Capstone Project

Rossmann Sales Prediction

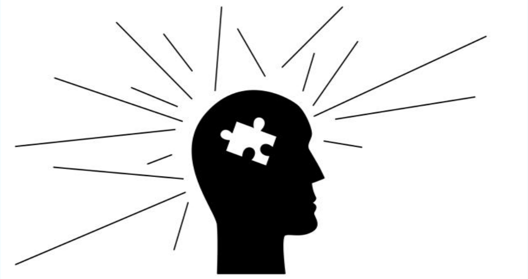
ML Supervised Regression

Individual Project:
Priyanka Shinde

Problem Statement:

- Rossmann store managers are tasked with predicting their daily sales for up to six weeks in advance.

We were provided with historical sales data for 1,115 Rossmann stores and were asked to predict the sales.



Description of data

AI

We are provided with 2 data sets:

1. Rossmann Stores Data.csv - This dataset includes the historical data including Sales. This dataset contain features like Sales, Customers, Open, StateHoliday, SchoolHoliday.

2. store.csv - This includes supplemental information about the stores.This dataset contain features like Assortment, CompetitionDistance, CompetitionOpenSince[Month/Year].

After merging two dataset on “store” column.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1017209 entries, 0 to 1017208
Data columns (total 18 columns):
 #   Column                     Non-Null Count    Dtype
---  ------                     --------------    -----
 0   Store                      1017209 non-null  int64
 1   DayOfWeek                  1017209 non-null  int64
 2   Date                       1017209 non-null  object
 3   Sales                      1017209 non-null  int64
 4   Customers                  1017209 non-null  int64
 5   Open                       1017209 non-null  int64
 6   Promo                      1017209 non-null  int64
 7   StateHoliday               1017209 non-null  object
 8   SchoolHoliday              1017209 non-null  int64
 9   StoreType                  1017209 non-null  object
 10  Assortment                 1017209 non-null  object
 11  CompetitionDistance        1014567 non-null  float64
 12  CompetitionOpenSinceMonth  693861 non-null   float64
 13  CompetitionOpenSinceYear   693861 non-null   float64
 14  Promo2                     1017209 non-null  int64
 15  Promo2SinceWeek            509178 non-null   float64
 16  Promo2SinceYear            509178 non-null   float64
 17  PromoInterval              509178 non-null   object
dtypes: float64(5), int64(8), object(5)
memory usage: 147.5+ MB
```

Data Exploration

- In below table each row of dataframe shows the daily sales for each store.

index	0	1	2
Store	1	1	1
DayOfWeek	5	4	3
Date	2015-07-31	2015-07-30	2015-07-29
Sales	5263	5020	4782
Customers	555	546	523
Open	1	1	1
Promo	1	1	1
StateHoliday	0	0	0
SchoolHoliday	1	1	1
StoreType	c	c	c
Assortment	a	a	a
CompetitionDistance	1270	1270	1270
CompetitionOpenSinceMonth	9	9	9
CompetitionOpenSinceYear	2008	2008	2008
Promo2	0	0	0
Promo2SinceWeek	NaN	NaN	NaN
Promo2SinceYear	NaN	NaN	NaN
PromoInterval	NaN	NaN	NaN

Missing Values in the Datasets

index	Missing values count
Store	0
DayOfWeek	0
Date	0
Sales	0
Customers	0
Open	0
Promo	0
StateHoliday	0
SchoolHoliday	0
StoreType	0
Assortment	0
CompetitionDistance	2642
CompetitionOpenSinceMonth	323348
CompetitionOpenSinceYear	323348
Promo2	0
Promo2SinceWeek	508031
Promo2SinceYear	508031
PromoInterval	508031

- **Dataset contains lots of NaN values.**

Data Preprocessing

- Initially the data type of Date was in string format, i have used pandas DatetimeIndex function to convert the data type

- Created month , day, year ,week_of_year columns based on Date column

- Similarly we created CompetitionOpenSince from CompetitionOpenSinceMonth and CompetitionOpenSinceYear.

Year	Month	Weekofyear	Day	CompetitionOpenSince
2015	7	31	31	b
2015	7	31	30	b
2015	7	31	29	b
2015	7	31	28	b
2015	7	31	27	b
2015	7	30	26	b
2015	7	30	25	b
2015	7	30	24	b

Dependent Variable

- The graph of sales distribution shows a positively skewed distribution, and some data falls below 0 because some stores are closed.

- Concerning the 16.99% (the outliers) of the time having 0 sales because some stores are closed.

