Capstone Project - 2

Team 4 : Retail Sales Prediction

Content

1. Problem Statement
2. Data Summary
3. Analysis of Data
4. Feature Engineering
5. Data Preprocessing
6. Model Selection
7. Stacking

Problem Statement

1. Rossmann operates over 3000 drug stores in 7 European countries.
2. Rossmann Managers are tasked with predicting their sales for 6 weeks in advance.
3. The sales are influenced by many parameters and the task is to predict the sales based on the parameters.

Data Summary

The dataset spans over three years - 2013, 2014 and 2015.

Below are few important features:

1. **Customer : - The Number of customers on a given day in a store.**
2. **State Holiday :- Indicates a state holiday.**
3. **Store Type : Differentiate between 4 different store models.**
4. **Assortment : Describes an assortment level i.e a : basic, b : extra and c : extended.**
5. **Competition Distance : Distance in meters to the nearest competition store.**

Data Summary(Contd.)

7. CompetitionOpenSince[Year/Month] :- Gives the approximate year and month of the time the nearest competitor is opened.

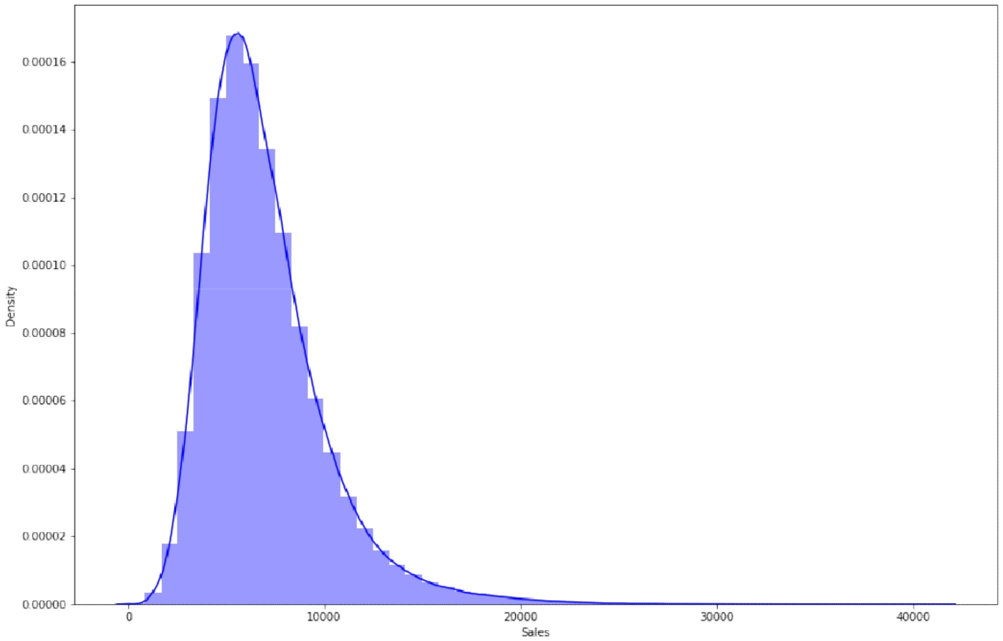
8. Promo :- Indicates whether a store is running a promo on that day.

9. Promo2 :- Indicates whether a store is continuing promotion.

10. Promo2Since[Year/Week] :- Gives the approximate year and calendar week of the time when the store started participating in Promo2.

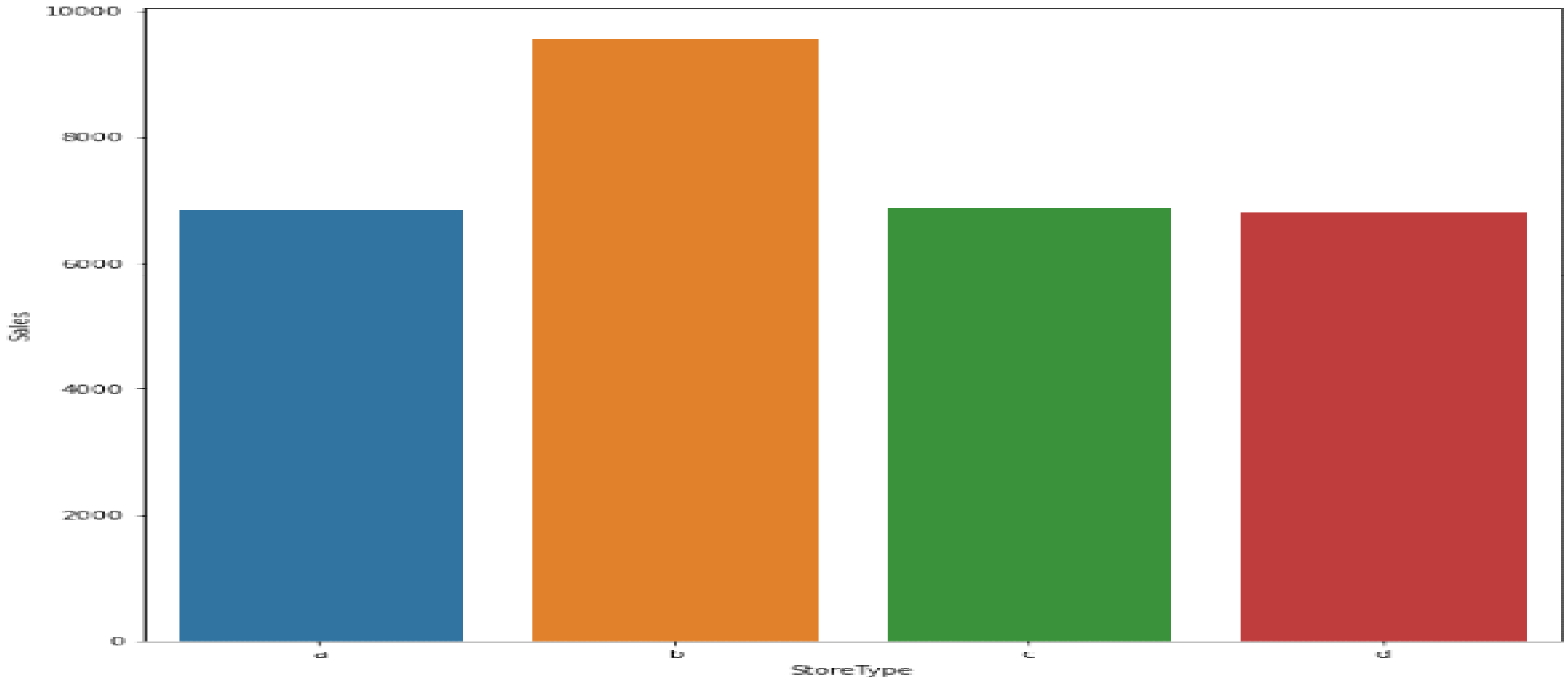
11. PromoInterval :- Describes an interval or name of months when the store runs Promo2.

