Rossmann Retails Sales Prediction

Regression Project by
Jayesh Dahiwale

1. Table of Content

Introduction

- **Dirk Rossmann GmbH**, commonly referred to as **Rossmann**, is one of the largest drug store chains in Europe with around 56,200 employees and more than 4000 stores.
- The company was founded in 1972 by **Dirk Rossmann** with its headquarters in **Burgwedel** near **Hanover** in Germany. The Rossmann family owns 60% of the company.
- The product range includes up to **21,700** items and can vary depending on the size of the shop and the location. In addition to drugstore goods with a focus on skin, hair, body, baby and health, Rossmann also offers promotional items ("World of Ideas"), pet food, a photo service and a wide range of natural foods and wines.

AI

- **Here we are going to do an exploratory data analysis on the dataset as well as try to run the regression model which can predict the sales if given some inputs.**

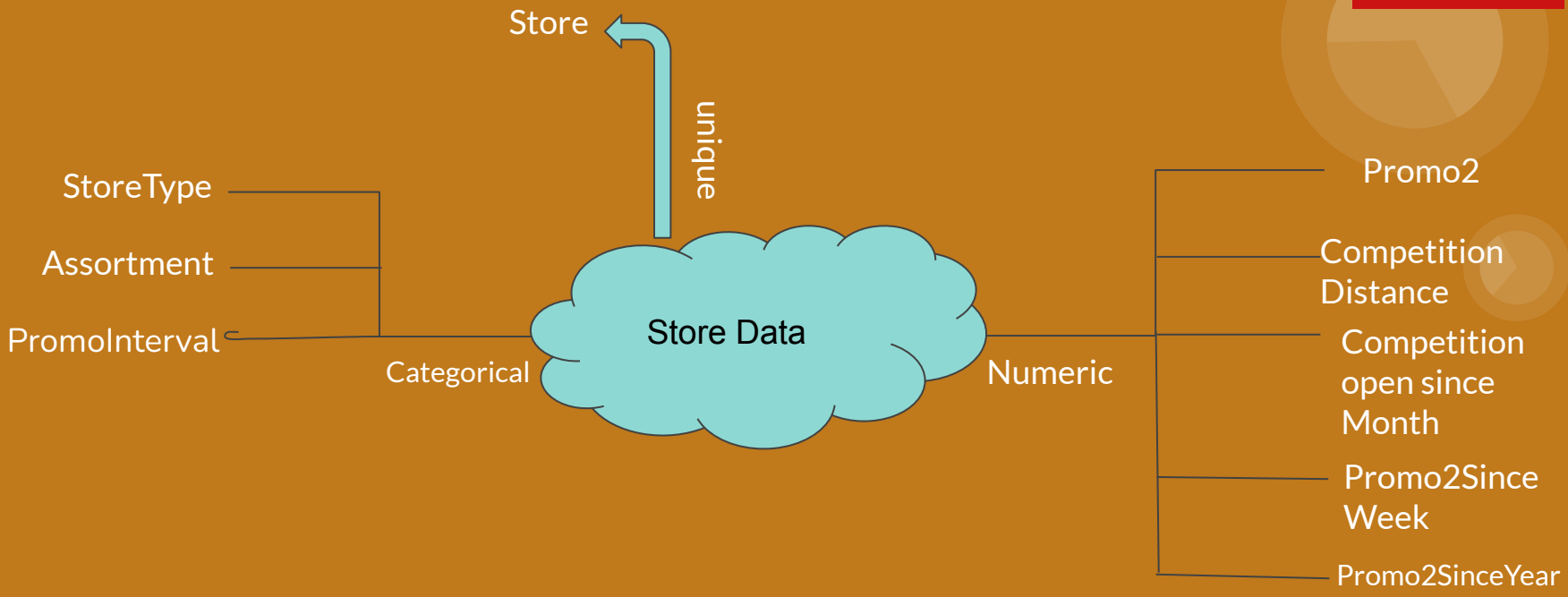
- Provided there are **two datasets,** one is **Store dataset** having **1,115** observations in it with **10** columns and It gives us static information about each store such as the model and **assortment** of the store, information about the nearest **competitor store,** and whether or not they participate in the consecutive promotion **"Promo2"**. Largely we're looking at numerical and date data, but **Store Type** and **Assortment** are flagged with letters to indicate store models and assortment level, per the variable explanations, as well as the PromoInterval column listing abbreviated months.

- Other dataset is about **Sales** dataset having **1,017,209** observations in it

  with **9 columns** and It gives us static information about each store such as the model and assortment of the store, information about the nearest competitor store, and whether or not they participate in the consecutive promotion **"Promo2".** Largely we're looking at numerical and date data, but **Store Type** and **Assortment** are flagged with letters to indicate store models and assortment level, per the variable explanations, as well as the PromoInterval column listing abbreviated months.

- The business objective is to increase the number of sales by predicting the rates at at optimal rate and finding the best suitable condition which attract the customers thereby increasing the profit for the Drug Store.

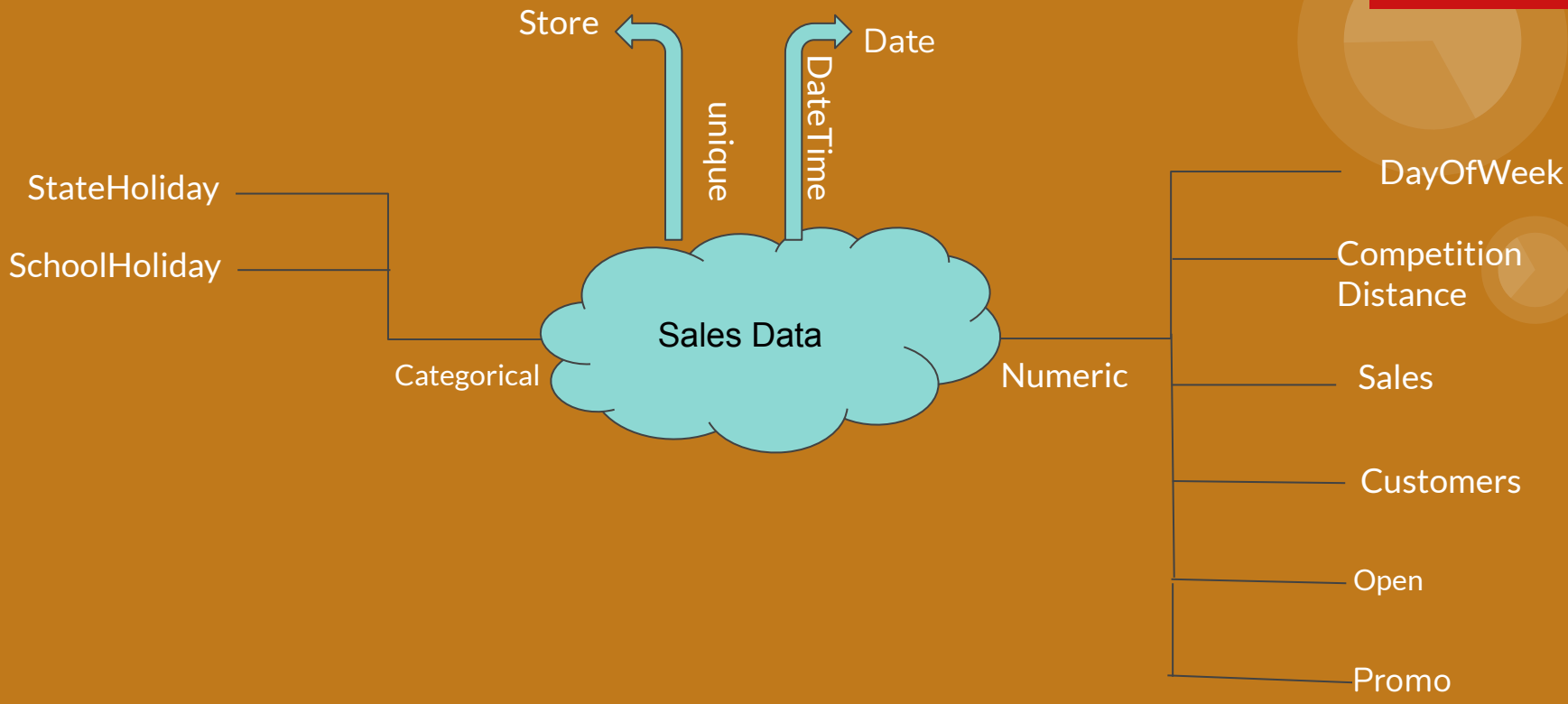
Data Summary For Store Dataset



Variables in Brief

- **Store** :
  - It is an unique no given to the each store starting from 1.
- **StoreType** :
  - It represent the type of store it is. There are **4** types of stores in dataset.
- **Assortment** :
  - Describes an assortment level: a = basic, b = extra, c = extended
- **PromoInterval**:
  - Describes the consecutive intervals Promo2 is started, naming the months the promotion is started anew. E.g. "Feb,May,Aug,Nov" means each round starts in February, May, August, November of any given year for that store
- **Promo2**:
  - Promo2 is a continuing and consecutive promotion for some stores: 0 = store is not participating, 1 = store is participating
- **Competition Distance**:
  - **(continuous)** distance in meters to the nearest competitor store
- **Competition Open Since [Month/ Year]**:
  - **(discrete)** gives the approximate year and month of the time the nearest competitor store was opened
- **Promo2Since[Year/ week]**:
  - **(discrete)** describes the year and calendar week when the store started participating in Promo2

Data Summary for Sales Datatset



Variables in Brief

- **Store** :
  - It is an unique no given to the each store starting from 1.
- **Date** :
  - **Date** in YYYY-MM-DD
- **Day of Week** :
  - **(ordinal)** Day of the week, using 1-7 for Monday - Sunday respectively
- **CompetitionDistance**:
  - **(continuous)** distance in meters to the nearest competitor store
- **Sales**:
  - **(discrete)** the number of transactions recorded at the store that day
- **Customers**:
  - **(discrete)** the number of customers on a given day
- **Open**:
  - **(nominal)** an indicator for whether the store was open: 0 = closed, 1 = open
- **Promo**:
  - **(nominal)** indicates whether a store is running a promo on that day
- **StateHoliday**:
  - **(nominal)** indicates a state holiday. Normally all stores, with few exceptions, are closed on state holidays. Note that all schools are closed on public holidays and weekends. a = public holiday, b = Easter holiday, c = Christmas, 0 = None

Graphs used for EDA :

- Count Plot
- Bar Plot
- Scatter Plot
- Box Plot
- HeatMap

Python Libraries used for EDA :

- Matplotlib
- Numpy
- Pandas
- Seaborn
- ScikitLearn
- StatsModels

Question we are trying to answer :

1) Visualising the distribution of "Sales" & "Customers"?

