Rossmann Store Sales By DP-The Sales Predictor

Bharadiya Pavan(MT2018023) and Devarakonda Deepak(MT2018031)

***Abstract*— Sales forecasting is a common topic in business.The task is to forecast the "Sales" for 1,115 stores located across Germany for a given day. Store sales are influenced by many factors. Our project aims to create a robust prediction model. We tried different models like linear regression,decision tree,random forest,XGBoost to predict sales for a given day. XGBoost turned out to be the best model. Then We tried stacking of random forest and XGBoost and tried to blend them with original features as well. Final selected model was stacking giving 0.05788 RMSPE on public leaderboard and 0.05602 on private leaderboard. We also observed that "Customers" column was the most important feature for sales prediction.**

I. INTRODUCTION

Rossmann operates over 3,000 drug stores in 7 European countries. Currently, Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance. Store sales are influenced by many factors, including promotions, competition, school and state holidays, seasonality, and locality. With thousands of individual managers predicting sales based on their unique circumstances, the accuracy of results can be quite varied. Competition is to predict sales for a given day for 1,115 stores located across Germany. Reliable sales forecasts enable store managers to create effective staff schedules that increase productivity and motivation.

The given features are mostly store related and there is only one column which is related to customer which shows the total customer visiting the store. As the target is continuous, the problem could be a regression problem based on both categorical feature(e.g, Store Type,Assortment,StateHoliday) and continuous feature(e.g, Days). Besides, some feature could be considered as categorical as well as continuous, for example, DayOfWeek could be considered as continuous by [1,2,3,4,5,6,7], assuming there is a relationship for adjacent days or as categorical [Mon, Tue, Wed, Thu, Fri, Sat, Sun],assuming there is no relationship between them.

II. DATASET

A. Data Collection

This project was hosted on kaggle as InClass competition where data is provided which contains four files - train.csv, store.csv, test.csv and samplesubmission.csv. you can find the data at https://www.kaggle.com/c/iiitb-ml-project-rossmann-store-sales/data

B. Description Of The Dataset

train.csv

This is the main training dataset which contains data about the sales figures for a store on a particular date.

Data Fields:

1. Store : a unique numerical store identifier (1 - 1,115)
2. DayOfWeek : the day of week (1 - 7)
3. Date : the date from 2013-01-01 to 2015-07-17
4. Sales : the turnover of a store on the specified date
5. Customers :the number of customers of a store
6. Open : (0 = Closed, 1 = Open)
7. Promo : indicates promotion(0 = No, 1 = Yes)
8. StateHoliday :shows type of state holiday
9. SchoolHoliday : (0 = No, 1 = Yes)
10. Id : Unique id for all data
No. of Records: 1,000,000

store.csv

This dataset contains supplementary information about each of the 1,115 stores and helps identify unique features which may (or may not) affect sales.

Data Fields:

1. Store : a unique numerical store identifier (1 - 1,115)
2. StoreType : 4 different types of stores (a, b, c, d)
3. Assortment : describes the assortment of goods (a = Basic, b = Extra, c = Extended)
4. CompetitionDistance : the distance (in metres) to the nearest competitors store
5. CompetitionOpenSinceMonth : the month in which the competition opened
6. CompetitionOpenSinceYear : the year in which the competition opened
7. Promo2 : indicates if a store is participating in a continuing and consecutive promotion(0 = No, 1 = Yes)
8. Promo2SinceWeek : the week of the year in which the store began participating in Promo2 (from 1 - 50, presumably, but some weeks are unrepresented in the data)
9. Promo2SinceYear : the year in which the store began participating in Promo2 (from 2009 - 2015)
10. PromoInterval : describes the consecutive intervals in which Promo2 is activated, giving the months the promotion is renewed (either Jan,Apr,Jul,Oct, Feb,May,Aug,Nov or Mar,Jun,Sept,Dec)
No. of Records : 1,115

test.csv

This dataset is to be used for testing and evaluating the model. Data Fields: Same as train.csv, with the exclusion of Sales.

No. of Records : 1115

III. Preprocessing The Data

1) ***Open column****:* Removing all those entries where store is closed as sales is zero.

2) ***CompetitionDistance****:* Replacing NaN values with 0 since no record was there.

