A screenshot of a Jupyter Notebook cell showing the output of the command `fsmerge.isnull().sum()`. The output is a table with two columns: the variable name and the count of missing values. The variables and their counts are: Store (0), Date (0), Temperature (0), Fuel_Price (0), Markdown1 (270892), Markdown2 (310793), Markdown3 (284667), Markdown4 (286859), Markdown5 (270138), CPI (585), Unemployment (585), IsHoliday (0), Type (0), Size (0), Dept (1755), and Weekly_Sales (1755). The dtype is int64.

Variable	Count
Store	0
Date	0
Temperature	0
Fuel_Price	0
Markdown1	270892
Markdown2	310793
Markdown3	284667
Markdown4	286859
Markdown5	270138
CPI	585
Unemployment	585
IsHoliday	0
Type	0
Size	0
Dept	1755
Weekly_Sales	1755
dtype	int64

We need to process all the missing values either by removing them or by providing the mean values or by filling them with zeroes. Here we fill 0 to all the markdown values which have not been filled.

Data Processing:

Once we merge the train , store and features data together into the single data, we start cleaning it based on its importance. We drop tables like Date, CPI, Fuel Price, Unemployment and Markdown3.

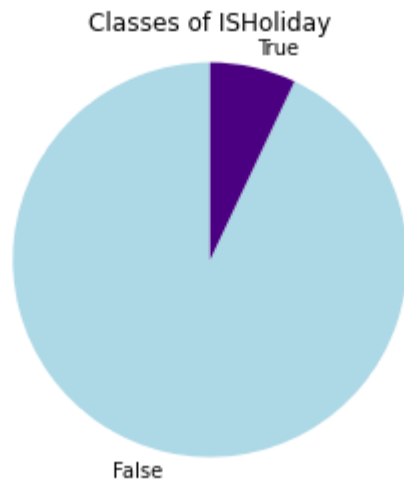
We need to convert the date from string to datetime format.

Create dummies of Type column to make it TypeA, TypeB, TypeC.

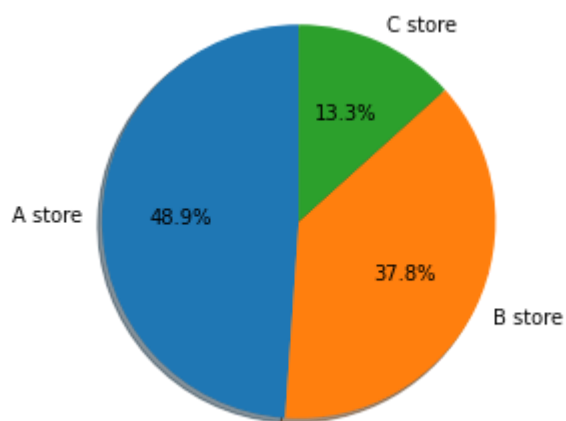
All these steps help in getting a stable model.

We convert the isHoliday variable of type 'bool' to int to facilitate easy access.