Sales - Distribution

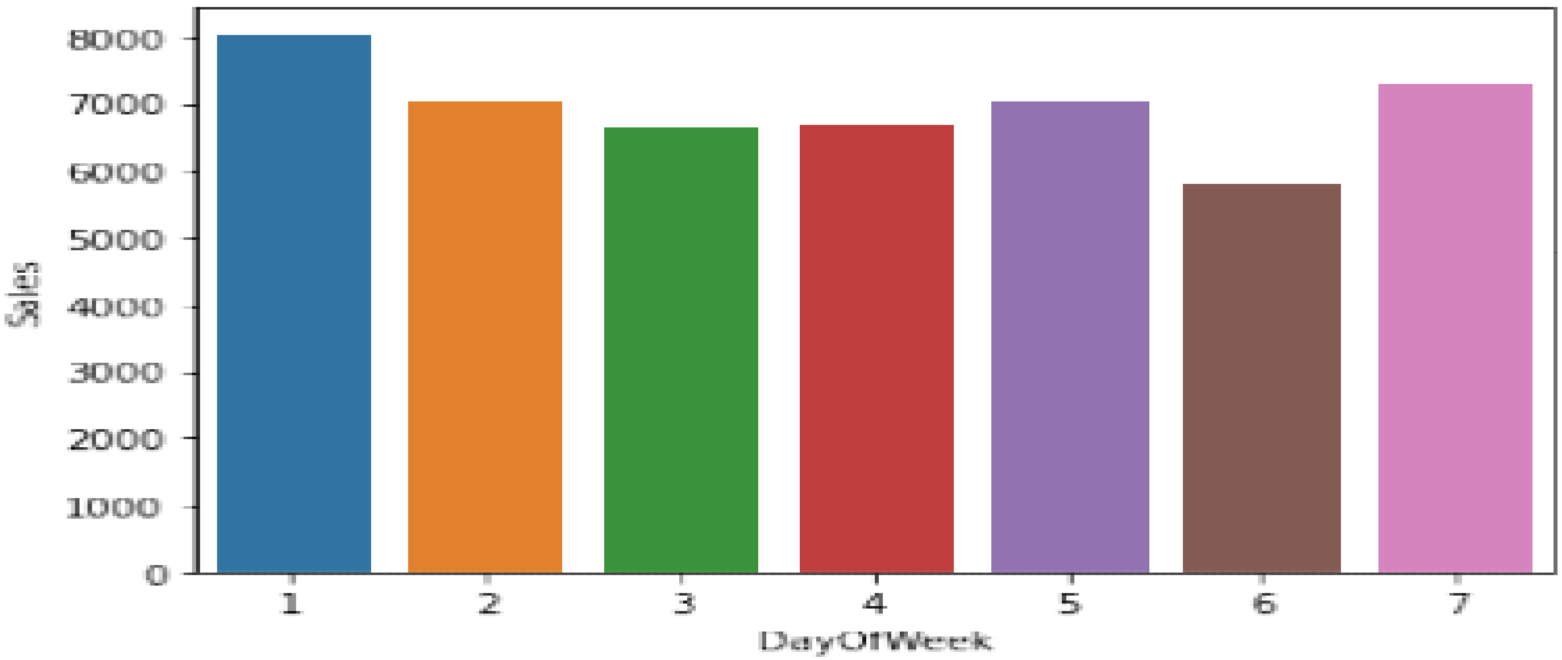


1. The Sales distribution lived up to the expectation with no irregularities.
2. It seems to be a perfect gaussian distribution with small positive skewness.