3) ***CompetitionSince[X]****:* Replacing all the NaN values with 0 for for CompetitionSinceYear and CompetitionSinceMonth respectively.

4) ***Promo2Since[X]****:* Replacing all the NaN values with 0 for for Promo2SinceYear and Promo2SinceWeek respectively.

5) ***StateHoliday****:* Converting all 0(integer) to "0"(String) because data in the StateHoliday is a mix of numerical and string values. Hence, for the sake of consistency,all values were converted to string.

6) ***Label Encoding****:* we performed label encoding on the categorical features present in our dataset, including StateHoliday in train.csv and test.csv and StoreType and Assortment in store.csv.

IV. Exploratory Data Analysis

The following subsections are trying to analyze dataset and figure out useful features that can be used to forecast sales. At first, we will attempt to extract features from training dataset. Then, in order to get more useful features, the store information will be reviewed. At last, we will try to get more information from what we have now based on store information.

A. Correlation Between Training data

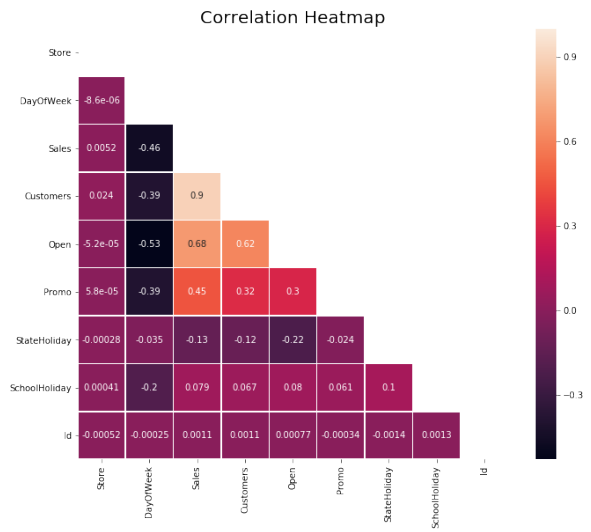

Fig. 1. correlation matrix

1) ***Store ID****:* It is customary to think the store ID as one feature because sales may change from store to store. However, if we just use the Store ID as one feature, we will find that the correlation coefficient between Store ID and Sales is only 0.0052.

2) ***Day of Week****:* Its also easy to think that in different day of week, every store will have different sales since people get used to shop in different days. The day of week seems to have large effect on sales as we can see that the correlation coefficient is -0.46.

3) ***Number of Customers****:* In the training dataset which represents the past information includes the numbers of customers. This is also present in test.csv and it is very highly correlated with sales. We can see that the correlation coefficient is 0.9.

4) ***Open****:* Open shows whether this store is open or not in a specified day. Because the sales must be 0 when the store is closed, we removed the data point with "Open = 0" and after prediction, we would set the value of sales as 0 for the data point with Open = 0 in testing data. We can see that the correlation coefficient is 0.68.

5) ***Promo****:* It indicates whether a store is running a promo on that day. As promotion would be a attracting thing for customer, it might have effect on the sales.So correlation coefficient is 0.45.

B. Frequency Distribution

1) Frequency Of Sales

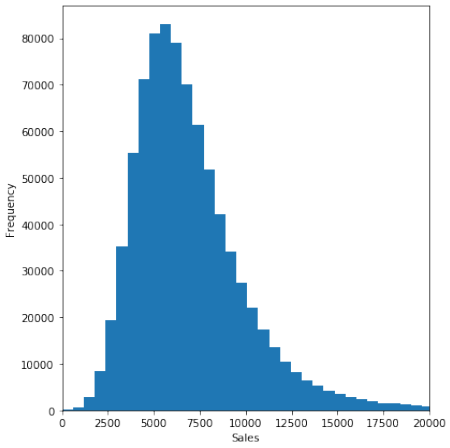

Fig. 2. frequency of sales

By looking at the chart above, we can see that the Sales values forms a Poisson-like distribution, reaching its peak of frequency between 5,000 and 7,500. At the same time, it is uncommon for Sales to fall below 2,500 (on open days) or go beyond 12,500.The average value for Sales for open stores is 6,914.95.