Visualization

- **0 = Store not runing promo, 1=Store runing promo**

Visualization

	StoreType	Sales	Customers
0	a	3165334859	363541434
1	b	159231395	31465621
2	c	783221426	92129705
3	d	1765392943	156904995

Visualization

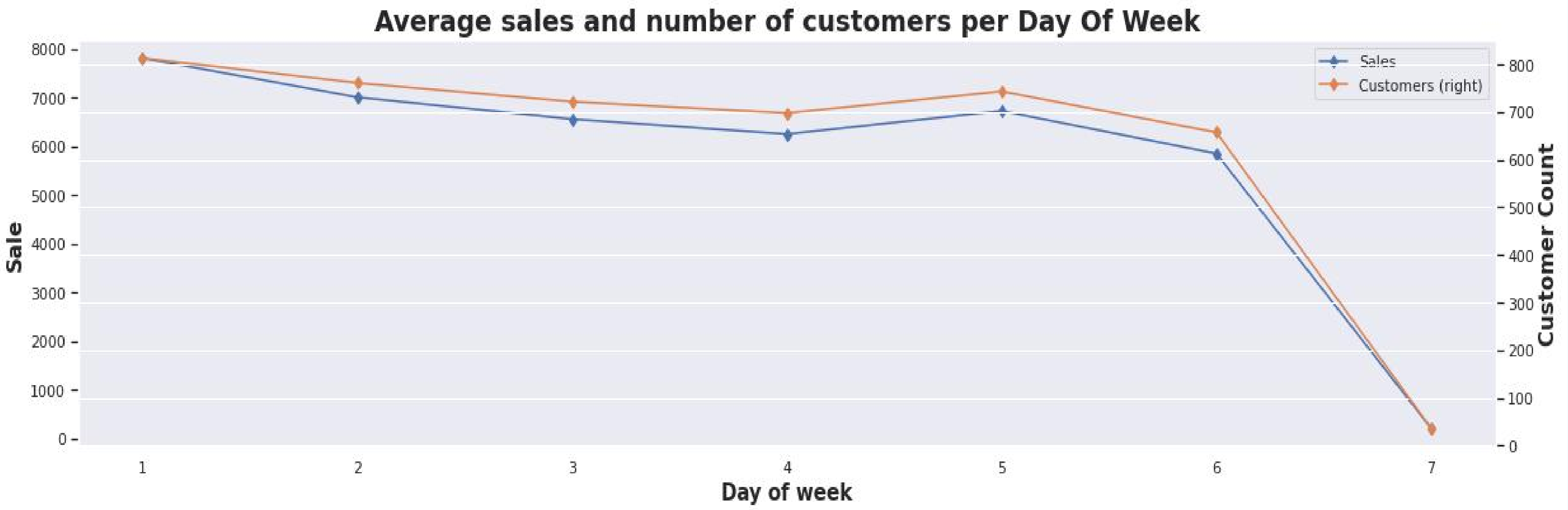
	Assortment	Sales	Customers
0	a	2945750070	332766938