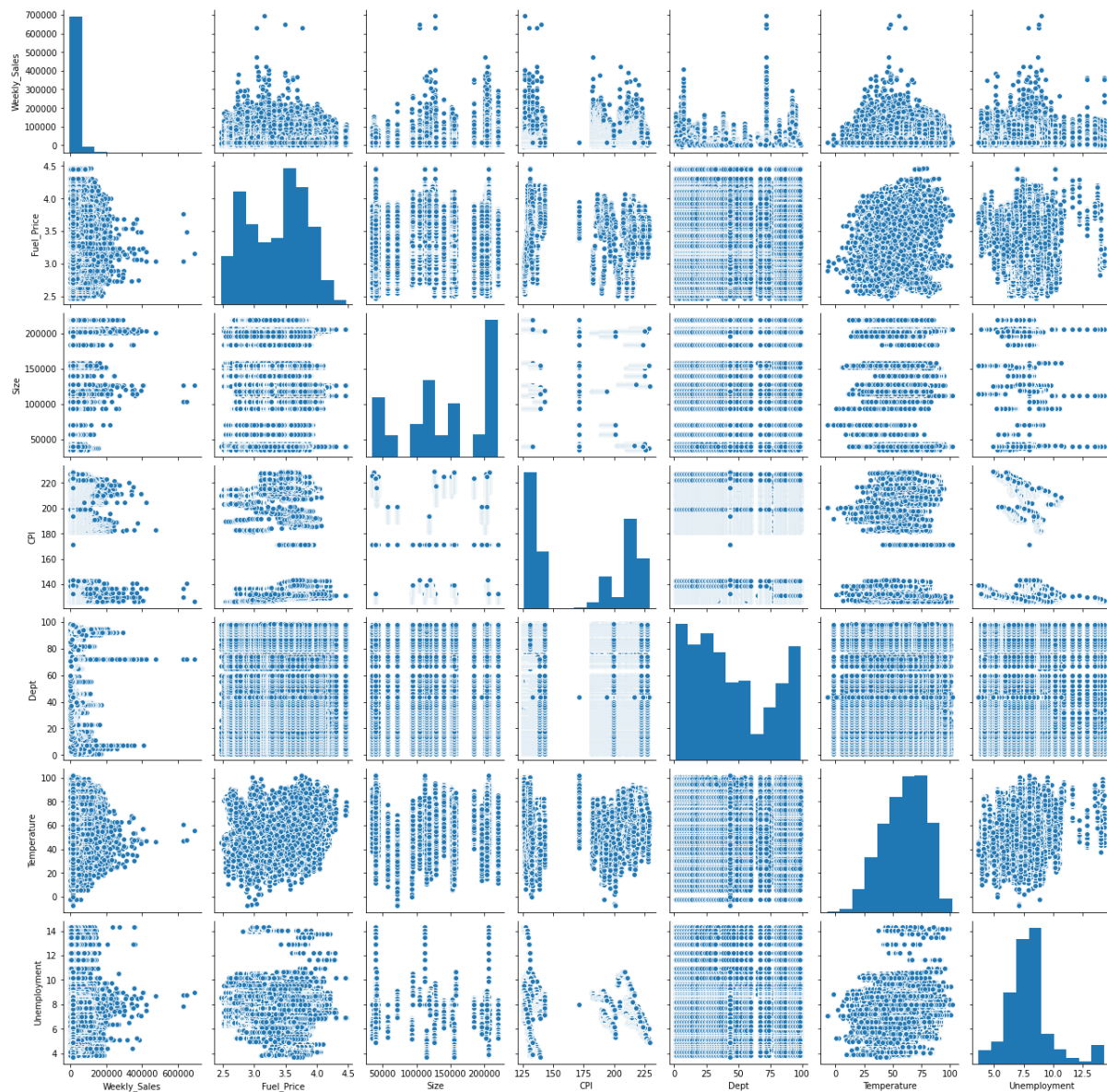
The total percentage if true or false values are as below.



The pie chart displays the store types in categories of each type that is available.



Pair plots of various features like 'Weekly_Sales', 'Fuel_Price', 'Size', 'CPI', 'Dept', 'Temperature', 'Unemployment' are as follows:

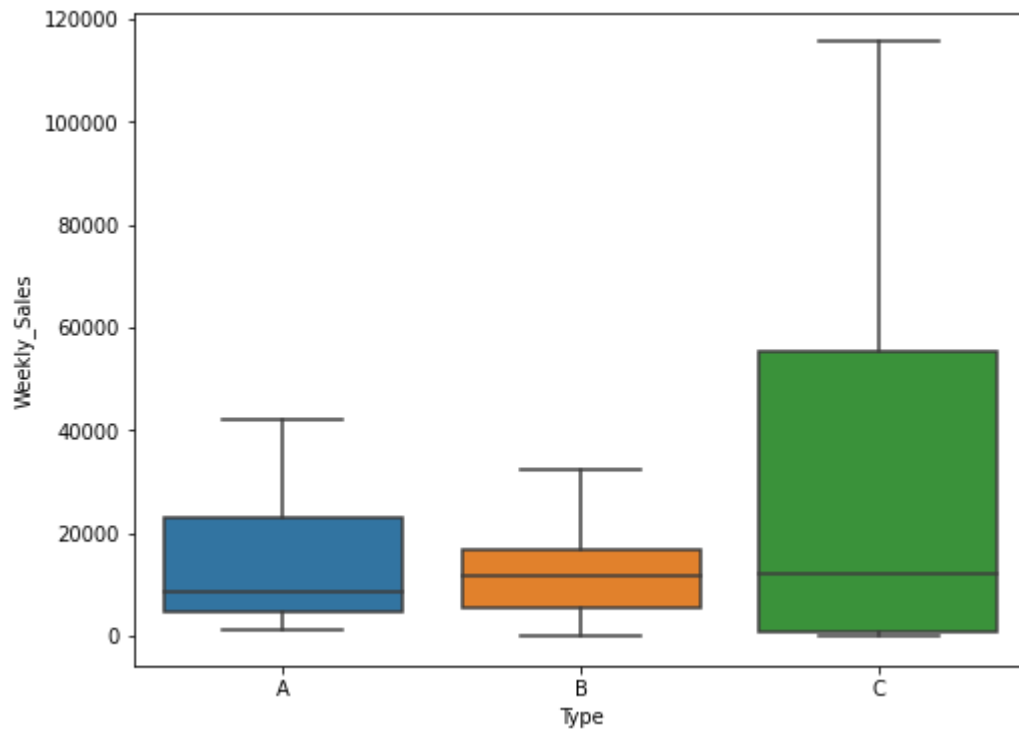


Depending on these we can find their relation between them.

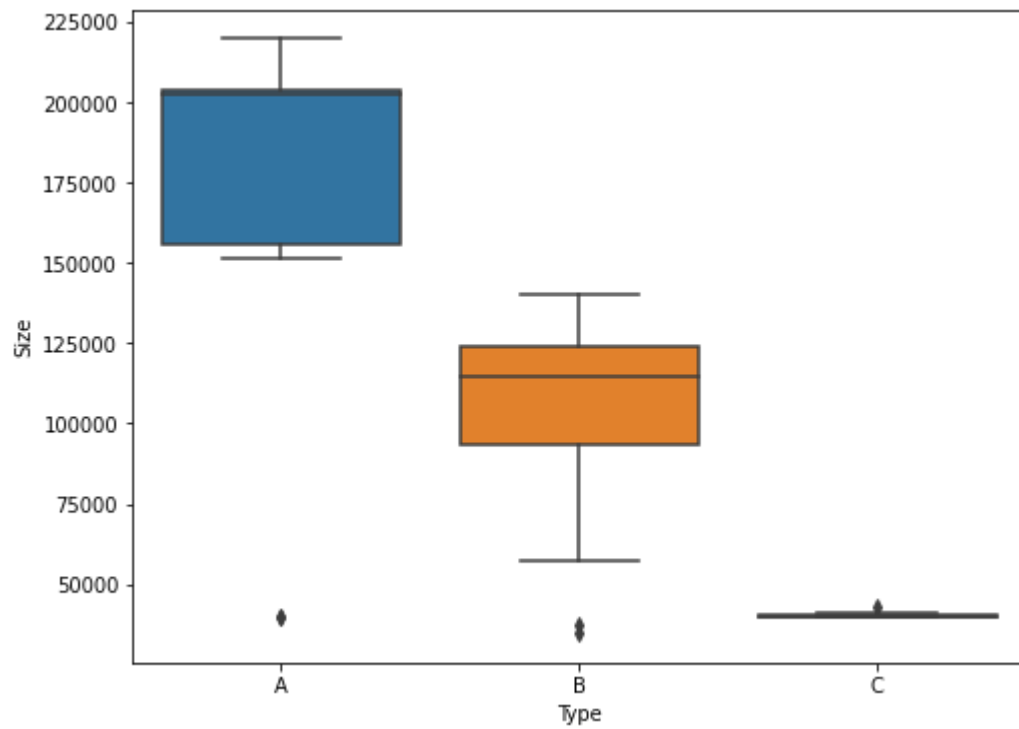
Outliers:

An **outlier** is an observation that lies an abnormal distance from other values in a random sample from a population. They are datas that should be handled to ensure a better model. In our case handling them where not necessary.

Few representations include,



Type with respect to weeklysales.



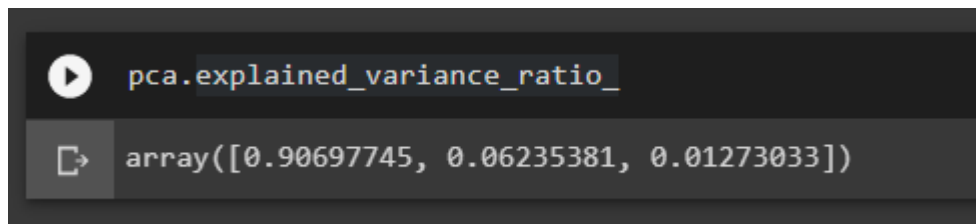
Type with respect to size

Data was now prepared well.