2)Frequency Of Customers

Similarly, the distribution of Customers values follows a Poisson-like distribution, indicating that the number of customers commonly ranges between 500 and 1,000, while

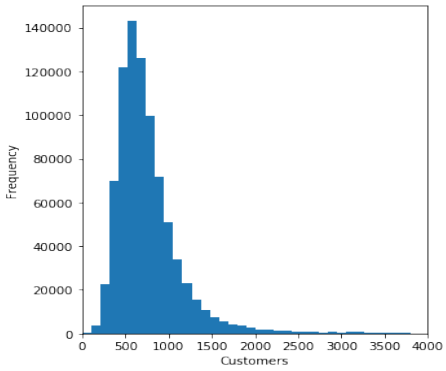

Fig. 3. frequency of customers

a value higher than 1,000 has relatively low frequency. The average value for Customers for open stores is 762.56.

C. Possible Outliers

We can check the max,min and average values of sales which shows that minimum value = 527, maximum value = 38722, mean value is = 6914 and 75 percetile = 8209 which is very less than max sales.

Sales $>$ 35,000 = 7

Sales $>$ 30,000 = 48

Sales $>$ 25,000 = 308

This suggests that Sales values greater than 30,000 can be safely ignored to avoid overfitting as those records are extremely rare and only occur in a handful of stores but could affect the models predictions for the other stores.

D. Trends For DayOfWeek

1) mean(Sales) v/s DayOfWeek

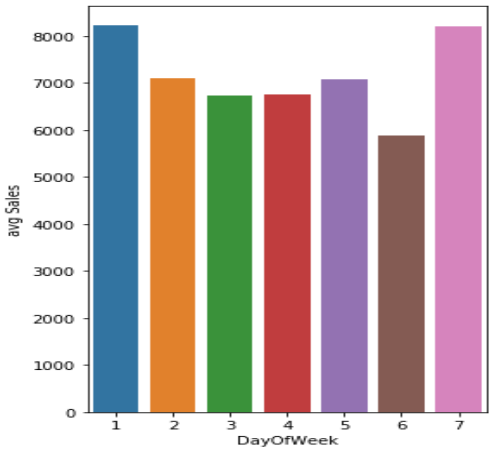

Fig. 4. sales for all Days of week

2) Customers v/s DayOfWeek

From the above graphs we can see that, there are three peaks in the average sales and average no. of customers in the week. The first two are on Mondays and Fridays, which is probably due to these days being the start and end of the week. The highest peak, however, is on Sundays,

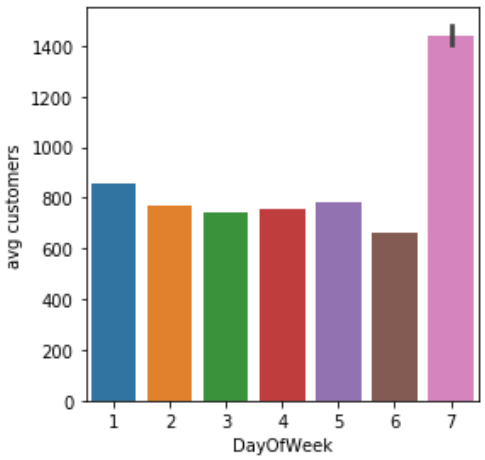

Fig. 5. mean(customers) for all days of week

which is probably due to the fact that most other stores are closed. An interesting observation here is that there is a larger difference between the average no. of customers and average sales on Sundays as compared to other days of the week. While the average sales is approximately the same on Sundays and Mondays, there is a drastic difference in the no. of customers, which hints at a large number of window shoppers on Sundays.