2) Statistics of Sales column ?

3) Which rows are unnecessary and need to be removed ??

4) What are the outliers ?

5) Establishing relationship between Sales and Customers ?

6) How stores are performing in Sales by month based on Assortment type ?

7) How UPT metric compares across stores of different assortment types ?

8)Correlation between competition distance and UPT metric?

9)Which linear regression model is best?

StoreLookup Dataset :

- Lets check the null values in the dataset:

```
RangeIndex: 1115 entries, 0 to 1114
Data columns (total 10 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Store                      1115 non-null   int64
 1   StoreType                  1115 non-null   object
 2   Assortment                 1115 non-null   object
 3   CompetitionDistance        1112 non-null   float64
 4   CompetitionOpenSinceMonth  761 non-null    float64
 5   CompetitionOpenSinceYear   761 non-null    float64
 6   Promo2                     1115 non-null   int64
 7   Promo2SinceWeek            571 non-null    float64
 8   Promo2SinceYear            571 non-null    float64
 9   PromoInterval              571 non-null    object
dtypes: float64(5), int64(2), object(3)
```

	ColumnName	Null_count
0	CompetitionDistance	3
1	CompetitionOpenSinceMonth	354
2	CompetitionOpenSinceYear	354
3	Promo2SinceWeek	544
4	Promo2SinceYear	544
5	PromoInterval	544

- We can see that CompetitionOpenSinceMonth/Year has 354 null values
- Promo2SinceWeek/Year,Promointerval has 544 null value each.

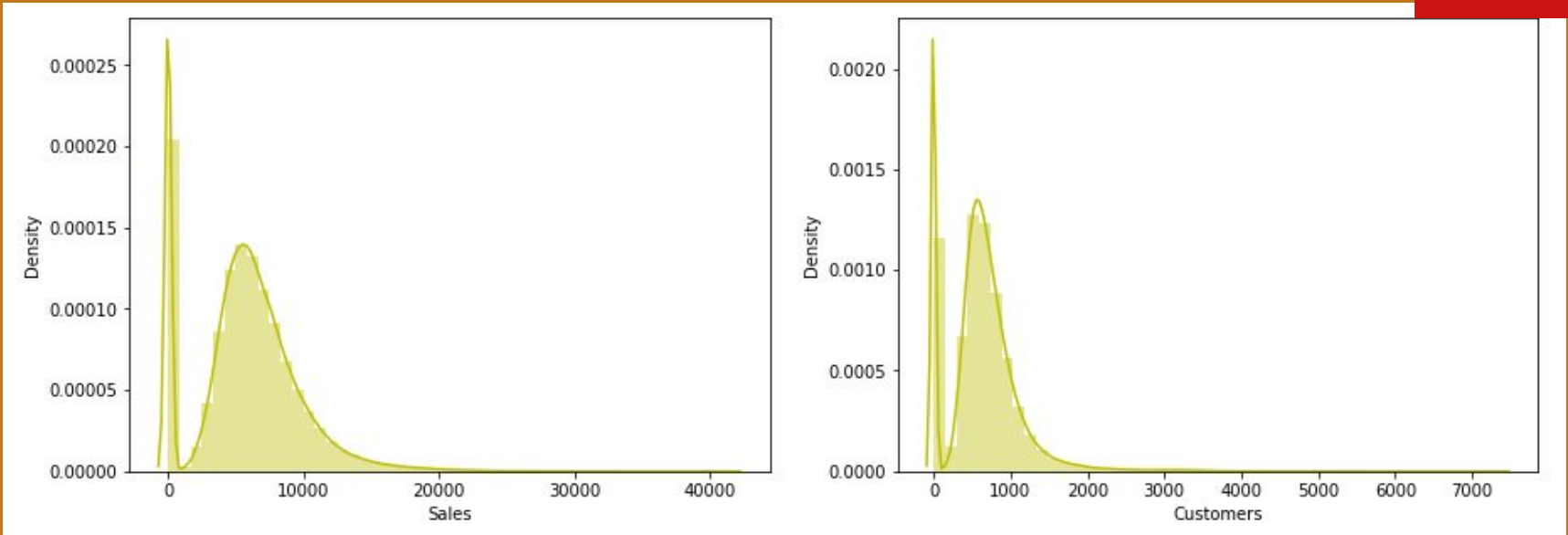
Sales Dataset :

- Lets check the null values in the dataset:

```
RangeIndex: 1017209 entries, 0 to 1017208
Data columns (total 9 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
```

- We can see that there is no null values in the dataset

1. Visualizing the distribution of “Sales” and “Customers” co



- Histograms of our Sales and Customers values show us a positive skew and high kurtosis.
- Customer and Scales column has positive skew of 1.59 and 0.64 respectively

- Kurtosis score for sales column is 1.778
- Kurtosis score for Customers column in 7.09

2. Statistics of Sales Column

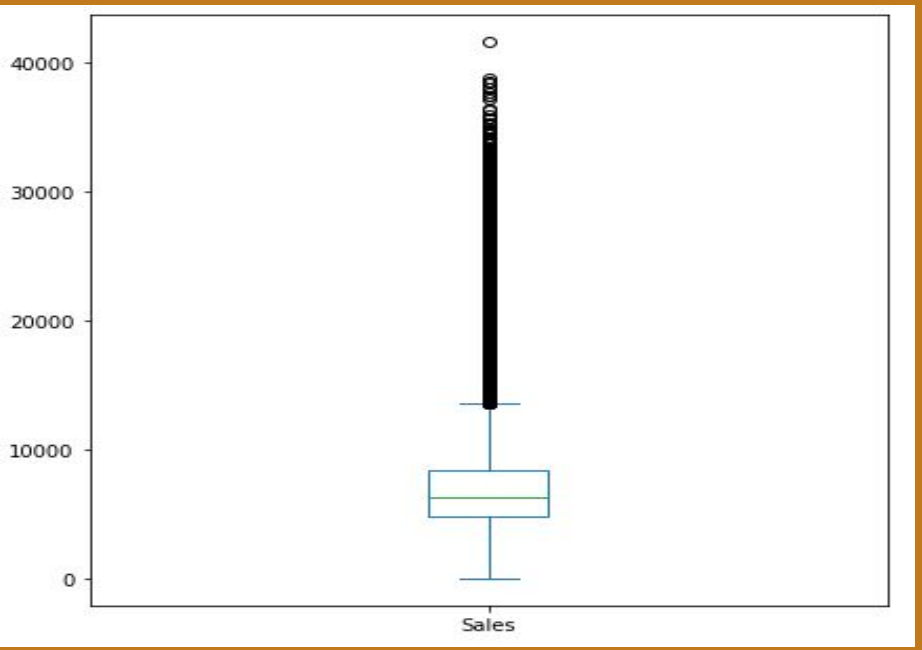
	Sales
count	1.017209e+06
mean	5.773819e+03
std	3.849926e+03
min	0.000000e+00
25%	3.727000e+03
50%	5.744000e+03
75%	7.856000e+03
max	4.155100e+04

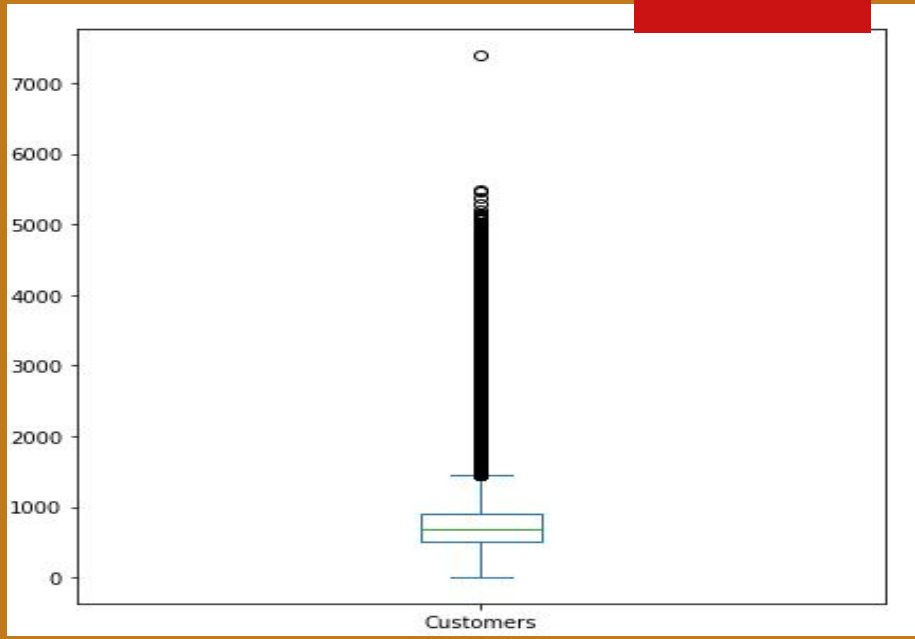
- We can see our sales figures have a slightly positive skew, with the **mean (5773.819)** only slightly larger than the **median (5744.000)**, suggesting most outliers are to the right of the mean.
- High kurtosis indicates it's **leptokurtic** with the likelihood of heavy tails and outliers that may be extreme. Considering our min and max values of **0** and **41,551** sales, we aren't surprised to see there may be some extreme outliers.
- The max value well above the mean of 5,773.819 and outside the **standard deviation of 3849.926** helps us see how our mean ends up getting pulled slightly to the right for our positive skew.
- There is no mode as we don't have any stores recording the exact same number of sales on any days, which isn't surprising.