1	b	70946312	16972525
2	c	2856484241	294302292

a:- means basic things.

b:- means extra things.

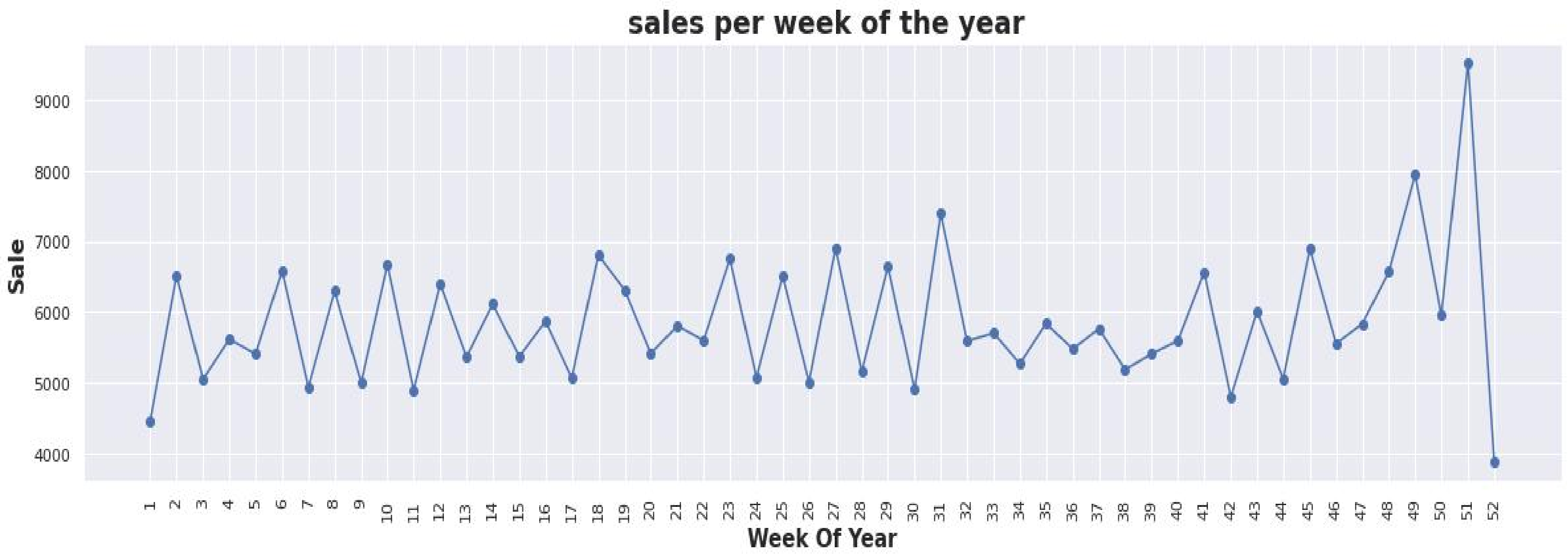
c:- means extended things so the highest variety of products.

Visualization



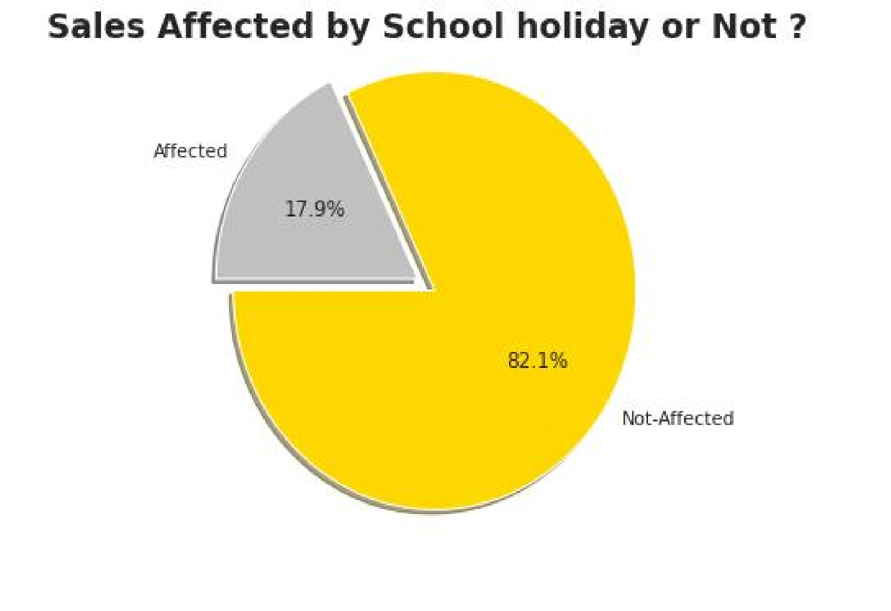
- **most stores are closed on Sundays**
- **both Sales and Customers are very low on Sundays**