E. Monthly trends

1)Monthly mean(Sales)

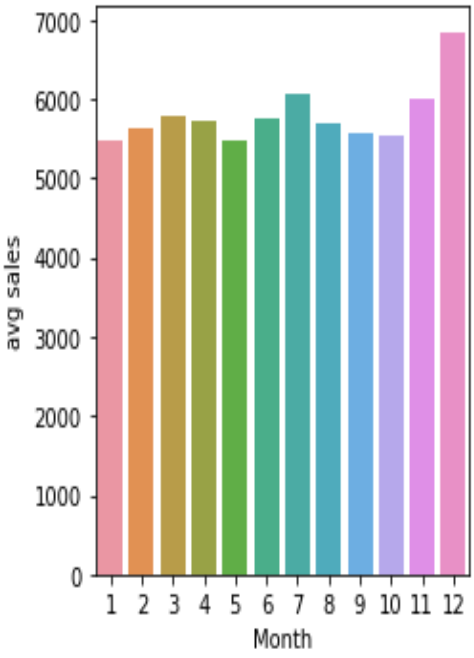

Fig. 6. mean(sales) for all months

2)Monthly mean(Customers)

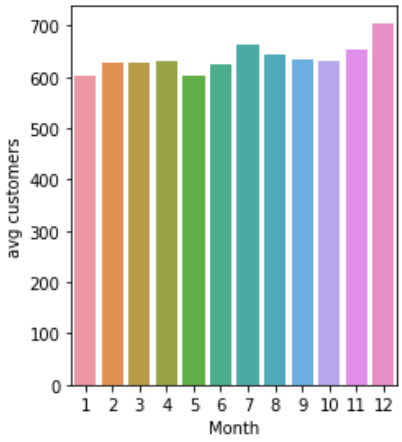

Fig. 7. mean(customers) for all months

When plotting the average sales and no. of customers by month, we see an annual trend similar to most retail stores. There is an uptick in the months of June - July owing to summer holidays and another bigger uptick in December, which is likely due to the holiday season.

F. Average percentage change in sales annually

The annual trends are further corroborated when plotting the average sales data per month for the entirety of the training dataset (January 2013 to July 2015). We see the same annual trends being followed in the 2.5 years, with increasing peak values every year hinting at improved store performance from 2013 to 2014 to 2015.

G. Effect Of Promo

1) Effect Of Promo On Sales

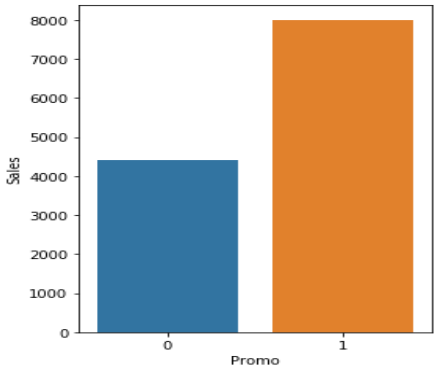

Fig. 8. sales for promo

2) Effect Of Promo On Customers

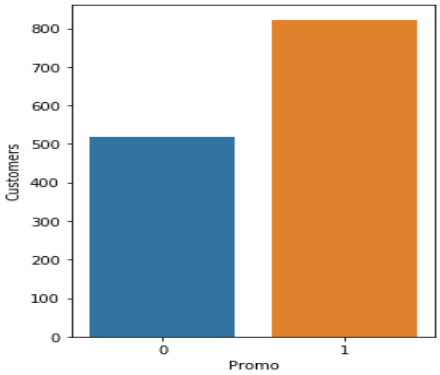

Fig. 9. Customers for promo

The figure below shows the average customers and average sales (across all stores) on days with and without promotions. The effect of having a promotion on sales and customers is clearly evident, with the promotion having a strong positive effect on both.