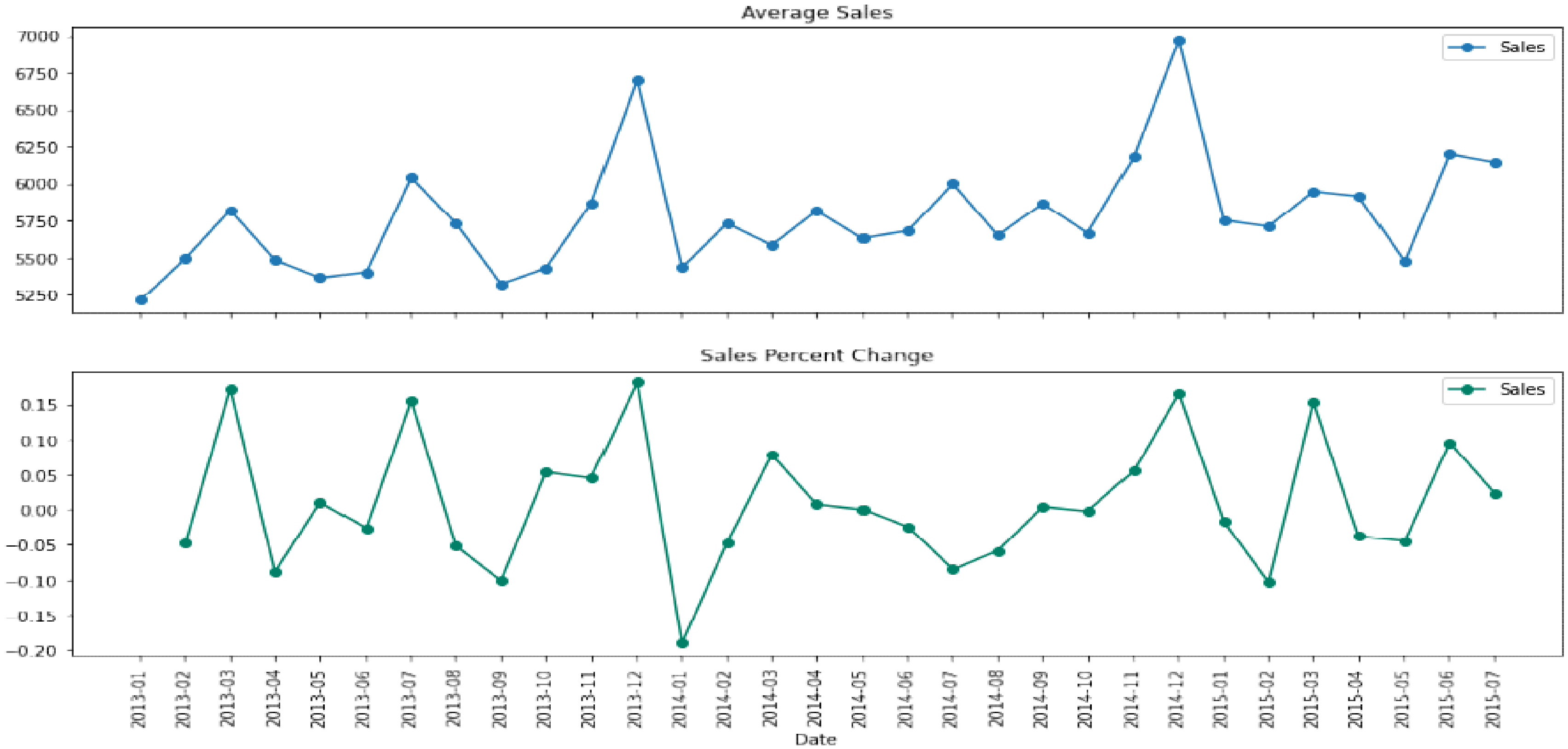
Sales VS Store Type



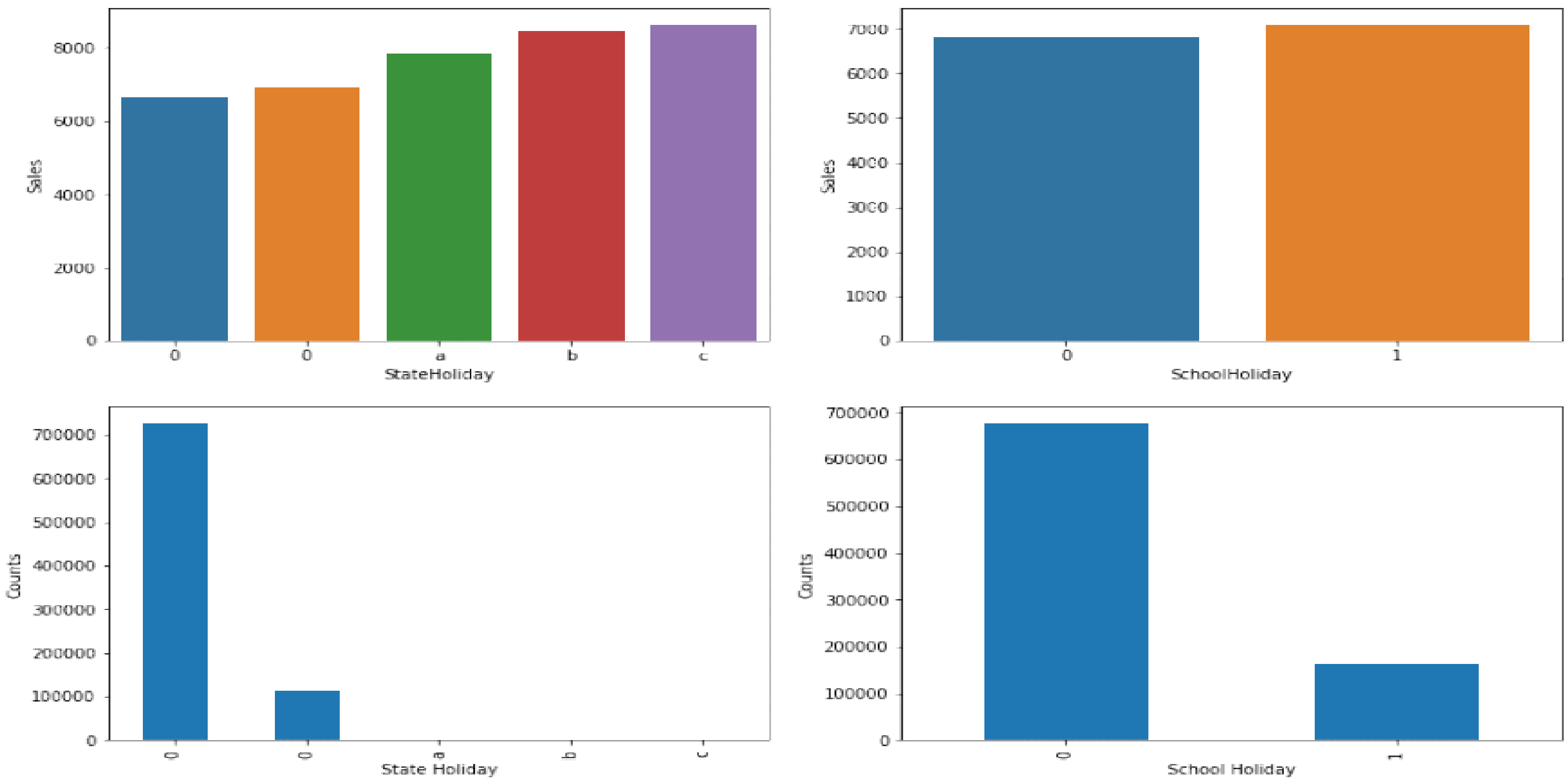
Weekly Sales Trend



Sales Trend over the years

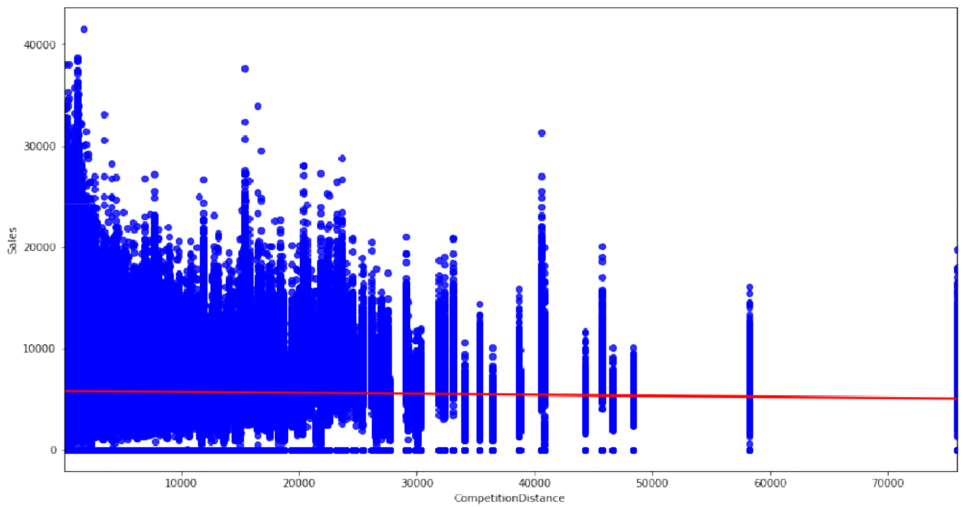


Holiday Sales

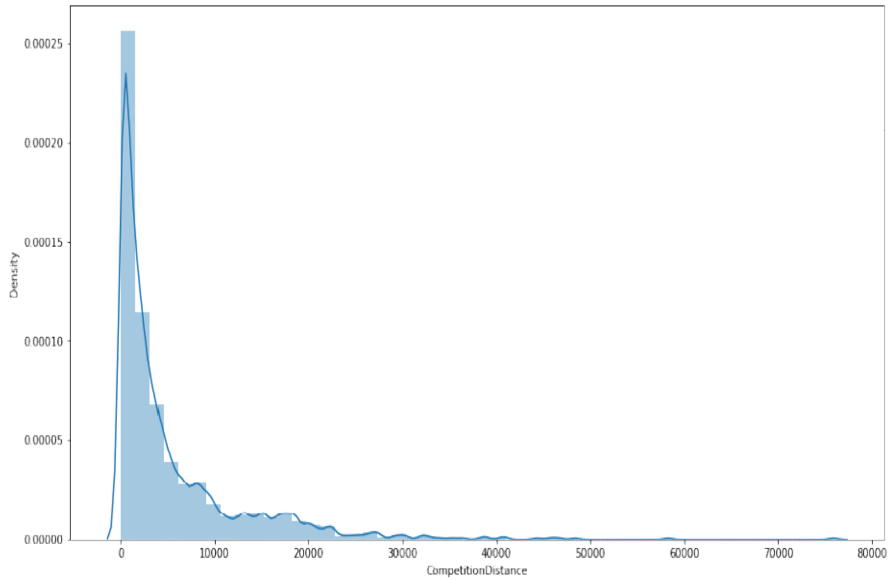


How Competition affects Sales

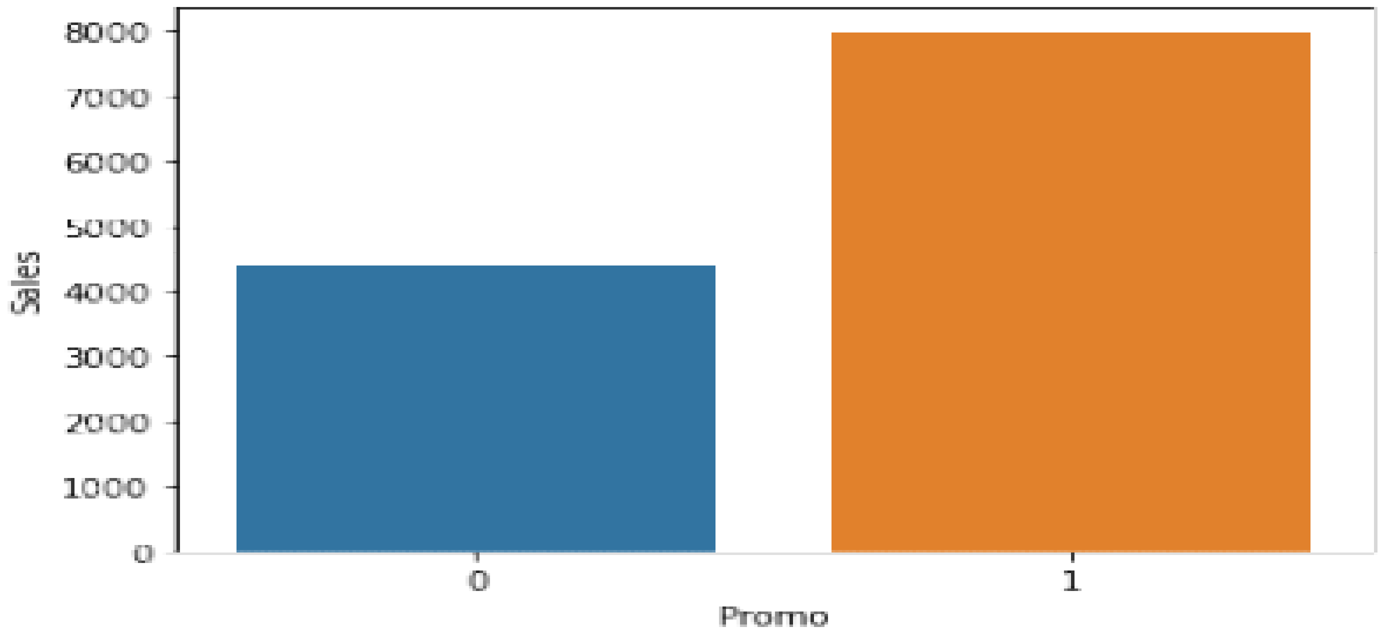
Sales VS Competition Distance



Competition Distance Distribution



Sales and Promotions



Sales are increasing because of Promotion. Let’s just go ahead with Promotion

EDA Conclusion

1. There are very few stores open on ‘State Holiday’ and they make a good profit on those days then any average day.
2. On School Holidays there is no large difference in sale. So promos running on School holidays can be reduced.
3. Sales for assortment type a and c seems to be less as compared to assortment type b.
4. At the start of month the sales increases. People might be planning to shop for the entire month in its beginning.