Principal Component Analysis:

Principal Component Analysis, or **PCA**, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. In case of huge rows we can use this to reduce the dimension and increase the accuracy. However, while reducing and using it the mean variance ratio should be around 70 to 80% to retain the information.

I tried to reduce them using PCA and had around



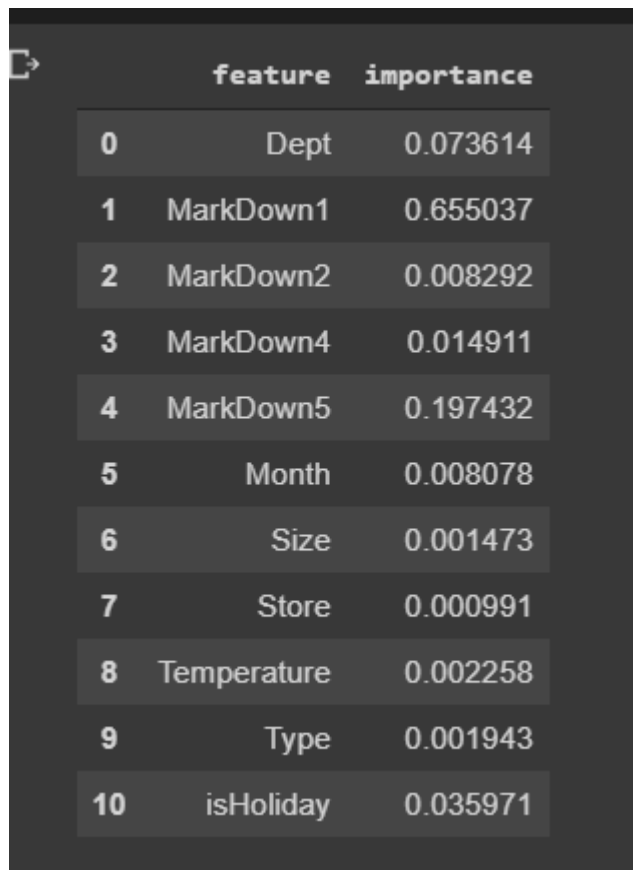
```
pca.explained_variance_ratio_
array([0.90697745, 0.06235381, 0.01273033])
```

Hence, I omitted PCA for this model.

Features Importance:

We determine which features are most important from Decision Tree Regressor.

For us it is,



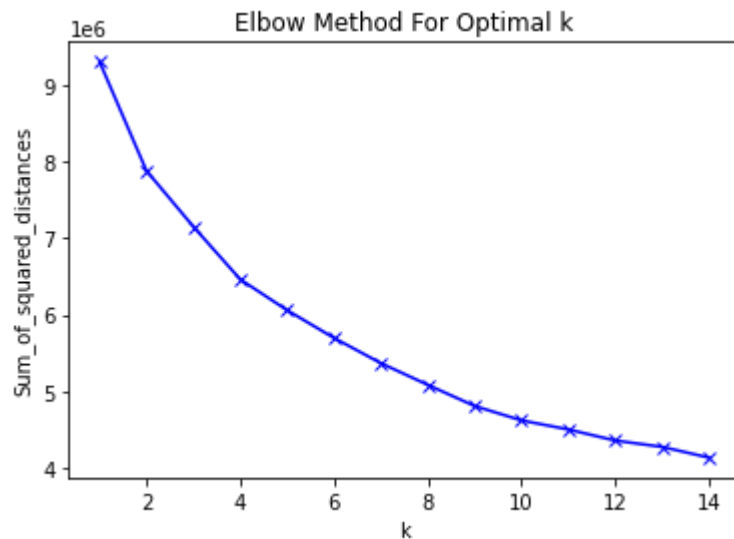
	feature	importance
0	Dept	0.073614
1	MarkDown1	0.655037
2	MarkDown2	0.008292
3	MarkDown4	0.014911
4	MarkDown5	0.197432
5	Month	0.008078
6	Size	0.001473
7	Store	0.000991
8	Temperature	0.002258
9	Type	0.001943
10	isHoliday	0.035971

Clustering:

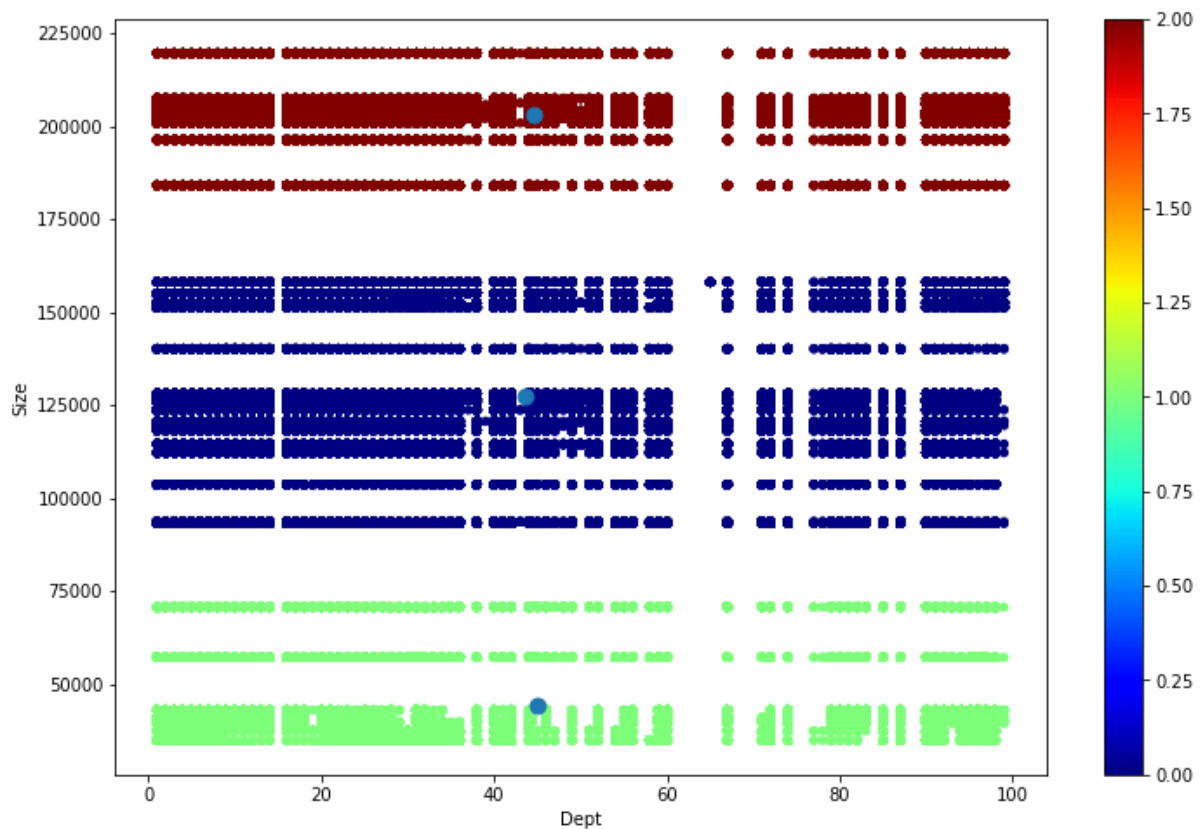
Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

It is important to depict which are closely or non closely related.

We use k-elbow method to determine the optimal number of clusters



The relation between Department and Size



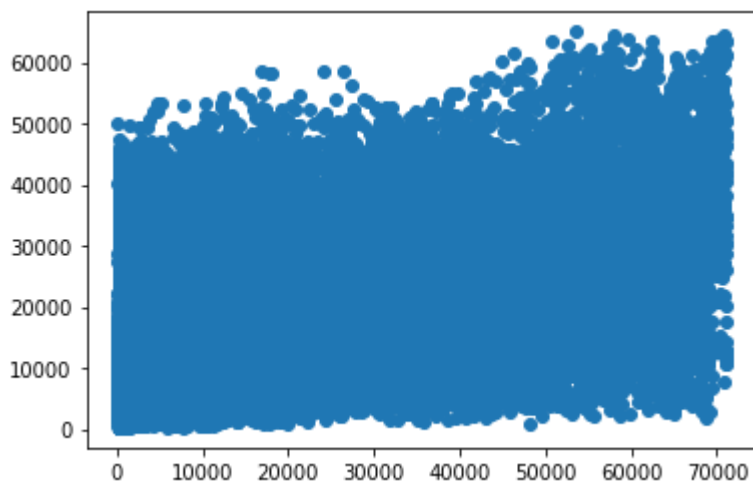
STEP 5: Model Selection & Evaluation

Each model has its own positives and negative sides. We use KNeighbours, Extra Trees Regressor, Random Forest, XGBoost and SVM for this models.