H. Effect Of StateHoliday

1)Total state holidays of different types

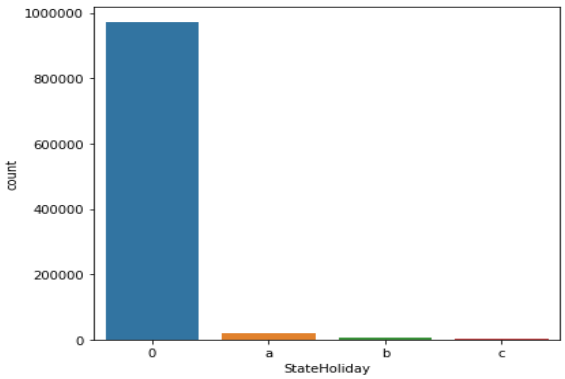

Fig. 10. different types of state holidays

2)Sales and Customers on state holiday

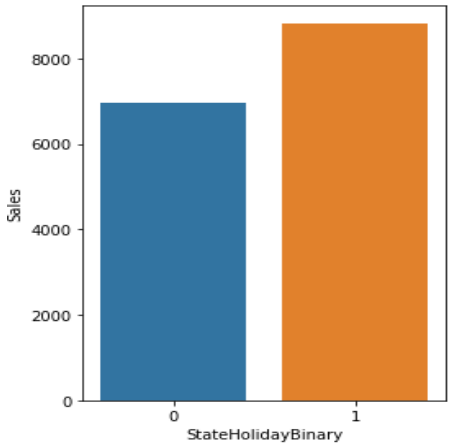

Fig. 11. Sales of BinaryStateHoliday

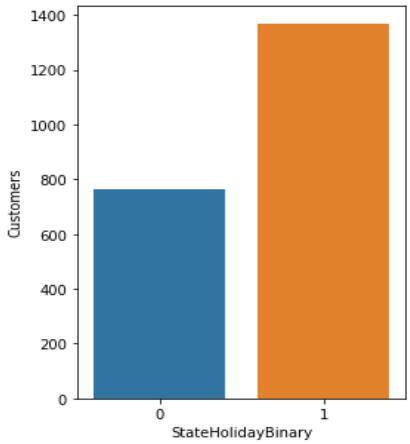

Fig. 12. Customers of BinaryStateHoliday

from the first figure we can see that state holiday = 0 means no holiday which has most of the entries. Since most stores are closed on public holidays and closed stores can be ignored when making predictions, the average sales figures and average number of customers on state holidays is plotted, but only for open stores in the figure below. It is evident that when a store is open on state holidays, it attracts more customers and accrues more sales.

V. Feature Engineering

Initially We merge train and Store csv files and do feature engineering on them as both files contains valuable information.

1) ***SalesPerCustomer****:* As the sales and customers are highly correlated, we can create new feature called SalesPerCustomer.

2) ***AvgSales****:* Group by the Store and calculate the mean value of sales for all the store and merge this new dataframe with store.csv using 'Store' as index and use as a new feature.

3) ***AvgCustomers****:* As customers are also very important we also group them by the Store and calculate the mean value of customers for all store and merge this new dataframe with store.csv using 'Store' as index and use as a new feature.

4) ***AvgSalesCustomers****:* Do the same thing for AvgSalesPerCustomer.

In the same way, instead of using mean we use meadian to calculate three more features called

5) ***MedSales****:* Using median values of sales for all store

6) ***MedCustomers****:* Using median values of customers for all store

7) ***MedSalesPerCustomers****:* Using median values of sales per customer for all store

8) ***Year****:* Using Date column

9) ***Month****:* Using Date column

10) ***DayOfMonth****:* Using Date column

11) ***DayOfYear****:* Using Date column

12) ***WeekOfYear****:* Using Date column

13) ***CompetitionOpen****:* We converted CompetitionOpenSinceYear and CompetitionOpenSinceMonth to all Months by doing CompetitionOpenSinceYear*12 - CompetitionOpenSincemonth and stored it in CompetitionOpen

14) ***WeeksPromoOpen****:* We converted Promo2SinceYear and Promo2SinceWeek to all weeks and by doing 12 * (Year - Promo2SinceYear) + (Date.weekofyear - Promo 2SinceWeek) / 4.0 and stored in WeeksPromoOpen

15) ***IsPromoMonth****:* This shows the current month is promo month or not. It is calculated using PromoInterval column.

Along with these 15 features some other features which we used directly are

16) ***Store****:* Denotes store from 1 - 1115

17) ***Customers****:* Customers for that store

18) ***CompetitionDistance****:* Distance to competitors

19) ***Promo****:* Indicates promo is there or not

20) ***Promo2****:* indicates if a store is participating in a continuing and consecutive promotion

These are the all features we are using to train our model.