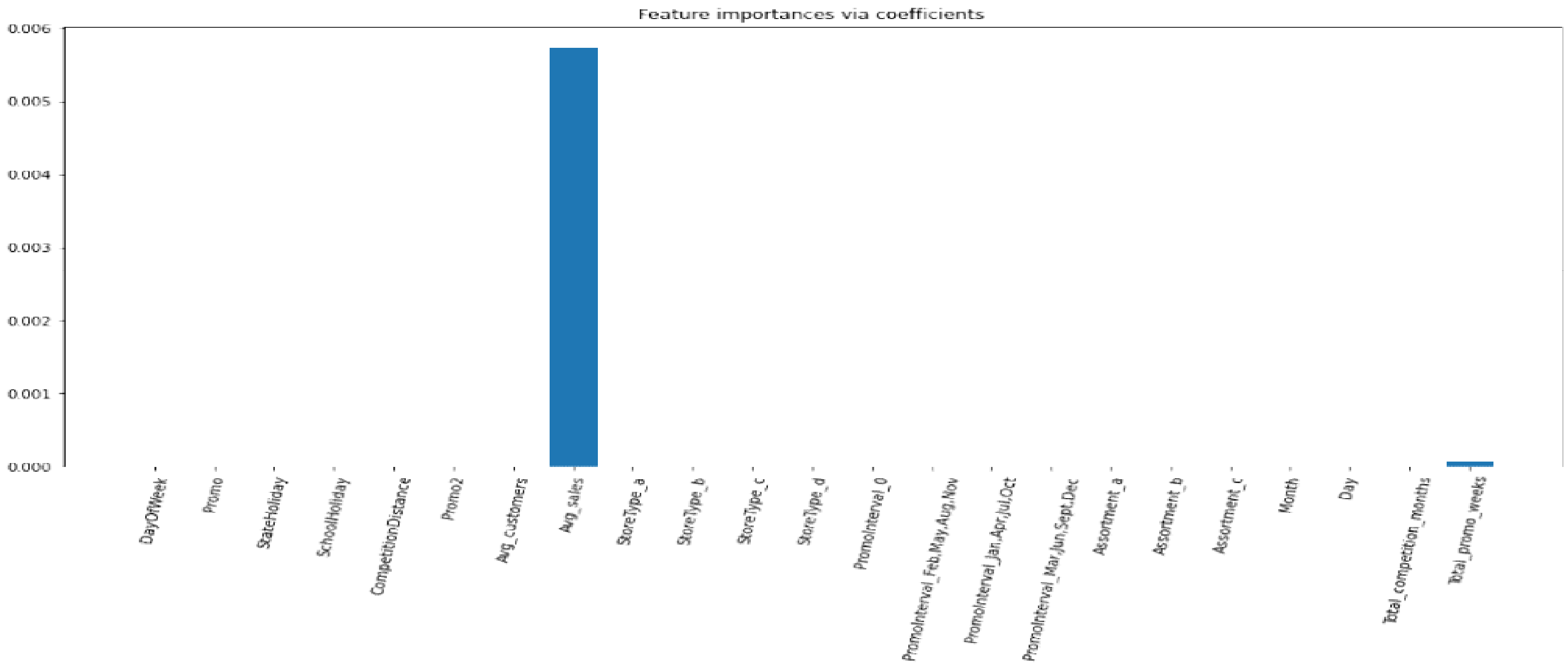
Feature Engineering

1. Extraction of Year, Month and Date from the Date column.
2. One hot encoding for Stateholiday, Storetype, Assortment and Promo Interval.
3. Creating Total Competition month as a new feature by using ‘CompetitionOpenSinceYear’ and ‘CompetitionOpenSinceMonth’.
4. Creating Total Promotion Year and Total Promotion Week as new features by using ‘Promo2SinceYear’ and ‘Promo2SinceWeek’.
5. Creating ‘IsPromoMonth’ as a new feature to account for whether a month is promotional or not using ‘PromoInterval’ feature.
6. Creating ‘Average Sales’ and ‘Average Customers’ columns and dropping the ‘Customers’ column.