3.Which rows are unnecessary and need to be removed

- I look at entries for stores on days they were closed.
- For the purposes of our analysis I've chosen to drop these rows, as no sales are recorded on days stores are closed. The zero sales recorded for each of these rows lowers the average sales, and we can see this by comparing the mean Sales for all entries in our table to the mean Sales of only days that stores were open. If we filter for entries of stores that are closed we'll see a return of **172,817** rows, all of which record the expected 0 sales, lowering our mean Sales statistic.
- **The potential information lost here is if we want to compare stores based on the number of days they are open or closed**, but that is beyond the scope of our analysis for now. To avoid losing this information we will make a copy of our dataframe with only the days stores are open, to further be referred to as sales, rather than altering merged_sales in case we wish to access this data at a later time.
- mean sales including entries for days stores are closed - $ **5773.81**
- Mean sales excluding entries for days when the store was closed - $ **6955.51**
- We can see that there is a whooping difference in mean sales value after deleting the rows having store status is “**Closed**”

4. What are outliers in the Dataset





- From the box plots above we can see that **Sales**, **Customers** appear to have significant outliers, so we'll explore further by calculating and investigating the outliers for each one.

Sales Outliers

- Interquartile Range for Sales column : 3501
- Lower Band : -392.5
- Upper Band : 13611.5
- Percentage of Sales that are outliers: 3.64%
- We know from our summary statistics that there aren't any sales below 0, so we'll just look at the upper outliers that we've calculated for the Sales column.

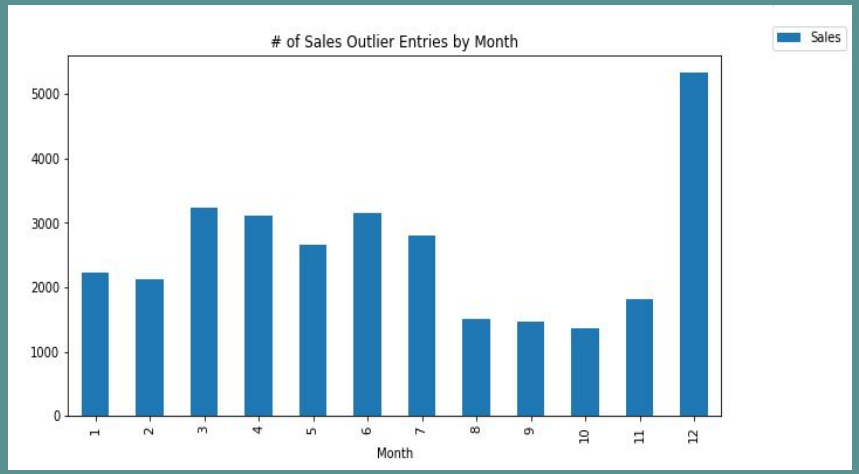
Upper Outliers

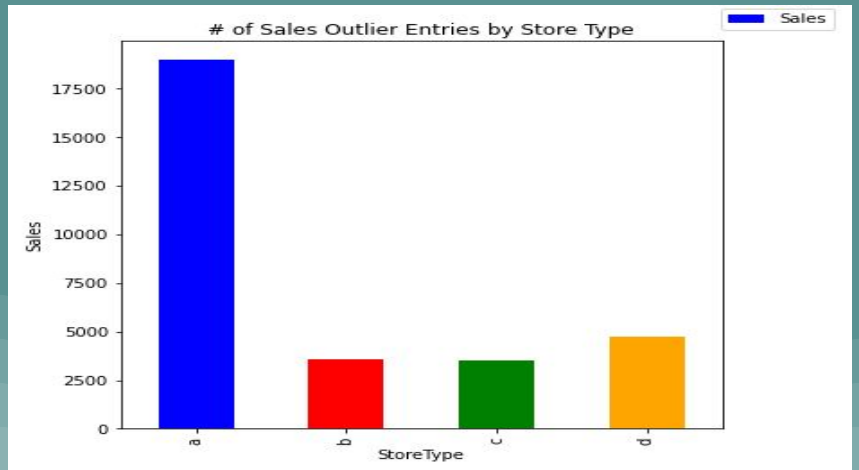
	Store	DayOfWeek	Date	Sales	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance
3	4	5	2015-07-31	13995	1498	1	0	1	c	c	620.0
6	7	5	2015-07-31	15344	1414	1	0	1	a	c	24000.0
23	24	5	2015-07-31	14190	1082	1	0	1	a	c	4590.0
24	25	5	2015-07-31	14180	1586	1	0	1	c	a	430.0
83	84	5	2015-07-31	14949	1439	1	0	1	a	c	11810.0
...	...	...	...	...	...	...	...	...	...	...	...
1015796	817	3	2013-01-02	25357	3462	0	0	1	a	a	140.0
1015821	842	3	2013-01-02	20355	1257	0	0	1	d	c	1200.0
1016012	1033	3	2013-01-02	13811	1408	0	0	1	a	a	7680.0
1016093	1114	3	2013-01-02	20642	3401	0	0	1	a	c	870.0
1016356	262	2	2013-01-01	17267	2875	0	a	1	b	a	1180.0

30769 rows × 20 columns

- We can see that there are **30,769** sales come under upper outliers which account for only **3.64%** of the total sales.

Sales Outliers trend based on Month or Type of Store





- When we look at the Sales outliers by month, we see the most represented month is **December at 17.33%**, which is unsurprising given the Christmas holidays. However, when we look at the outliers by Store Type we see that the **61.71% majority** are coming from **Type A stores**, while **Type B, C, D** are more equally represented at **11-15%**. This suggests that Type A stores may be the best performers in regards to outstanding sales days, and is worth looking into further.
-
- Below we will treat our Sales outliers by imputing them with our upper range value we calculated earlier, **13611.5**, rounded up to **13612** as our Sales column is a measure of discrete values using whole numbers. As these outliers represent exceptionally high sales day, they are intended to be high numbers, but we would like to treat the outliers to limit their influence on any future modelling. As such imputing with our upper range value feels more appropriate than using our mean Sales value.

Customer Outliers

- Interquartile Range for Sales column : 374.0
- Lower Band : -42.0
- Upper Band : 1454.0
- Percentage of Sales that are outliers: 4.84%
- We know from our summary statistics that there aren't any customers below 0, so we'll just look at the upper outliers that we've calculated for the Customers column.

Upper Outliers

	Store	DayOfWeek	Date	Sales	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance
3	4	5	2015-07-31	13612	1498	1	0	1	c	c	620.0
24	25	5	2015-07-31	13612	1586	1	0	1	c	a	430.0
124	125	5	2015-07-31	13612	2041	1	0	1	a	a	760.0
210	211	5	2015-07-31	13612	1659	1	0	1	a	c	350.0
250	251	5	2015-07-31	13612	2508	1	0	1	a	c	340.0
...	...	...	...	...	...	...	...	...	...	...	...
1016093	1114	3	2013-01-02	13612	3401	0	0	1	a	c	870.0
1016356	262	2	2013-01-01	13612	2875	0	a	1	b	a	1180.0
1016517	423	2	2013-01-01	9643	1751	0	a	1	b	a	1270.0
1016656	562	2	2013-01-01	8498	1675	0	a	1	b	c	1210.0
1016827	733	2	2013-01-01	10765	2377	0	a	1	b	b	860.0

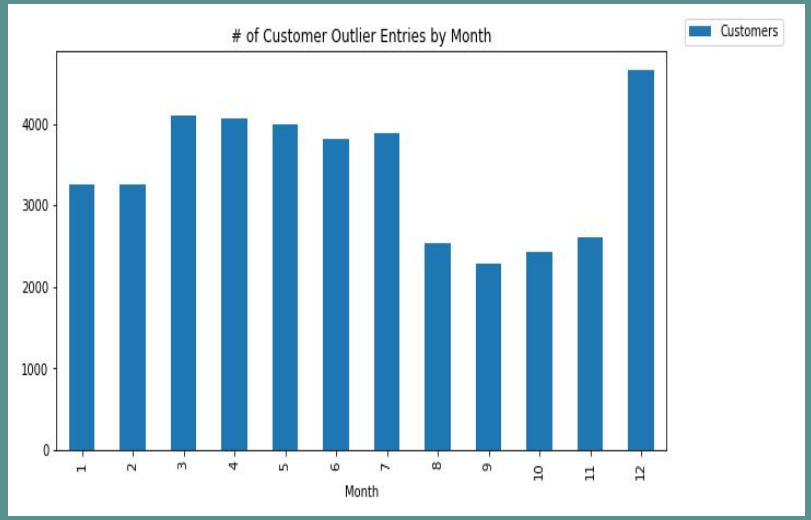
40853 rows × 20 columns

- We Can see right away that several of these entries have a **Sales value of 13,612**, which we know to be our newly imputed upper range value for Sales outliers. We expect a high correlation between Customers driving Sales, so we'll check to see how much crossover we have between our Sales and Customers outliers

	Store	DayOfWeek	Date	Sales	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance
3	4	5	2015-07-31	13612	1498	1	0	1	c	c	620.0
24	25	5	2015-07-31	13612	1586	1	0	1	c	a	430.0
124	125	5	2015-07-31	13612	2041	1	0	1	a	a	760.0
210	211	5	2015-07-31	13612	1659	1	0	1	a	c	350.0
250	251	5	2015-07-31	13612	2508	1	0	1	a	c	340.0
...	...	...	...	...	...	...	...	...	...	...	...
1015735	756	3	2013-01-02	13612	2465	0	0	1	a	c	50.0
1015767	788	3	2013-01-02	13612	1791	0	0	1	a	c	1530.0
1015796	817	3	2013-01-02	13612	3462	0	0	1	a	a	140.0
1016093	1114	3	2013-01-02	13612	3401	0	0	1	a	c	870.0
1016356	262	2	2013-01-01	13612	2875	0	a	1	b	a	1180.0