- **Sales on Monday are the highest of the whole week**

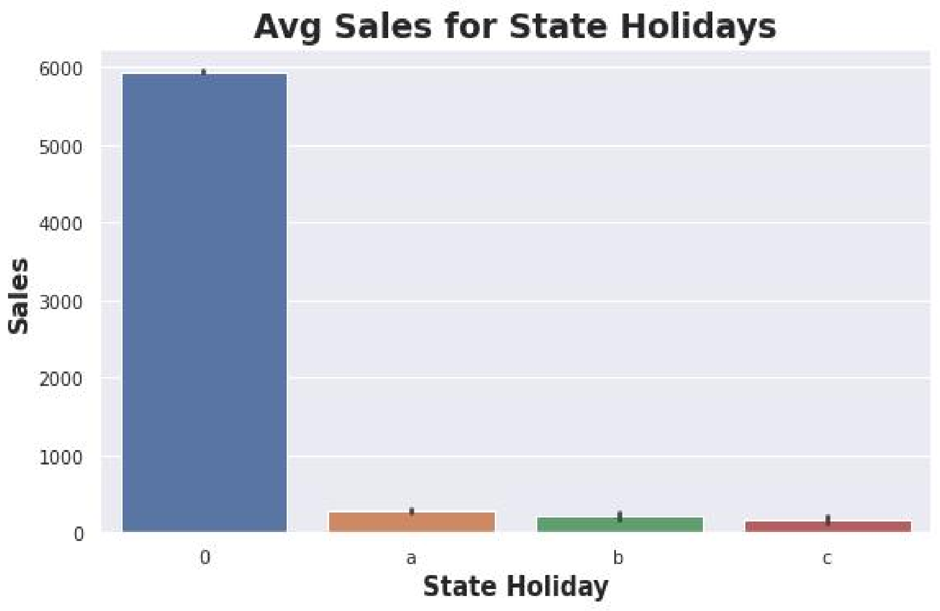
Visualization



- Christmas and New Year(see graph at weeks near 51) lead to increase in sales

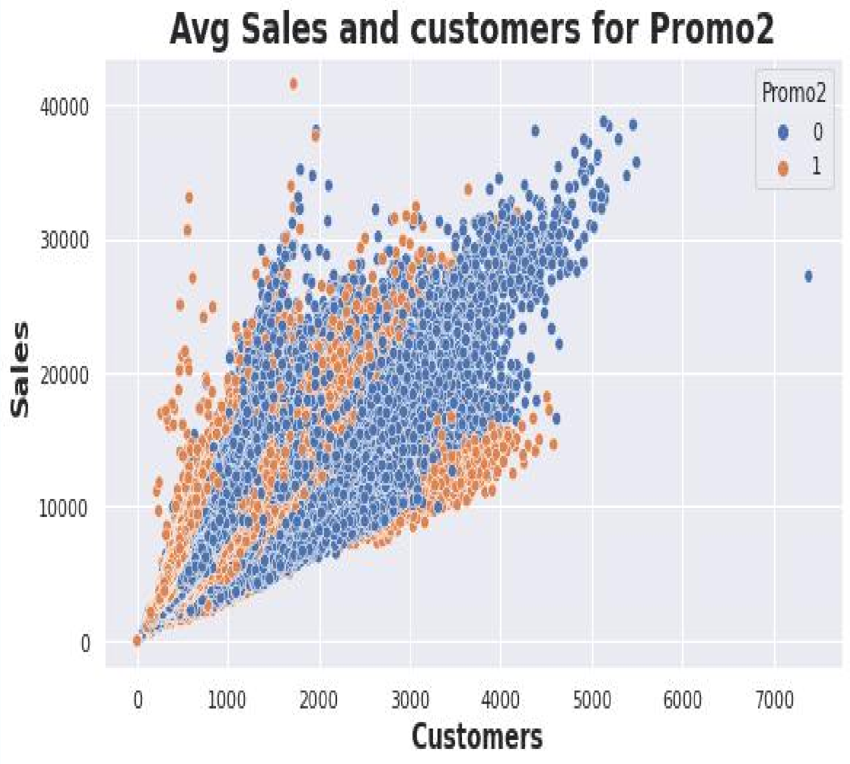
Holiday Sales

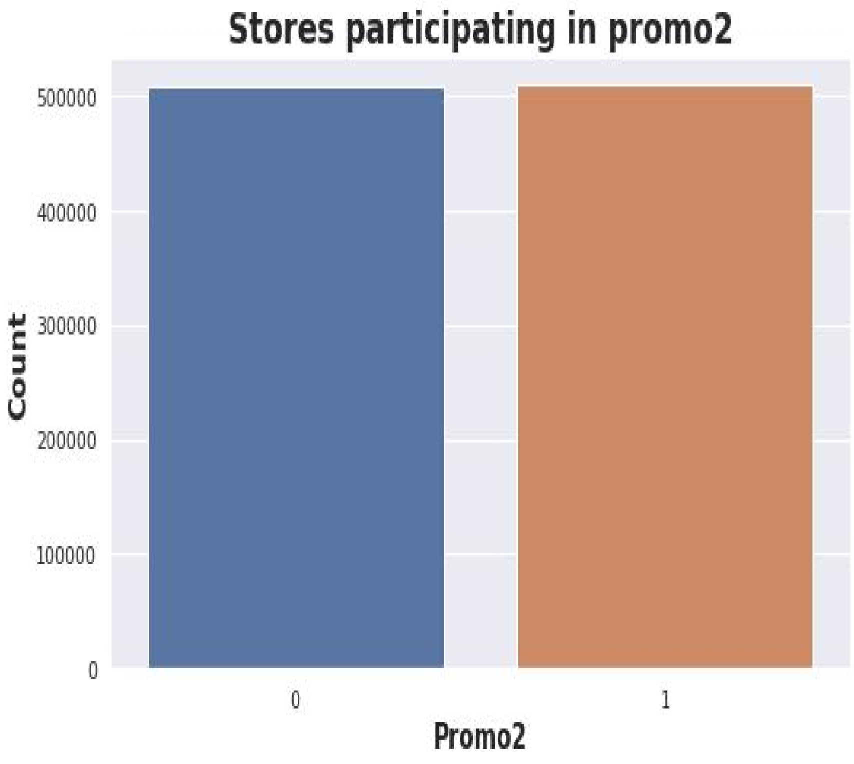