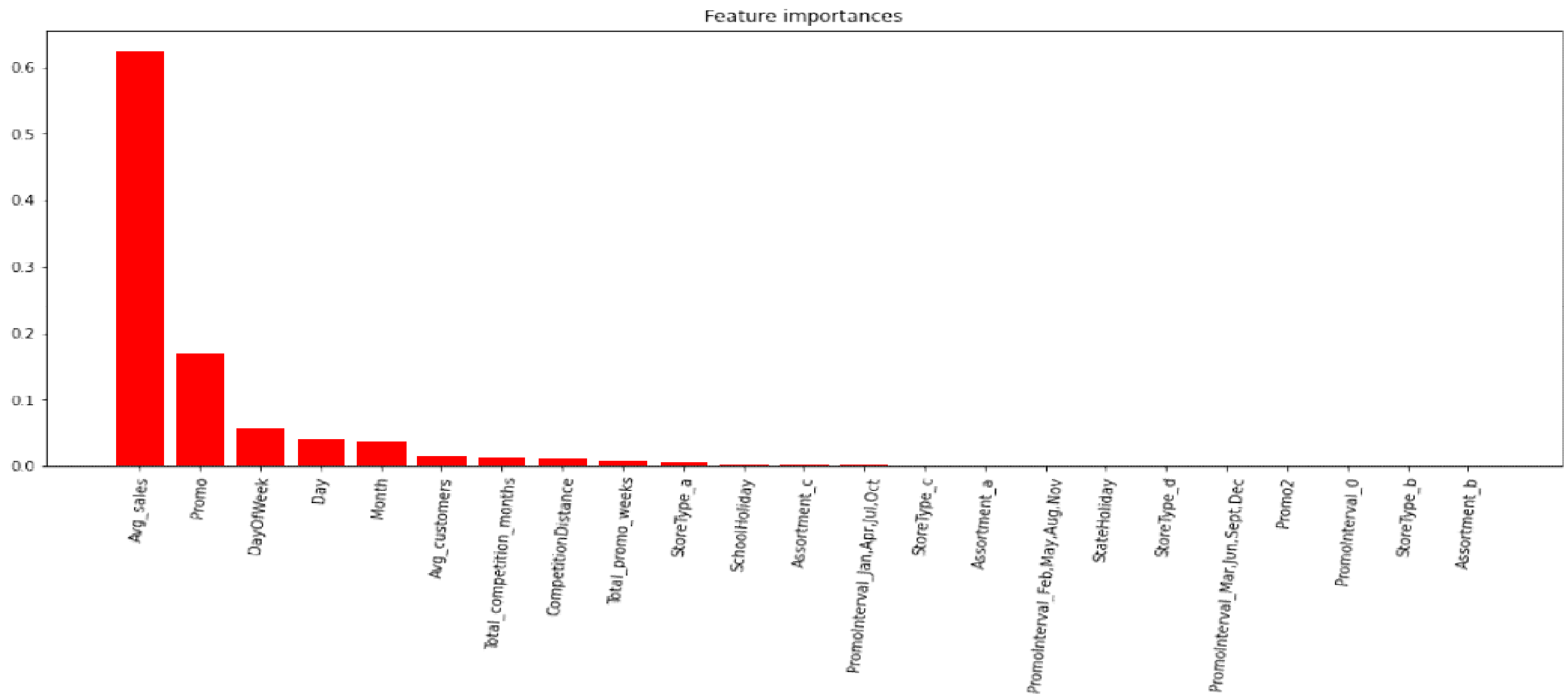
Models Used So Far For Prediction

1. Linear Regression (Baseline Model)
2. Decision Tree Regressor
3. Random Forest Regressor
4. Light GBM

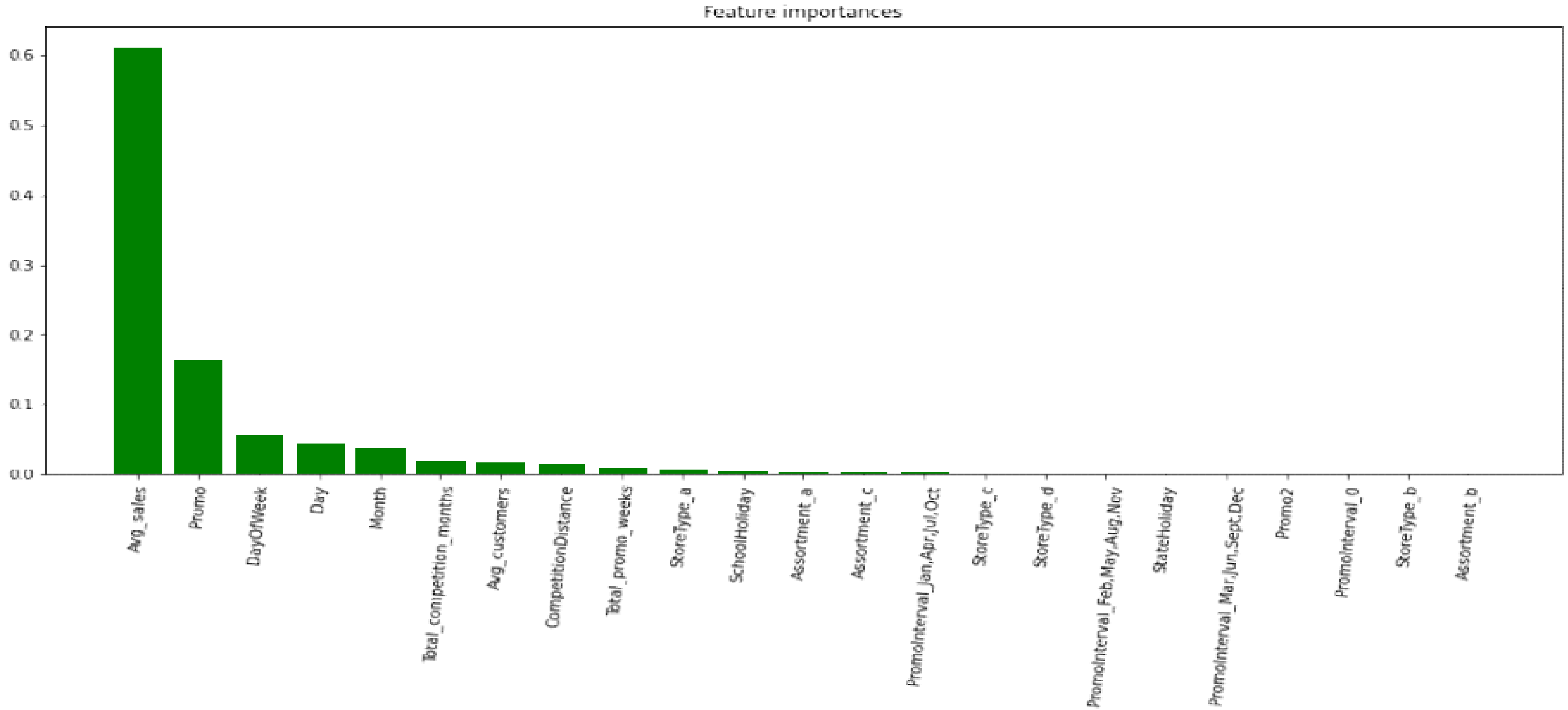