We split our merged data into test and train categories with X and y. X consists of the columns except 'Weekly_Sales' and y contains 'Weekly_Sales'. We train these models and obtain the accuracy score for both test and train data, the rmse value and mean absolute error. Depending upon all these, we choose up the best model.

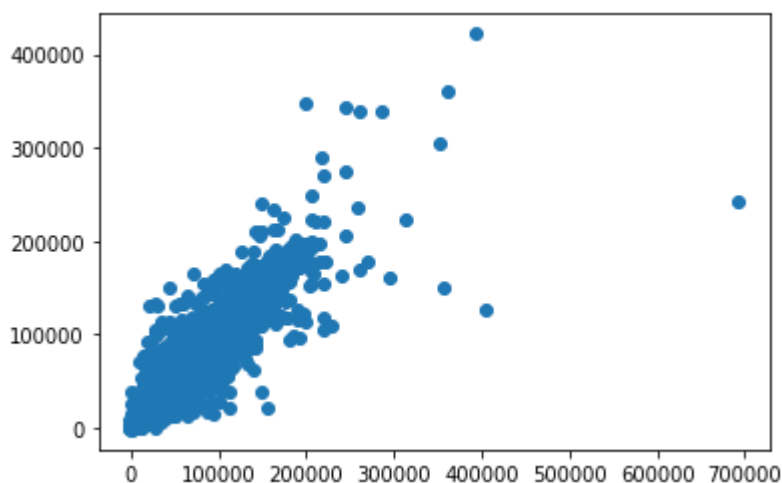
The plots of a few models are below.

K- Neighbours:



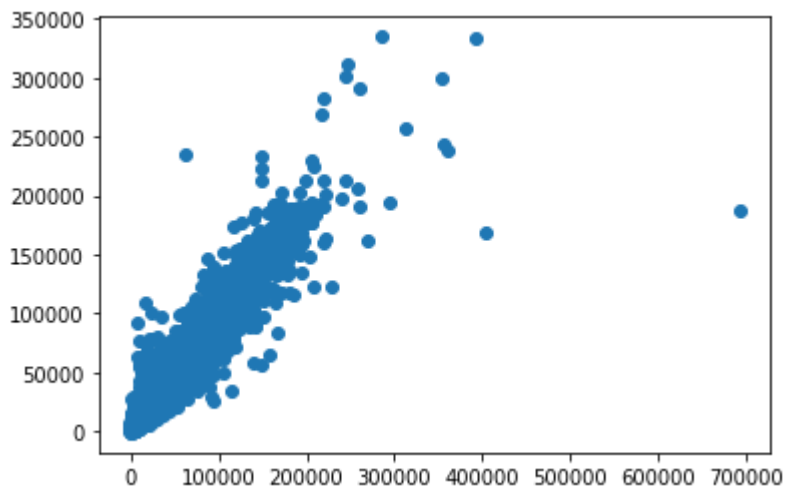
Train accuracy: 45 %

Decision Tree:



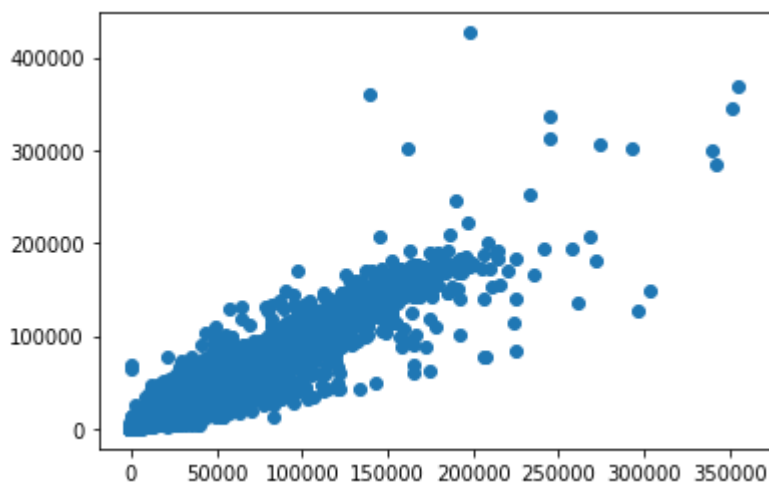
Train accuracy: 100%

Extra Tree Regressor:



Train Accuracy: 99.99%

Random Forest:



Train Accuracy: 97.8%

Of these models, the overall grids of test accuracy, MAE and RMSE are:

Model	Accuracy	MAE	RMSE
Linear Regression (Baseline)	9.210437565935193	12296.779325709611	15972.900772919114
KNNRegressor	32.318987843986235	12296.779325709611	15972.900772919114
DecisionTreeRegressor	97.40009315637927	1373.506702887852	2701.6763535430528
RandomForestRegressor	97.28522514630203	1529.5007584732932	2760.7134869720026
ExtraTreeRegressor	98.46175261448953	1103.731239985039	2078.105160563339
XGBRegressor	76.07683474335761	1103.731239985039	2078.105160563339

Here I tried Grid Search and shortlisted Extra tree Regressor and Random Forest Regressor.

After Grid search i got an accuracy of 96.78% and 96.18% for random forest and chose it to create pickle file.

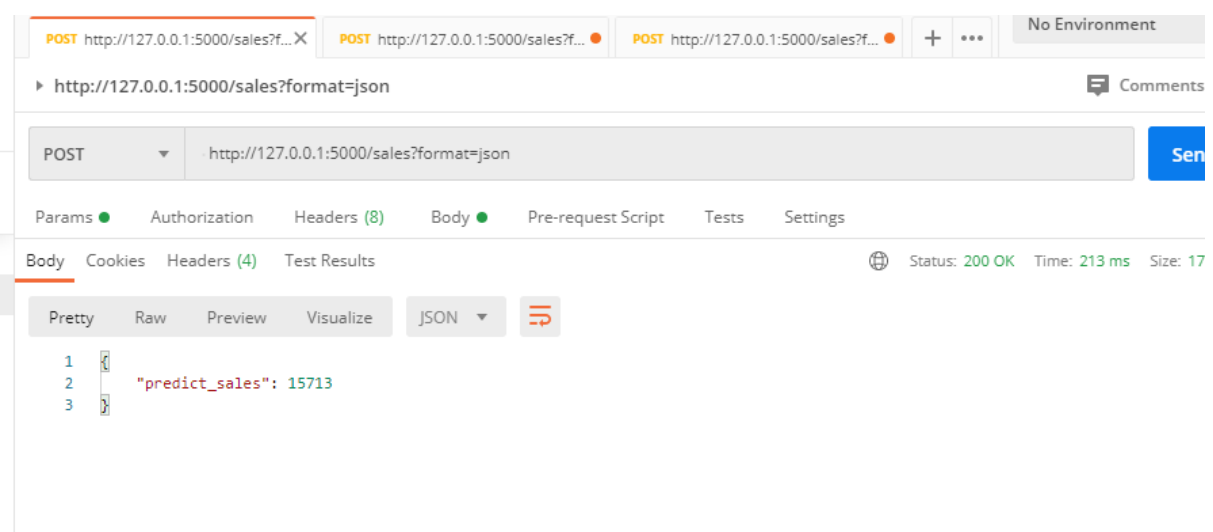
Compared to these two regressors, overfitting was less in Random forest and is generally more stable than the other. On the whole Random forest is better as there are more trees lesser overfitting.

Test Data Prediction:

We merge the features, stores and test data together and create a clean model by performing the feature removal and use this Random forest model to predict the Weekly_Sales. It is attached as Output.csv in the folder.

STEP 6: DEPLOYMENT

Once the pickle file is loaded, we create an app.py file using flask along with other files required to check with our own inputs. It is checked up in postman which enables us to input data in json format and check the prediction as the output. The screenshot is attached below.



Conclusion:

On the whole, prediction was done with an accuracy score of 96.17% without much overfitting. They need to focus on Weekend datas, Year end inventories and missing datas to get better accuracy. All these will help in better model predictions.

Improvements:

1. A Website which provides UI friendly experience
2. A model without any outliers and better classification experience.

Done by: Shanmuhapriyaa Raju.