a = public holiday,b = Easter holiday
c = Christmas,0 = No Holiday

Effect of Promo2 on Sale

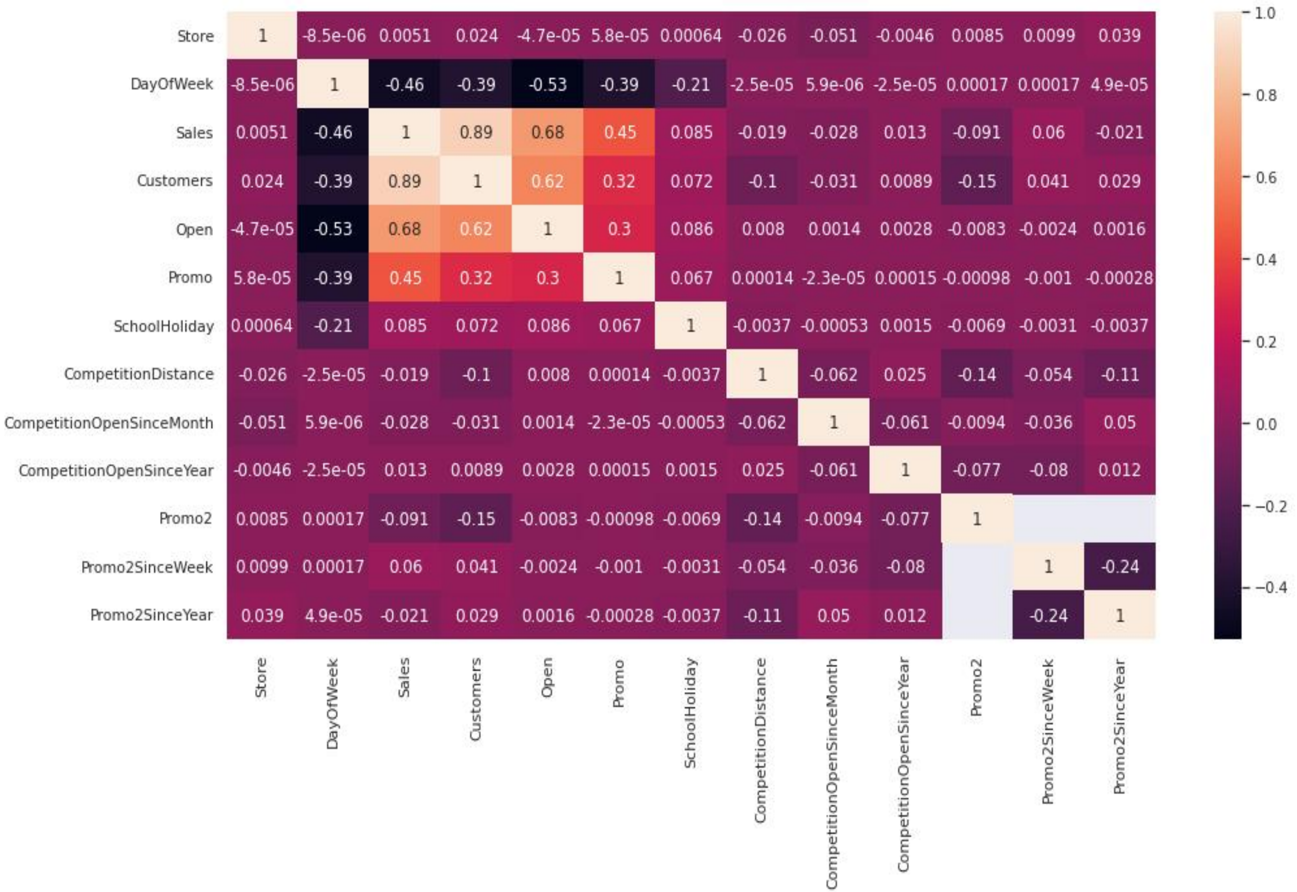




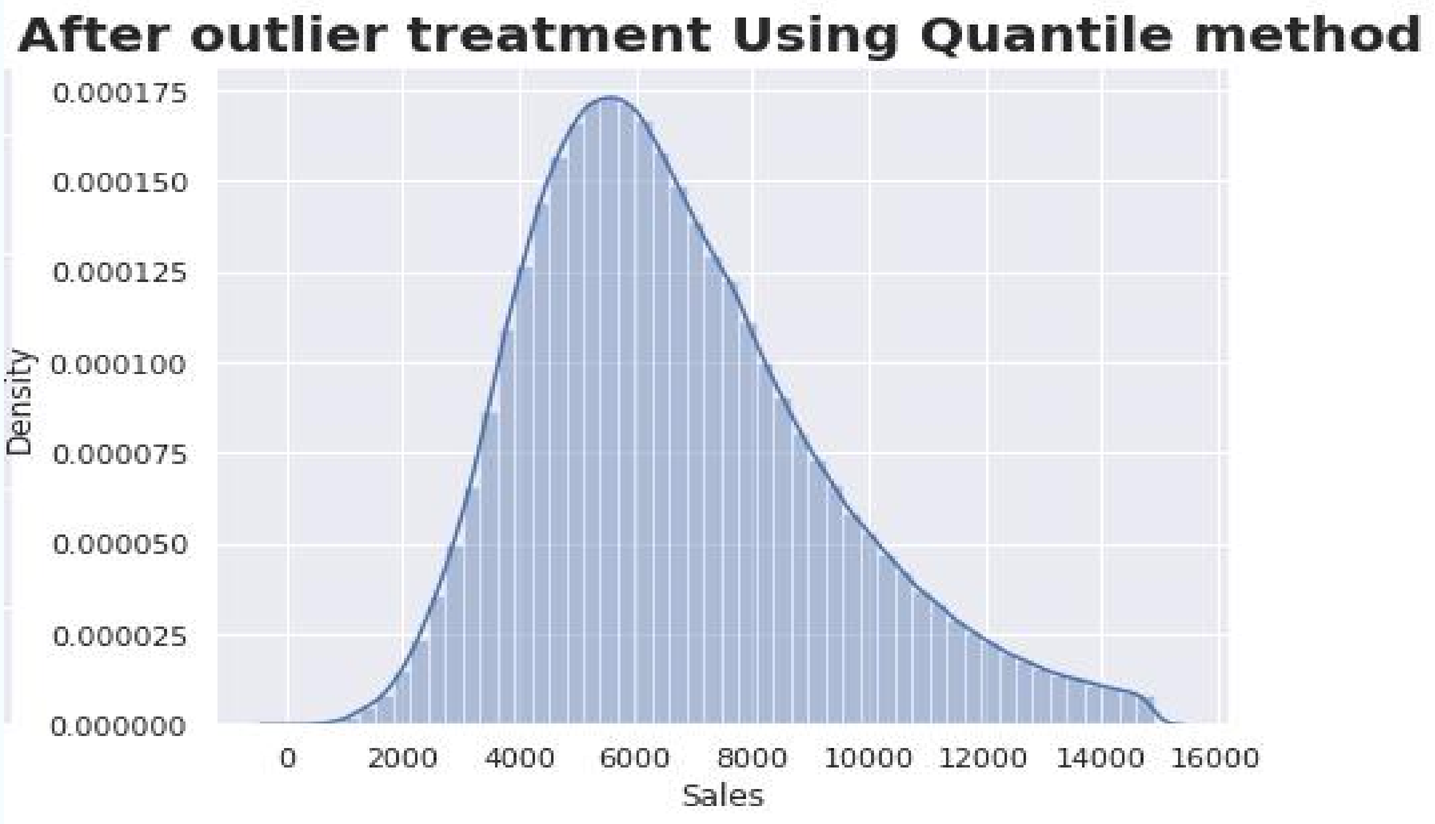
0 = store is not participating, 1 = store is participating

Correlation

- **Customers are highly positively correlated to sale.**
- **Promo is also postively correlated to sale.**
- **Day_of_week is negatively correlated to the sales.**



Removing Outliers



Data Preparation

- **Standaredscalar**: removes the mean and scales each feature/variable to unit variance

Standardization aims to reduce all the features to a common scale without distorting the range of values.