Feature Importance From Linear Regression



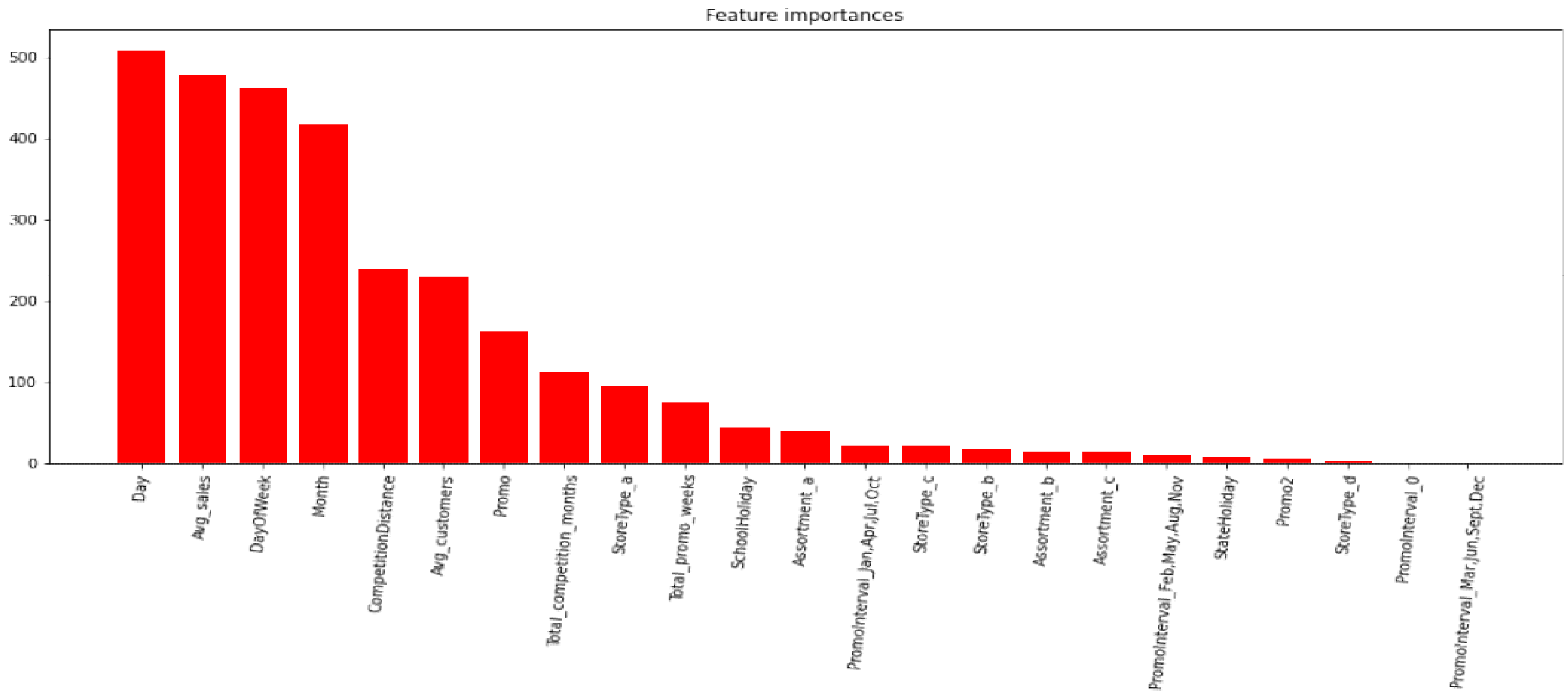
Feature Importance From Decision Tree



Feature Importance From Random Forest



Feature Importance From Light GBM



Let’s Stack...