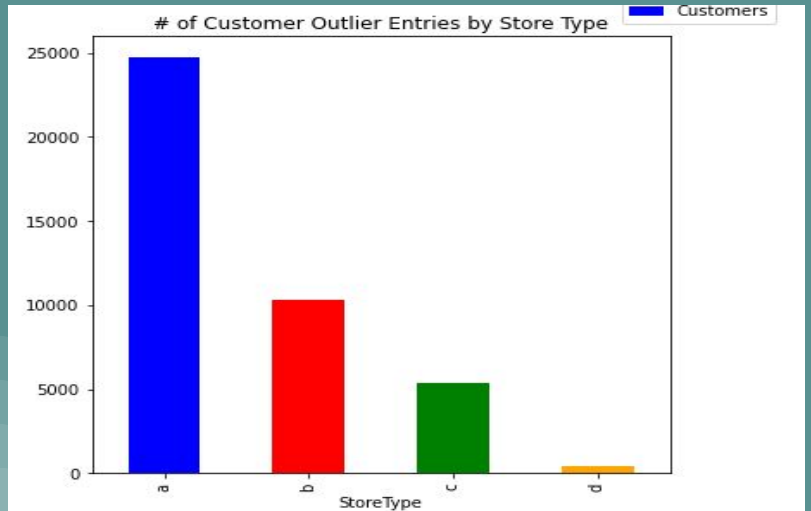
21420 rows × 20 columns

- We can see a crossover of **21,420 rows**, or approximately **52%** of our Customer outlier entries are also Sales outlier entries.
- We will also investigate how these Customer outliers break down by Month and StoreType just as we did with our Sales outliers.



Customer Outliers Entries by Month:

- **December** is our most represented month for Customer outliers, similar to our Sales outliers, but it's percentage of outliers is less than we saw with our Sales.



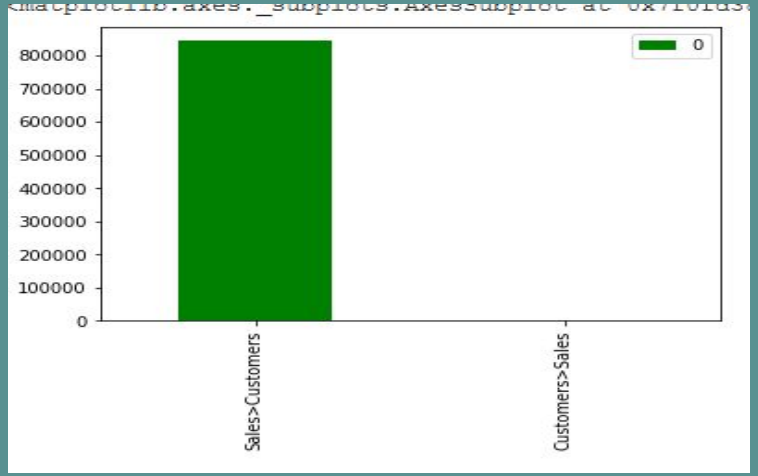
Customer Outliers Entries by StoreType:

- We also see **store Type A** with the strongest showing when we break down the outliers by store type. Much like the Sales outliers **Type A stores** represent a **strong 60%+** of the outliers. Surprisingly, **Type D stores** represent a tiny **1.02%** of these Customer outliers, where as they represented the second largest percentage of Sales outliers at **15.29%**. Further investigation into the number of items bought (Sales) per transaction (Customer) may prove insightful.

5.Establishing relationship between Sales and Customer

- We can see that for almost all entries we have in our dataframe, the number of Sales at a given store is greater than the number of Customers.

	0
Sales>Customers	844338
Customers>Sales	54



- Suggesting that the Customers value is derived by how many transactions there are at a store, and the Sales value is indicative of how many individual items are sold. Thus we can calculate the average number of items sold for each transaction as **Units Per Transaction** (UPT).

- First we'll quickly investigate the **54 rows** where there aren't more Sales than Customers.

	Store	DayOfWeek	Date	Sales	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance
86825	971	5	2015-05-15	0	0	0	0	1	c	a	1140.0
142278	674	4	2015-03-26	0	0	0	0	0	a	a	2640.0
196938	699	4	2015-02-05	0	0	1	0	0	a	a	180.0
322053	708	3	2014-10-01	0	0	1	0	0	c	c	11470.0
330176	357	1	2014-09-22	0	0	0	0	0	a	a	2060.0
340348	227	4	2014-09-11	0	0	0	0	0	a	a	2370.0
340860	835	4	2014-09-11	0	0	0	0	0	a	a	2890.0
341795	835	3	2014-09-10	0	0	0	0	0	a	a	2890.0
346232	548	5	2014-09-05	0	0	1	0	1	d	c	3760.0
346734	28	4	2014-09-04	0	0	1	0	0	a	a	1200.0
347669	28	3	2014-09-03	0	0	1	0	1	a	a	1200.0
348604	28	2	2014-09-02	0	0	1	0	1	a	a	1200.0
386065	102	4	2014-07-24	0	0	0	0	1	a	a	150.0

- We can see the majority of these days are entries with both **zero Sales** and **zero Customers** recorded. This seems odd for a day that the store is open. A quick check of merged_sales, which still has the Open column, gives us the same results and assures us that the stores are indeed marked as open on these days.

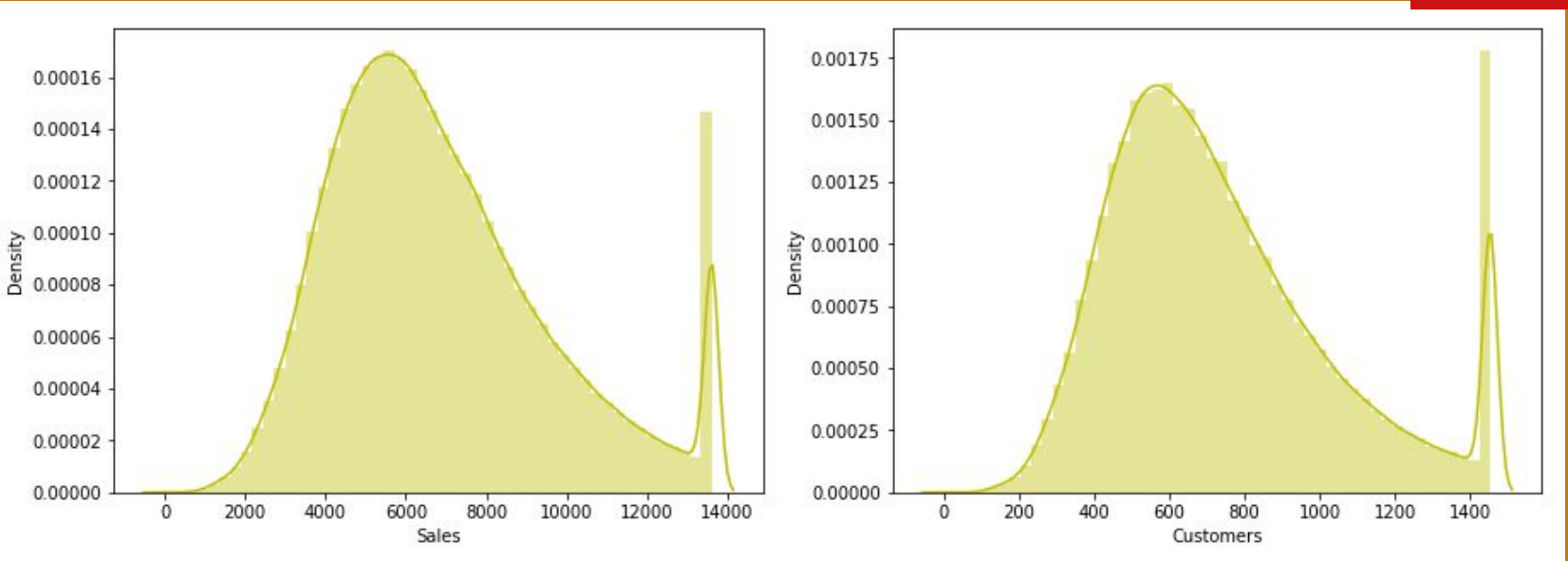
- Let's also look at the cases that aren't zero Sales and zero Customers. Where Customers > Sales numbers

	Store	DayOfWeek	Date	Sales	Customers	Promo	StateHoliday	SchoolHoliday	StoreType	Assortment	CompetitionDistance
478649	1100	2	2014-04-29	0	3	1	0	0	a	a	540.0
889932	948	4	2013-04-25	0	5	1	0	0	b	b	1430.0

- I'm unsure of why we have two days with zero Sales and a small handful of Customers. As I have no explanation for these **2 days**, nor the **52 other open days with zero Sales and zero Customers**, I don't feel comfortable deleting them.

- This does pose a small problem for calculating our average UPT, however. As such we will create our UPT column by dividing the day's Sales by the days Customers to find the average Units Per Transaction for each day and store. The resulting 52 null values will be imputed with a zero to reflect the zero Sales for those entries.