After applying standaredscalar on our dataset

```
X_train[0:15]

array([[ 0.6689788 ,  0.17478017,  1.12909868, -0.48875344, -0.42570034,
        -1.00945184,  0.37882476, -0.2134301 , -0.03072515, -0.11212159],
       [ 1.2484194 ,  0.29848084,  1.12909868, -0.48875344,  0.26967716,
        -1.00945184, -0.66762956, -1.71407817, -0.03072515, -0.11212159],
       [-0.6324386 , -0.19632185,  1.12909868, -0.48875344, -0.35766568,
```

- **Independent features :-**

'Store', 'Customers', 'Promo', 'SchoolHoliday', 'CompetitionDistance', 'Promo2', 'Year', 'Month', 'Weekofyear', 'Day', 'is_holiday_state', 'is_sale_in_week', 'is_Assortment_a', 'is_Assortment_b', 'is_Assortment_c', 'is_StoreType_a', 'is_StoreType_b', 'is_StoreType_c', 'is_StoreType_d', 'is_CompetitionOpenSince_a','is_CompetitionOpenSince_b', 'is_CompetitionOpenSince_c', 'is_open_1'

- **Dependent feature :-** 'Sales'
- Split data into 70% for train data and 30% for test data.

Baseline Models

- We performed three linear models and analyzed its single performance.

1) Linear Regression
2) Ridge Regression
3) Lasso Regression

- For the feature selection for these models we use VIF.

- A variance inflation factor (VIF) provides a measure of multicollinearity among the independent variables in a multiple regression model.

	Feature	VIF
0	Store	3.408914
1	Customers	4.660675
2	Promo	1.847322
3	SchoolHoliday	1.246836
4	CompetitionDistance	1.204054
5	Promo2	1.853563
6	Weekofyear	3.307568
7	Day	3.537397
8	is_holiday_state	1.002274
9	is_CompetitionOpenSince_c	1.015946

	Model_Name	Train R2	Test R2	Train RMSE	Test RMSE
0	LinearRegression	0.673637	0.675229	1457.694051	1455.601393
1	Lasso	0.673636	0.675228	1457.697421	1455.602286
2	Ridge	0.673637	0.675229	1457.694051	1455.601426

Baseline Models

AI

- We performed four tree regressor models and analyzed its single performance.

1. GradientBoostingRegressor
2. XGBRegressor
3. LGBMRegressor
4. KNeighborsRegressor

	Model_Name	Train R2	Test R2	Train RMSE	Test RMSE
0	GradientBoostingRegressor	0.855545	0.854347	969.800303	974.794539
1	XGBRegressor	0.855539	0.854342	969.819671	974.812956
2	LGBMRegressor	0.914868	0.913629	744.497855	750.648494
3	KNeighborsRegressor	0.909788	0.856347	766.388954	968.078290

- The tree regressor perform vey well as compared to linear models.
- Out thse four models LGBMRegressor and KNeighborsRegressor having less RMSE and high R2SCORE.

Hyperparameter Tuning

AI

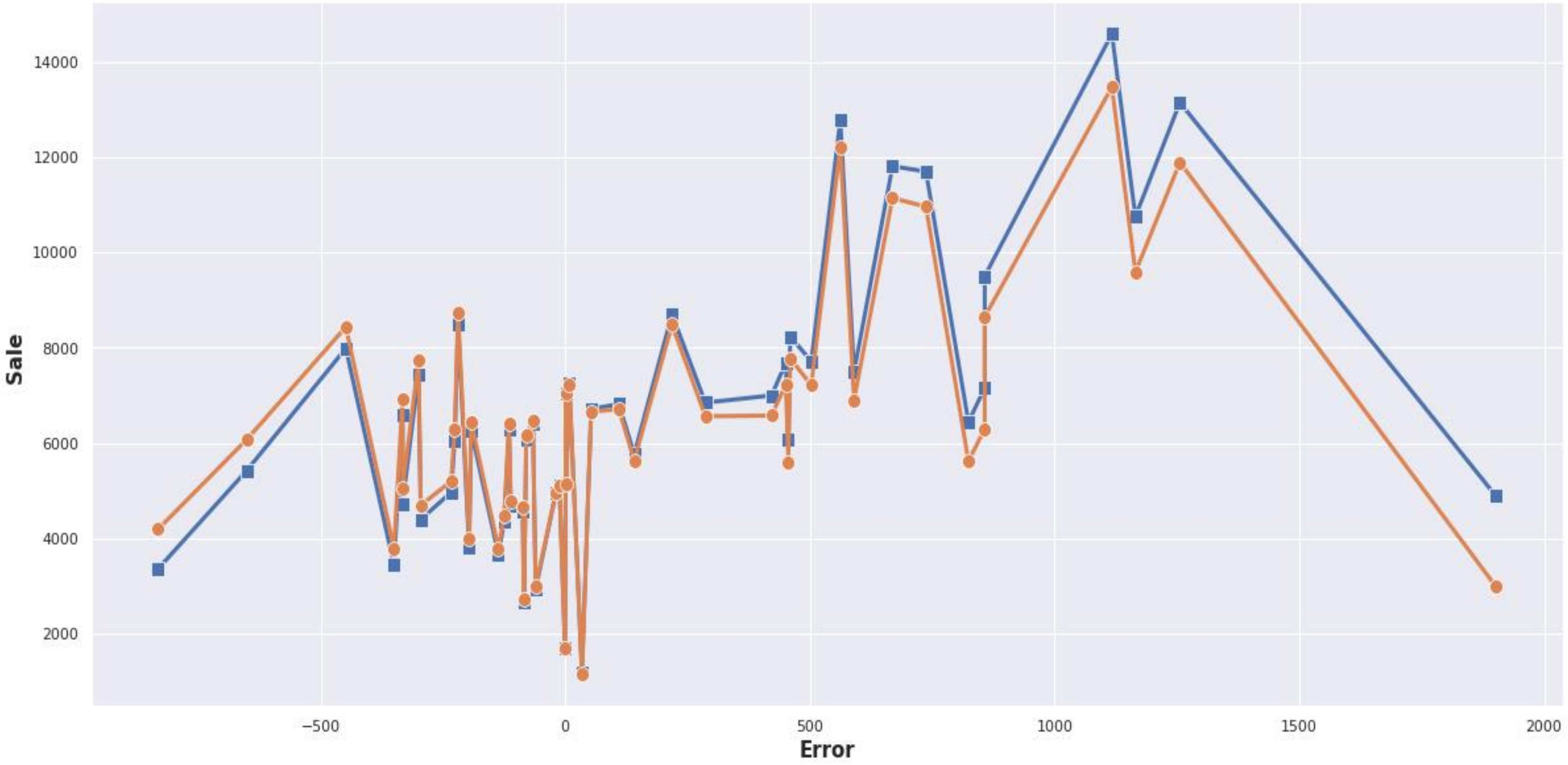
- Used RandomSearchCV to do hyperparameter tuning