1. Even though Random forest gave a 92% R2-Score, but was overfitting on the train dataset.
2. Decision Tree Regressor, Random Forest Regressor and Light GBM participated in stacking to overcome the issue of overfitting.
3. XGBoost Regressor has been used as meta learning algorithm.
4. Finally overfitting was resolved with stacking.

Evaluation of Models

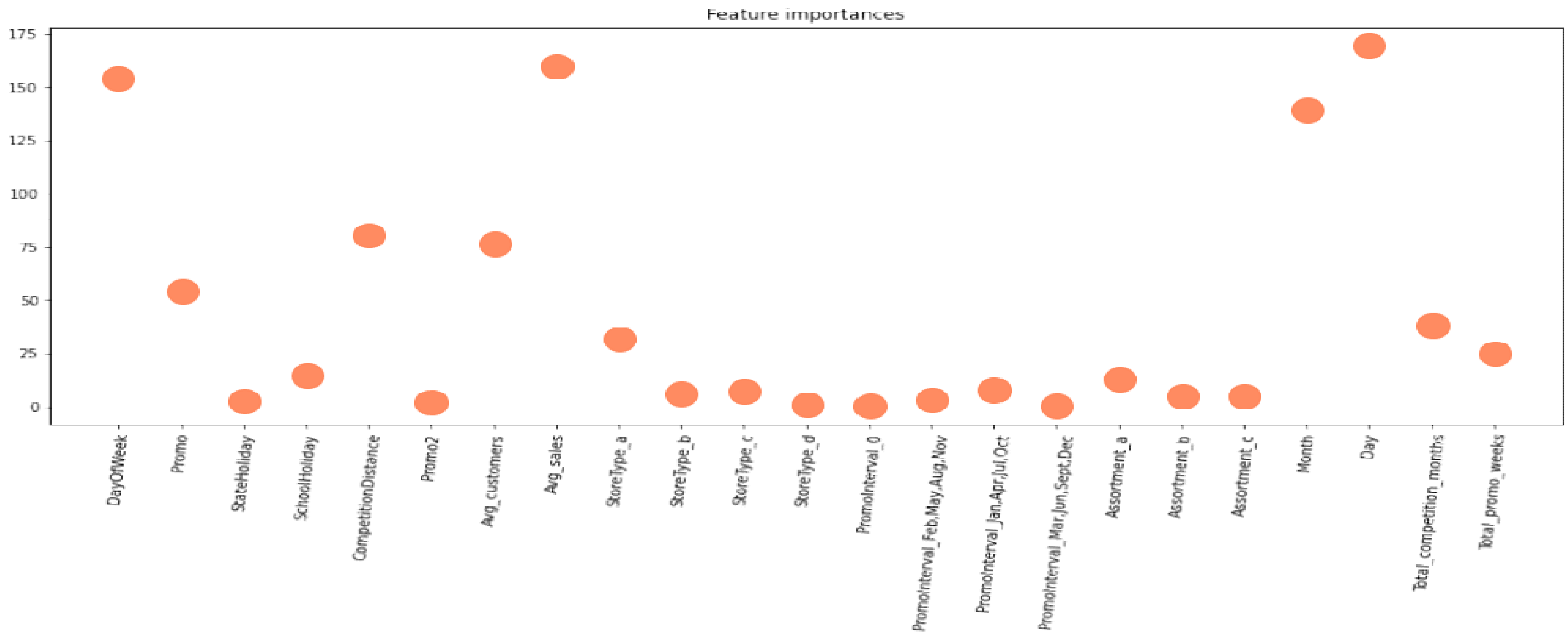
```
Result_df = pd.DataFrame({'Models':['Linear_regression', 'Decision Tree', 'Random Forest', '

# Model conclusion Df
Result_df.set_index('Models')
```

Models	Train R2 score	Test R2 score	Conclusion
Linear_regression	73.99	74.00	Simple base Model
Decision Tree	93.92	88.16	a bit of overfitting model
Random Forest	96.54	92.46	Good Accuraccy, but a bit a overfitting
LGB	87.69	87.77	Good Model but comparitively low accuracy
Stacked	92.31	92.48	Optimal model with good accuracy and best fit

```
feature_df = pd.merge(feature_importances_DTR, feature_importances_rf, left_index=True, right
feature_df = pd.merge(feature_df, feature_importances_LGB, left_index=True, right_index=True)
```

Conclusion for features



Challenges

1. Handling large amount of sales data (10,17,210 observations on 13 variable).
2. Prediction of sales of individual stores(out of 1115) and most of stores have different pattern of sales.

Conclusion

1. Our final optimal model would be the stack model as it resolves the issue of overfitting and gives us an R2- score of 92%.
2. Moreover getting lowest RMSE value for stacked model.
3. Applied only three model for stacking. So there are scope of applying more algorithms like SVM, Principal Component Regression.