A. Effect Of CompetitionDistance on AvgSales and AvgCustomers

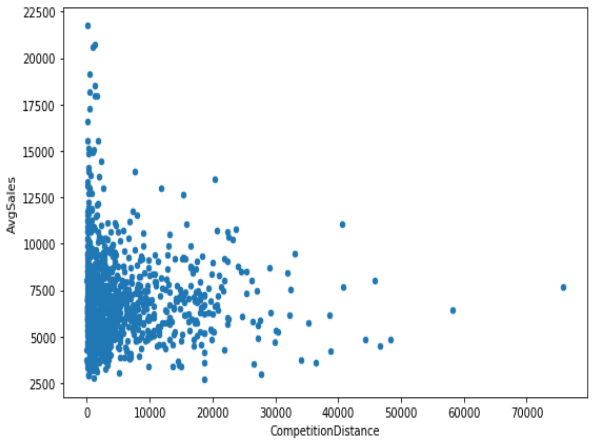

Fig. 13. distance vs. AvgSales

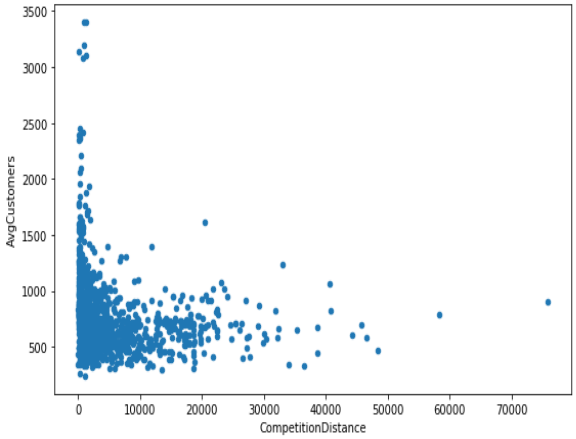

Fig. 14. distance vs. AvgCustomers

From above graphs we can see that higher average sales and average customer figures are achieved when CompetitionDistance equals 0, or in other words, there is no nearby competition.

B. Variation of AvgSales and AvgCustomers with StoreType

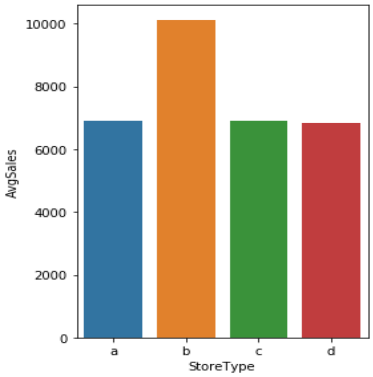

Fig. 15. StoreType vs. AvgSales

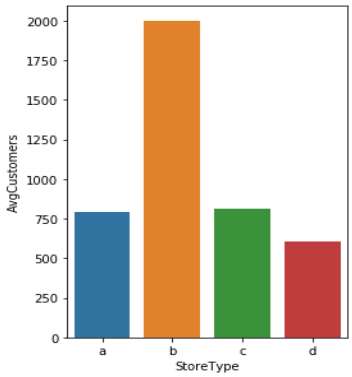

Fig. 16. StoreType vs. AvgCustomers

A plot of the average sales and average customers of each store type shows that there is a strong correlation between the store type and average sales.

VI. Model Selection

We used different models such as linear regression,decision tree regressor,random forest regressor and XGBoost regressor. We also tried blending and stacking of models. The RMSPE for the following models on our machine on validation data and on public leaderboard are as follows:

Linear Regression:
We got RMSPE = 0.085
public leaderboard score = 0.087

DecisionTreeRegressor:
We got RMSPE = 0.069
public leaderboard score = 0.072

RandomForestRegressor:
We got RMSPE = 0.058
public leaderboard score = 0.065

XGBoostRegressor:
We got RMSPE = 0.054
public leaderboard score = 0.062

Blending random forest and xgboost with features and using xgboost:
We got RMSPE = 0.055
public leaderboard score = 0.59

Stacking random forest and xgboost and applying xgboost:
We got RMSPE = 0.054
public leaderboard score = 0.057

For stacking and blending, we tried different splitting of the data. We tried training first random forest and XGBoost with 50%,60%,70% of the data and then trained XGBoost again on the test_size with previous model's output to predict for the test.csv. We got best results by training on 60% data initially and training XGBoost again on remaining 40% of the data.