Distribution curves for sales and Customer Post cleaning the D



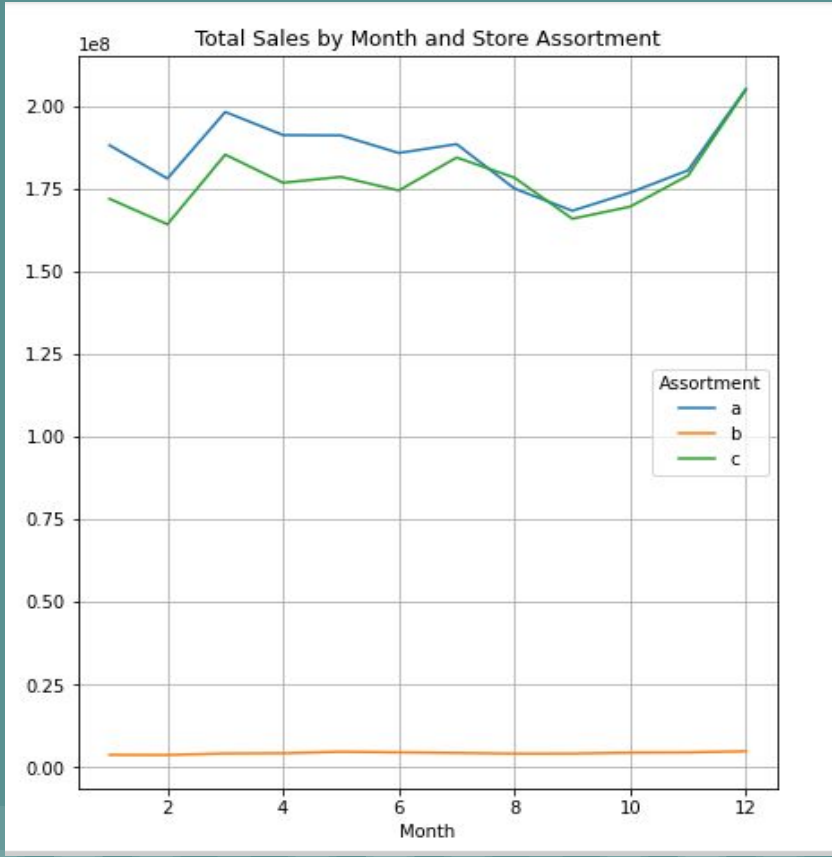
- Histograms of our Sales and Customers values shows us a slight positive skew, which changed a little after we cleaned, and we see a more significant change in kurtosis being lowered. We also see the effect of imputing our outliers with our upper range limit on the right side of either histogram.
- We now got results that are looking far closer to a standard normalization.

6.How stores perform in Sales by month based on Assortment Type

Assortment	a	b	c
Month			
1	188100262	3754636	171893846
2	178031925	3710558	164191126
3	198172555	4173361	185276735
4	191177863	4233401	176759688
5	191082283	4691504	178547553
6	185757989	4503724	174429599
7	188440724	4338877	184388970
8	175000464	4111225	178291640
9	168298236	4127263	165868274
10	173794666	4427807	169542336
11	180521975	4489868	178945019
12	205095239	4839992	204907502

- Let's explore how stores perform in Sales by month, based on Assortment type. We know that **Assortment type A** offers a "**basic**" assortment of merchandise, **Type B** offers an "**extra**" assortment, and **type C** offers an "**extended**" assortment.

- Because our data ranges from **Jan. 1, 2013 - July 31, 2015,** we will exclude the **2015** data for now so as we are only looking at a complete years' worth of numbers.



- A quick look at Sales by volume of total sales shows that stores of **Assortment types A and C** have significantly **more volume than type B stores**. Type B stores stay fairly consistent in total Sales volume across all months, with minor upticks during **mid-year** and **end year**. Type A and C stores can be seen to follow very similar trends in terms of Sales volume.

- A quick look at how many stores we have of each Assortment type will show us significantly less stores of Assortment type B, which accounts for the significantly volume of Sales.

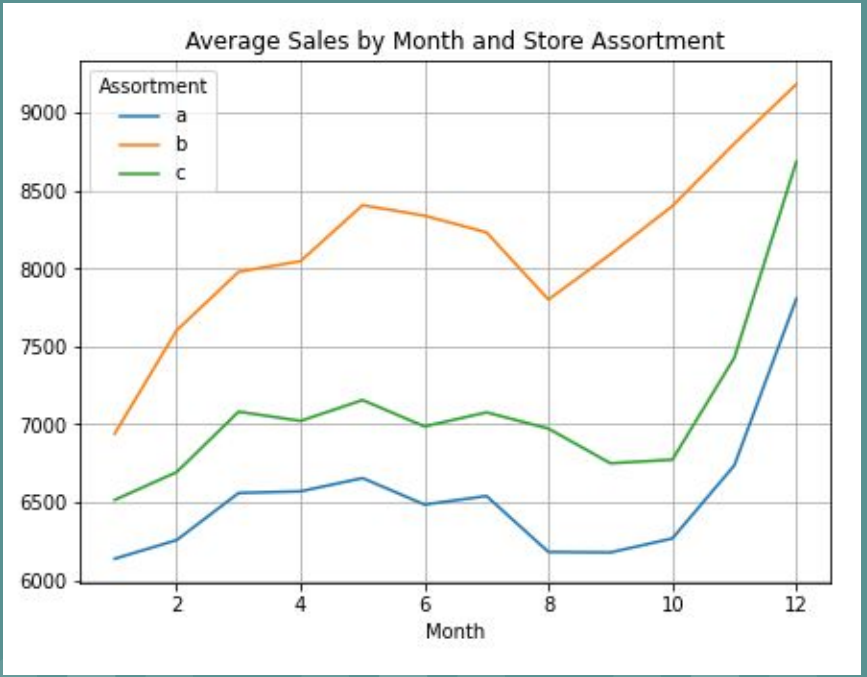
Assortment	Store
a	340968
b	6304
c	301088

- Due to the vast differences in total Sales volume based on Assortment type, lets also look at the average number of Sales.

- (Note: We could include our 2015 data since we're calculating the mean Sales now, but for the sake of consistency when comparing it with the total Sales we will continue to use the same 2013-2014 data.)

Assortment	a	b	c
Month			
1	6136.639110	6940.177449	6514.338348
2	6256.393204	7603.602459	6692.936817
3	6559.181644	7979.657744	7081.361222
4	6569.460259	8048.290875	7021.796687
5	6654.673086	8407.713262	7155.927738
6	6484.604796	8340.229630	6986.406016
7	6539.447668	8233.163188	7076.641465
8	6180.267834	7801.185958	6972.688307
9	6176.988769	8092.672549	6749.746643
10	6266.935886	8401.910816	6772.753406
11	6736.145938	8803.662745	7427.570106
12	7805.123835	9184.045541	8686.570096

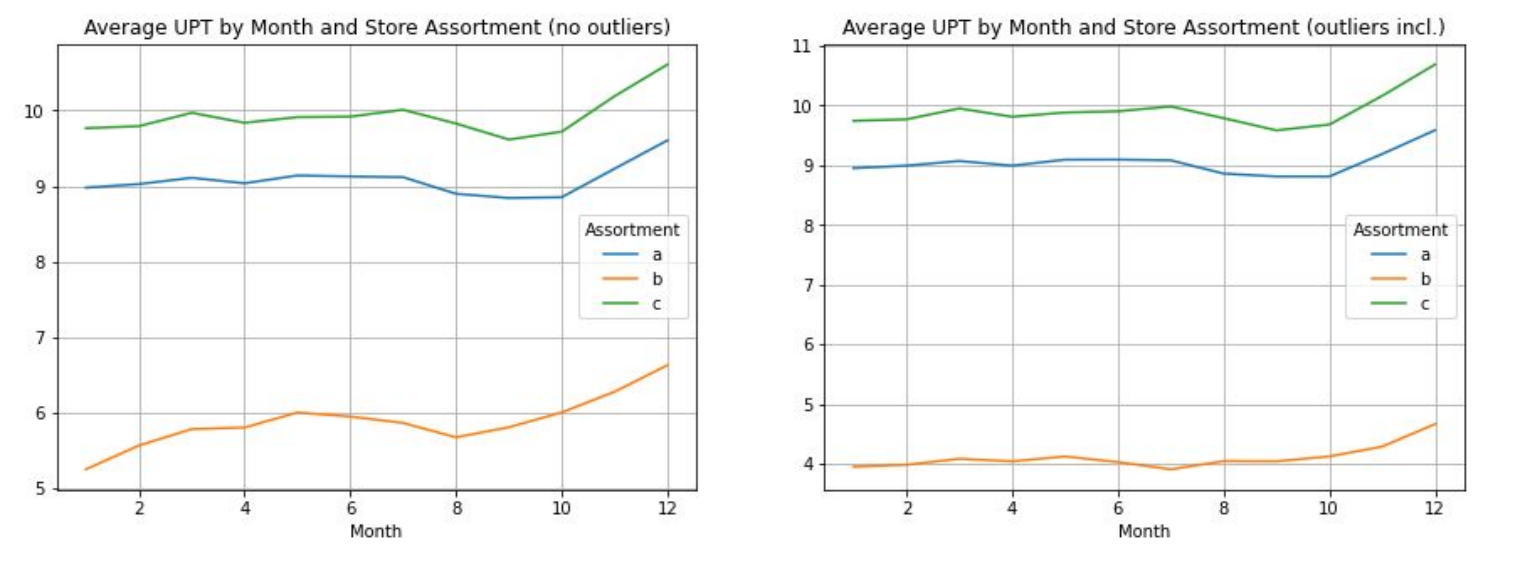
Average Sales Number
of each Assortment type
over months



Look at average Sales by store Assortment type we can see that s**tores of type B** actually perform quite well when compared to **types A and C,** despite there being significantly less type B stores! Types A and C continue to follow very similar trends for Sales, but Type C stores consistently outperform type A stores.

7. How UPT metric compares across stores of different Assortment types

- Let's also take a look at how our UPT metric compares across stores of different Assortment types.
- (Note: Because we included a UPT metric for our table that still has our extremely high Sales outliers, we will plot that too, to compare.)



Continued.....

- We can see stores of Assortment type C are our best performers for UPT. Comparing UPT with and without outliers treated, we can see that with the outliers treated we can more clearly see upward and downward trends, whereas with outliers included these trends look less impactful.
- Exploring out Sales and Customers outliers prompted us to create out UPT metric when comparing them by StoreType, so let's explore Sales and UPT by month and StoreType as well.
- Let's start with a look at how many stores of each StoreType we have.