- Hyperparameters I have used for LGBMR AND XGBR :-

	Model_Name	Train R2	Test R2	Train RMSE	Test RMSE	Train MAE	Test MAE
0	LGBMRegressor_with_hyper	0.886275	0.885135	860.485528	865.658822	651.472778	653.539364

	Model_Name	Train R2	Test R2	Train RMSE	Test RMSE	Train MAE	Test MAE
1	XGBRegressor_with_hyper	0.964254	0.961817	482.425459	499.103135	355.639222	365.554885

- But, XGBR is the best model with their performance with less RMSE as compared to LGBM. XGBR is the best and final model.
- We observed that the LGBM increase the RMSE after hyper tuning.
- We can improve it using a specific parameter which is the “n_estimators”. However, it takes a lot of time to compute.

Prediction



- **prediction made by our model quite good i.e it having low bias and low varianc.**

Conclusion

AI

- We selected three types of linear regression models, namely Linear Regression , Ridge Regression , and Lasso Regression. These models have R2SCORE around 0.67 for both train and test. They perform well, but not as well as other models.

- All three linear regression models are perfectly fitted ,as they have low bias and low vaiance.

- Based on tree regressor ,we developed baseline models for further evaluation , namely Gradient Boosting Regressor, XGB Regressor , KNearestNeighbors Regressor,LightGBM Regressor.

- According to RMSE and R2SCORE xgbr,knn and lgbm the most reliable models.The KNearestNeighbors Regressor had a very high R2SCORE, however, it required much computational time, so we dropped it.

- For the hyper parameter tuning ,we choose XGBR and LGBM based on the RMSE and R2SCORE obtained from the baseline models.

- XGB has perform very good on performing hyperparameter tuning which has R2SCORE 0.96 (train data) & 0.96 (test data) and RMSE. with no underfitting and overfitting.The predictions is pretty close to the real value for sales.

Conclusion

❑ Important Visualization

- It is also observed that the out all the features "Customers" is most important. this implies that "customers" are directly impacting the sales.

- We can see that stores of type "A" has higher amount of total Customers and Sales.

- The store which is run promo leads to increase in sales and customers as compared to store which are not runing the promo.

- There were more stores open during the school holidays than during state holidays. Another important point to note is that the stores which were open during school holidays had more sales than usual.

- Sales are increased during Chirstmas week.

Thank you!