We tuned parameters for Random Forest and XgBoost using GridSearchCV. The parameters used for random forest are as follows.

randomForest=RandomForestRegressor(n_estimators=100, max_depth=100, min_samples_leaf=1, min_samples_split=20, bootstrap=True, verbose=1, n_jobs=-1)

The parameters used for XGBoost are as follows:

xgb.XGBRegressor(n_jobs=-1,n_estimators=4000,learning_rate=0.1,max_depth=2,min_child_weight=2,subsample=0.8,colsample_bytree=0.8,tree_method='exact', reg_alpha=0.05,silent=0,random_state=1023)

VII. Feature Importance Plots

A. Random Forest

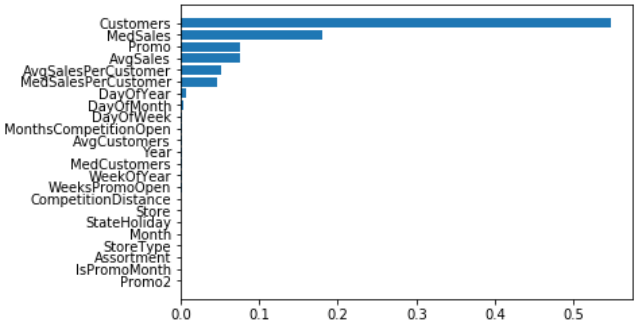

Fig. 17. random forest feature importance

The most important features for random forest are Customers,MedSales,AvgSales,AvgSalesPerCustomer, DayOfYear,DayOfMonth.

B. XGBoost

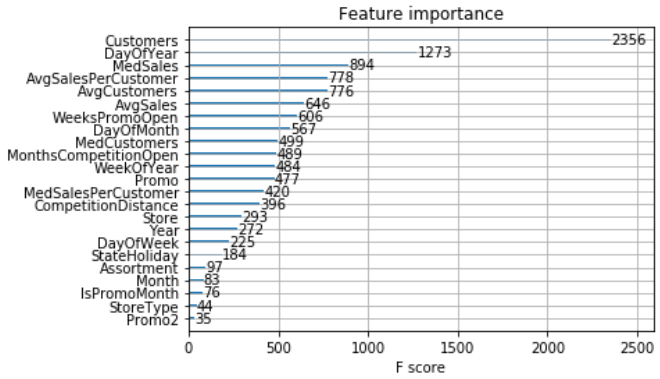

Fig. 18. XGBoost feature importance

The most important features for XGBoost are Customers,DayOfYear,MedSales,AvgSalesPerCustomer, AvgCustomers,AvgSales,WeeksPromoMonth,DayOfMonth.

C. Stacking of Random Forest and XGBoost

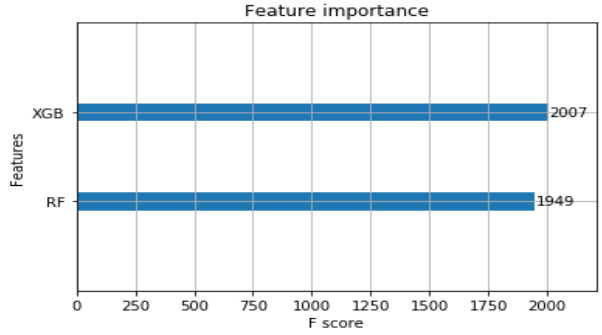

Fig. 19. Stacking feature importance

It is giving more importance to the XGBoost as shown in above figure.

VIII. Conclusions

We selected stacking and blending models as our two final models because they performed well on public leaderboad. Stacking model performed better on private leaderboard giving RMSPE of 0.05602 and using that we finished on third position. We observed that when the store is closed, value for Sales is 0. So for the test.csv,we predicted 0 when store was closed. The most important feature is Customer as it is highly correlated with Sales. Other important features were MedSales,DayOfYear,DayOfmonth,AvgSalesPerCustomers, WeeksPromoOpen. On the private leaderboard, we observed that our best score was RMSPE=0.05488 which was using only XGBoost but we didn't selected that as our final model to be evaluated as other models were performing better than only XGBoost.

References

[1] https://www.kaggle.com/c/rossmann-store-sales.
[2] https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/
[3] https://machinelearningmastery.com/xgboost-python-mini-course/
[4] https://seaborn.pydata.org/generated/seaborn.countplot.html?highlight=countplots
[5] https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.