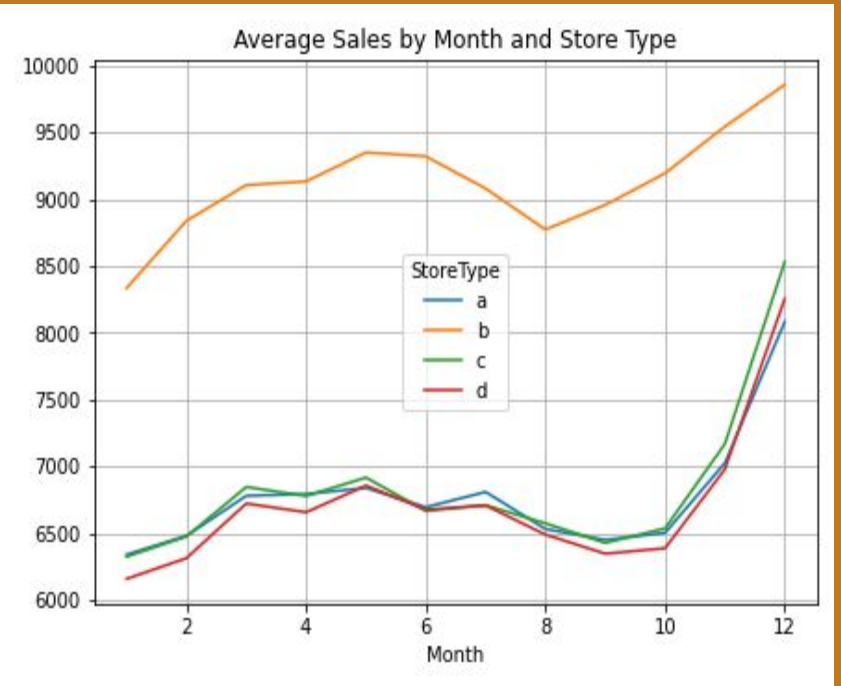
StoreType	Store
a	351476
b	11959
c	87079
d	197846

average Sales per month, broken down by StoreType.

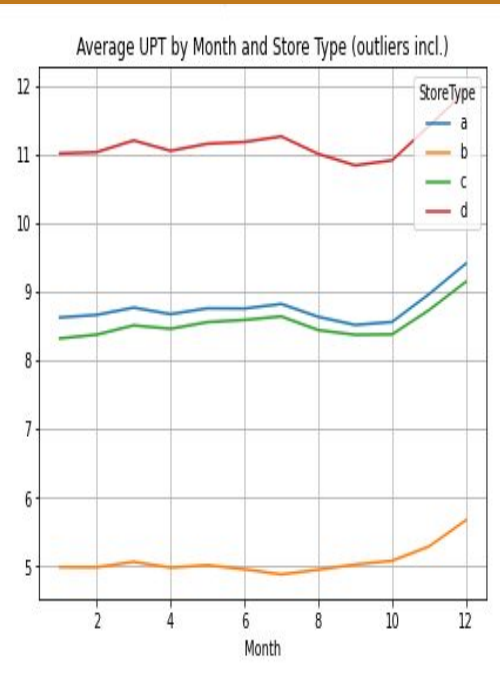
StoreType	a	b	c	d
Month				
1	6340.939457	8336.488072	6325.695836	6159.515050
2	6481.948327	8839.864537	6476.370302	6314.299544
3	6781.173209	9106.575911	6847.562350	6723.579013
4	6794.130360	9133.661885	6779.331581	6657.805468
5	6840.385002	9350.734115	6917.847151	6857.707885
6	6695.761585	9321.768687	6667.874042	6677.662636
7	6810.202874	9081.286555	6711.115334	6709.382984
8	6533.024121	8773.740958	6575.015995	6491.047891
9	6451.144734	8957.519192	6428.605156	6348.017432
10	6503.911678	9194.587488	6537.462481	6388.625455
11	7026.628471	9544.331313	7169.819236	6979.515297

Continued.....

- We can see that stores of type A, C, and D closely follow very similar trends, whereas stores of type B significantly outperform them when it comes to the average number of Sales.



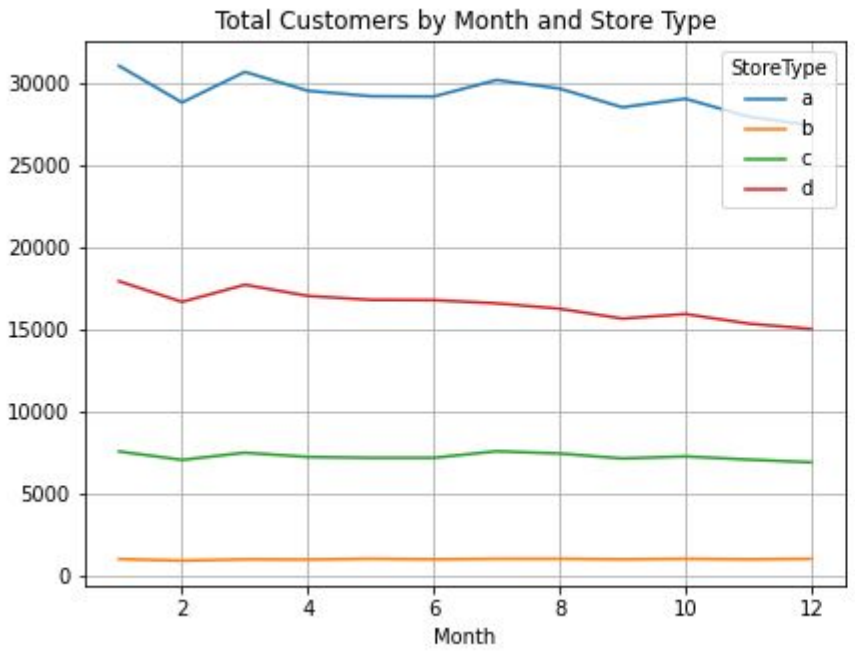
- Now let's see how UPT compares across Store Types.

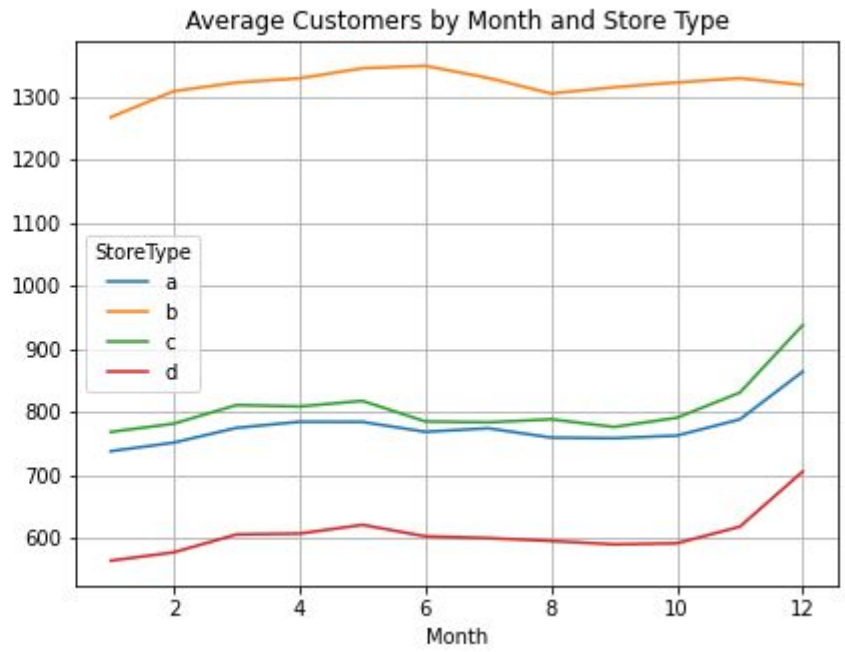


Continued.....

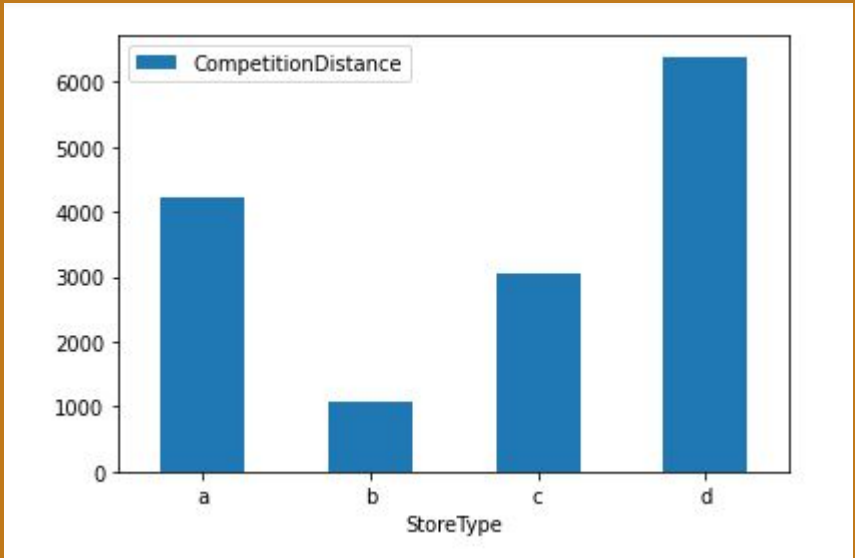
- Similar to our look at UPT across stores of different Assortment types, we can see that treating the outliers leaves us with a graph that more clearly shows changes in the trend of monthly sales. As we look at average UPT by StoreType, we see that while they follow similar trends, store of StoreType D see customers purchasing approximately 3 more items per purchase on average than at a type A or C store, and approximately 6 more items per purchase on average than customers at type B stores.
- It seems customers of type B stores buy less items per purchase on average, but overall type B stores see more sales. It would reason that type B stores must see more customers on average to account for high average sales.

- Below we will look at total customers and average customers by month. We know of the different store types that there are the fewest stores of type B, so we aren't surprised to see them at the bottom of the Total Customers chart. However, if we look at average customers by store type we see that they average far more customers than the other store types.



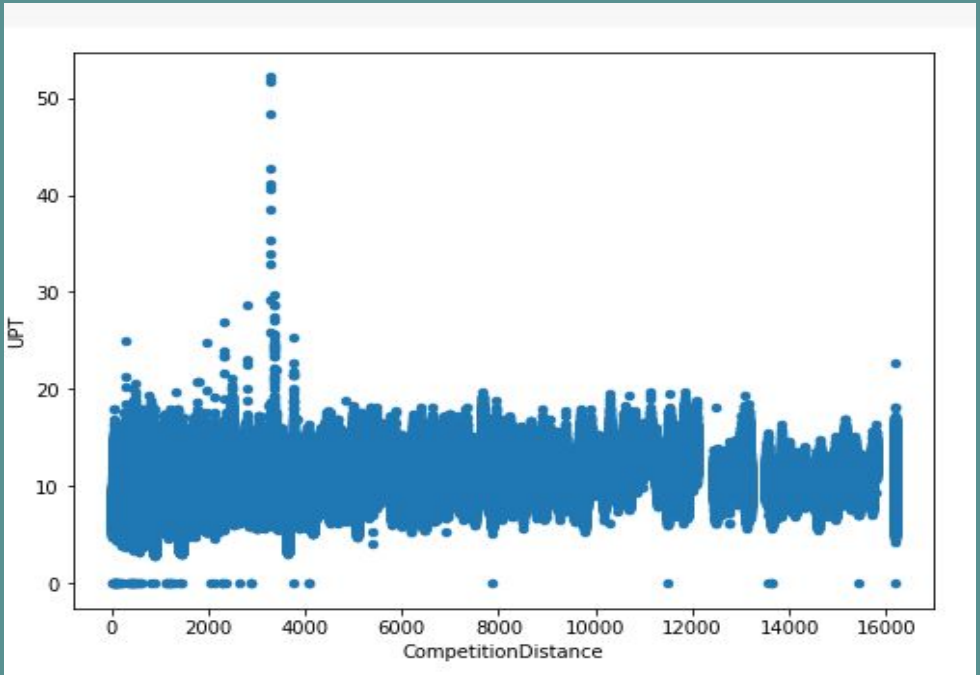


- To understand a bit more about our different Store Types, let's also quickly see how they compare in relation to the Competition Distance.

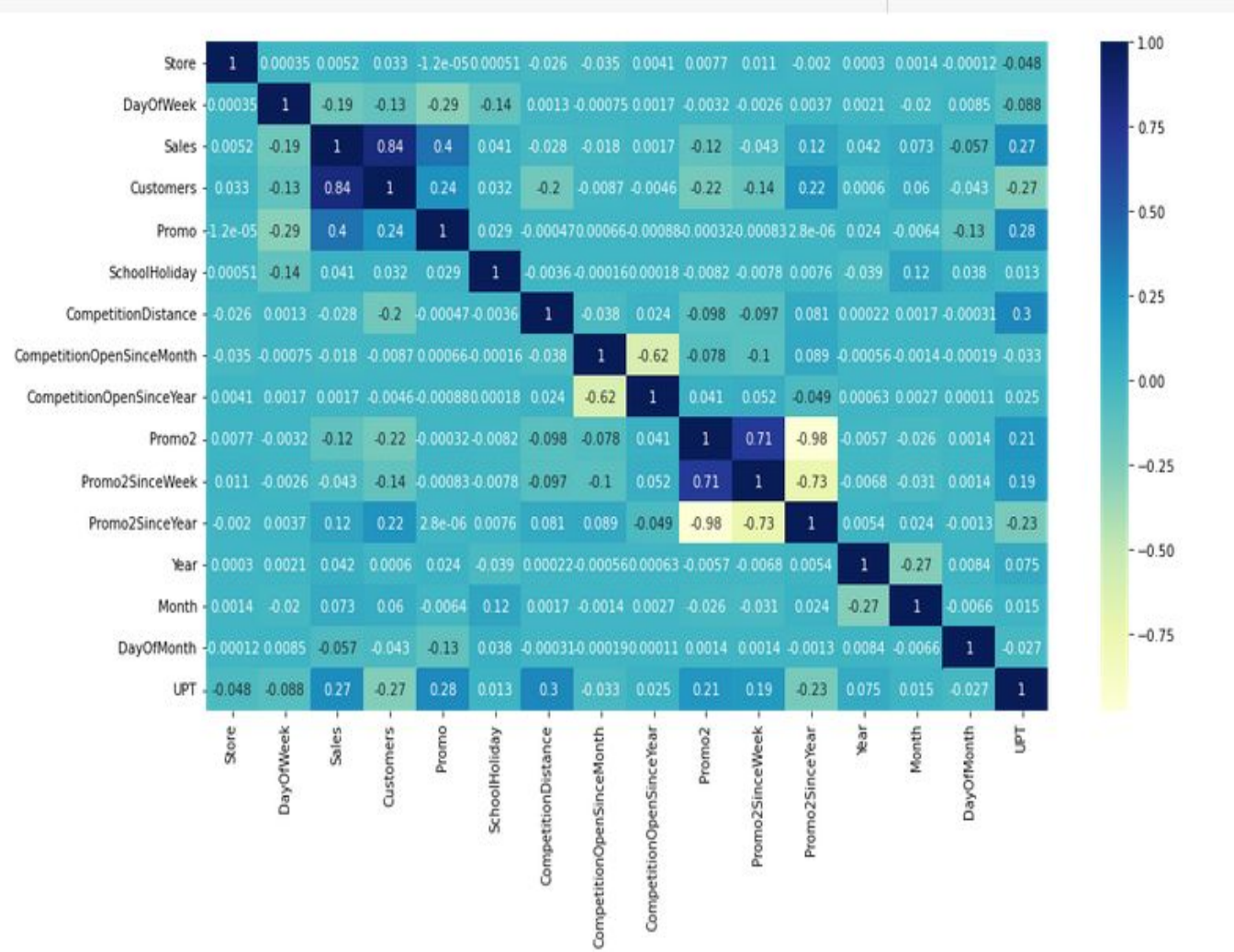


- Stores of **Store Type B** are s**ignificantly closer** to their nearest competitor store, on average. This might suggest the Type B stores are most often in **dense urban shopping areas**. Perhaps with many other stores available in closer proximity, customers are less likely but multiple items at the store when they can more easily purchase additional items at other nearby stores. Let's see if there's any correlation between **CompetitionDistance** and **UPT.**

8. Correlation between CompetitionDistance and UPT metrics



- We don't see a strong positive or negative correlation between CompetitionDistance and UPT on our scatter plot, but we do see what look to be outliers in our UPT metric.



AI

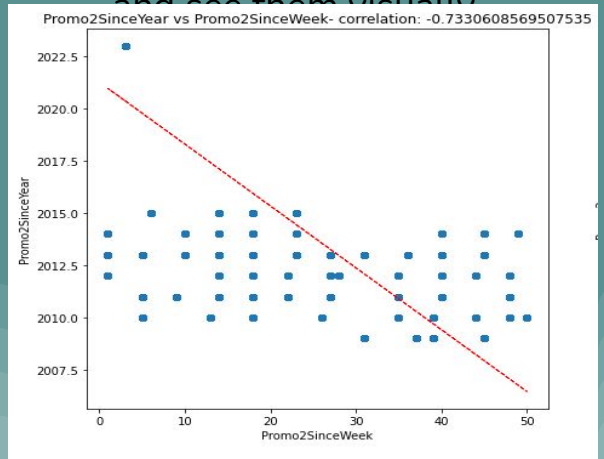
- Looking at the correlation calculations , we don't see any meaningful correlations with **CompetitionDistance.**
- The notable correlations are a strong **positive correlation** between **Customers** and **Sales**, which isn't surprising. In addition, we see a smaller but positive correlation between **Promo** and **Sales.**

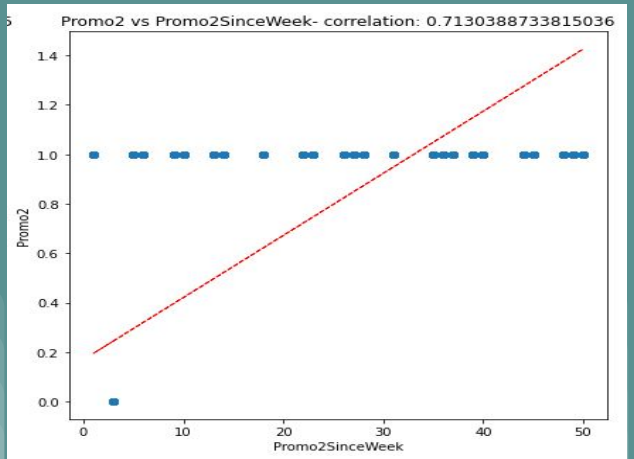
Regression Model

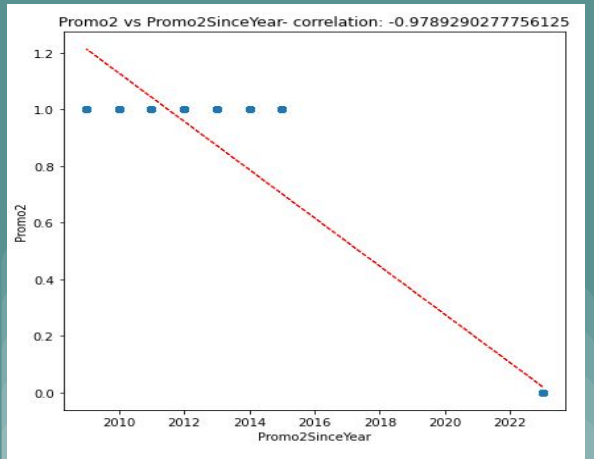
- ★ Here we will be mainly talking about three regression algorithms:
  - Simple Linear Regression
  - Lasso Regression
  - Ridge Regression
- ★ In this project , we have to predict the Sales columns considering rest of the columns as predictors.

Note : Lasso and Ridge regression are helpful in penalizing the regression coefficients and used for feature selection and better for learning the model.

- We can observe that columns **Promo2sinceWeek** & **Promo2SinceYear**,**Promo2sinceYear** and **Promo2**,**Promo2** and **Promo2sinceweek** showing some kind of correlation... Lets plot and see them visually







Lets apply the regression model considering correlated features

AI

★ Here is the result of one-fold regression model:

- MSE train :1262772.0267041493
- MSE test : 1251220.7445700767
- RMSE train :1123.731296486909
- RMSE test : 1118.579789094223
- R2_train : 0.8291506776669179
- Adjusted R2_train : 0.8291435956619773
- R2_test : 0.8298980889701556
- Adjusted R2_test : 0.8298698813686819

❖ We can see that the r2_score for both training set and test set is nearly similar and r2_score value is 0.82. So we can say 82% of the variation it can able to explain perfectly. From this we can conclude that our model not only performed well on training set but also performed well on test dataset.. And it shows that the model is not underfitting as well as not overfitting.

★ **While splitting the data we can get such a split which may give us the good result by luck, so to avoid such by chance result lets split up the data in 5 part and then apply the regression model and then average the mean_squred error and check.**

★ **As I have already splitted X and y in 5 different dataset. Lets see the result of model on each fold**

For 1st Fold

- MSE is 1325319.53019872
- RMSE is 1151.22523000441
- r2score is 0.8164021067836
- MAE is 876.2341518297678

For 2nd Fold

- MSE is 1295484.5989173402
- RMSE is 1138.193568299057
- r2score is 0.82951668585014
- MAE is 881.8630900862344

For 3rd Fold

- MSE is 1235444.7456977929
- RMSE is 1111.50562108240
- r2score is 0.82924768052350
- MAE is 853.7869031476845

For 4th Fold

- MSE is 1304636.8812892754
- RMSE is 1142.207022080181
- r2score is 0.83030902821623
- MAE is 870.1566965311616

For 5th Fold

- MSE is 1200492.3566388097
- RMSE is 1095.669820994815
- r2score is 0.8300340565396059
- MAE is 845.0850988565812

Final average result of these 5 folds:

Final Result :

- Mean Squared Error; 1272275.6225483834
- Root Mean Squared Error: 1127.7602524921738
- R2 Score Values: 0.8271019115826211
- Mean Absolute Error: 865.425188090279

❖ **From above information we can see the on every fold we got the approx 0.83 and the Avg. r2 score is 0.83 which represent good model for prediction**

- Lets check the **Variance Inflation Factor** for these 3 columns. **Promo2", "Promo2SinceYea "Promo2SinceWeek"**

variables	VIF
Promo2	4.050637
Promo2SinceYear	2.073812
Promo2SinceWeek	3.764762

- From the above results I can see that **Promo2** has the highest variance inflation factor...
- And even **promo2** has significant **positive correlation** with the **Promo2Since week** and very high negative correlation so....
- I have decided to drop the **promo2** column from regression_data and make a new copy of it

- **After running the simple regression model after removing the correlated feature the results are:**

Final Result:
Mean Squared Error; 1272275.6225483876 Root Mean Squared Error: 1127.7602524921756 R2 Score Values: 0.8271019115826206 Mean Absolute Error: 865.4251880902859

➢ **From above results we can see that there is not much significant difference on the result compared to the previous results if we go by performance ..But if feature selection is our main aim in that case we should go with this model...**

Lets Apply the Regularization Algorithms

Lasso Regression

- ★ We know that Lasso regularization is used to penalize the regression coefficients in such a way that the model should not be overfitted and to avoid multicollinearity by doing the feature selection by making the regression coefficient of unnecessary feature close to zero sometimes even = 0. **In lasso regularization, model tries to minimize the loss function using absolute sum of regression weights**
- ★ After applying the **hyperparameter tuning** with **alpha values** [1e-9,1e-5,1e-1,1,10,1e4,1e9] ,we narrowed down our range of alpha parameters to => `[1e-5,0.0003,1e-4]` using **GridSearchCV.**
- ★ **The final Alpha value for this model is => 0.0003**

Evaluation Result of this models :

Lasso Score for Training Set : 0.8291750336539018
MSE train :1262592.008085251
MSE test : 1251113.8673249667
RMSE train :1123.6511950268425
RMSE test : 1118.5320144389998
R2_train : 0.8299046120516298
Adjusted R2_train : 0.8291676997550116
R2_test : 0.8299046120516298
Adjusted R2_test : 0.8298834061059761

	features	coefficients
0	Store	-89.311915
1	DayOfWeek	-92.153659
2	DayOfMonth	44.610006
3	Month	332.469659
4	Year	341.000315
5	Customers	12138.544953
6	Promo	980.863152
7	StateHoliday_0	-129.347356
8	StateHoliday_a	-97.197423
9	StateHoliday_b	94.967249
10	StateHoliday_c	721.458835
11	SchoolHoliday	26.253182
12	StoreType_a	-755.542358
13	StoreType_b	-1196.469518
14	StoreType_c	-927.277166
15	StoreType_d	383.170257
16	Assortment_a	-266.830913
17	Assortment_b	-2714.931195
18	Assortment_c	56.158758
19	CompetitionDistance	837.395244
20	CompetitionOpenSinceMonth	-32.478084
21	CompetitionOpenSinceYear	-883.691671
22	Promo2	364.348883
23	Promo2SinceWeek	554.197355
24	Promo2SinceYear	-706.357474
25	PromoInterval_Feb-Nov	-657.026853
26	PromoInterval_Jan-Oct	-519.839532
27	PromoInterval_Mar-Dec	-763.105760
28	PromoInterval_NA	186.757057

- Coefficient for Lasso regression

- We can see that the Lasso has unselected some unrelevant features by making the coef_ value close to zero And more than that Lasso performed better than Simple linear regression

Lets Apply Ridge Regression

Ridge Regression

- ★ We know that Ridge regularization is used to penalize the regression coefficients in such a way that the model should not be overfitted and to avoid multicollinearity by doing the feature selection by making the regression coefficient of unnecessary feature close to zero but not != 0. **In Ridge regularization, model tries to minimize the loss function using sum of squares of regression weights**
- ★ After applying the **hyperparameter tuning** with **alpha values** `[1e-15,1e-10,1e-8,1e-5,1e-4,1e-3,1e-2,1,5,10,20,30,40,45,50,55,60,100]` **GridSearchCV.**
- ★ **The final Alpha value for this model is => 5**

Evaluation Result of this models :

Ridge Score for Training Set : 0.8291752605603164
MSE train :1262590.3309871552
MSE test : 1251097.5771071059
RMSE train :1123.6504487549298
RMSE test : 1118.5247324521285
R2_train : 0.8299046120516298
Adjusted R2_train : 0.8291679266711678
R2_test : 0.8299046120516298
Adjusted R2_test : 0.8298856211212906

	features	coefficients
0	Store	-89.103727
1	DayOfWeek	-92.328361
2	DayOfMonth	44.586571
3	Month	332.492279
4	Year	340.962823
5	Customers	12135.583784
6	Promo	981.083131
7	StateHoliday_0	-266.110977
8	StateHoliday_a	-232.264060
9	StateHoliday_b	-40.247260
10	StateHoliday_c	538.622297
11	SchoolHoliday	26.279685
12	StoreType_a	-131.393003
13	StoreType_b	-572.063521
14	StoreType_c	-303.415930
15	StoreType_d	1006.872455
16	Assortment_a	707.365121
17	Assortment_b	-1737.926551
18	Assortment_c	1030.561429
19	CompetitionDistance	836.657938
20	CompetitionOpenSinceMonth	-32.096302
21	CompetitionOpenSinceYear	-881.907466
22	Promo2	-206.835148
23	Promo2SinceWeek	552.263734
24	Promo2SinceYear	-723.935219
25	PromoInterval_Feb-Nov	-79.730069
26	PromoInterval_Jan-Oct	57.760568
27	PromoInterval_Mar-Dec	-184.865643
28	PromoInterval_NA	206.835151

- Coefficient for Ridge regression

- So we got the best alpha parameter for ridge regression is **5**. Which will have significant impact for penalizing the weights

Lets Apply Elastic Regression

Elastic Net Regression

- ★ **In ElasticNet regression, which is a combination of Lasso regression and Ridge regression.**
- ★ **It is a regularized linear regression method that tries to balance the trade-off between the Ridge and Lasso methods.**
- ★ **After applying the hyperparameter tuning** with **alpha values** `[0.001,0.003,0.004]` **using GridSearchCV.**
- ★ **The final Alpha value for this model is => 0.001**

Evaluation Result of this models :

Elastic Score for Training Set : 0.8291753284280492
MSE train :1262589.8293669368
MSE test : 1251104.3257339278
RMSE train :1123.6502255448254
RMSE test : 1118.527749201569
R2_train : 0.8299046120516298
Adjusted R2_train : 0.8291679945418143
R2_test : 0.8299046120516298
Adjusted R2_test : 0.829884703496254

	features	coeficients
0	Store	-89.111650
1	DayOfWeek	-92.139578
2	DayOfMonth	44.603491
3	Month	332.397026
4	Year	340.962481
5	Customers	12138.624807
6	Promo	980.846701
7	StateHoliday_0	-130.179366
8	StateHoliday_a	-96.933107
9	StateHoliday_b	88.236740
10	StateHoliday_c	709.218508
11	SchoolHoliday	26.253733
12	StoreType_a	-753.992869
13	StoreType_b	-1194.739295
14	StoreType_c	-926.089576
15	StoreType_d	384.598594
16	Assortment_a	-266.156289
17	Assortment_b	-2713.874803
18	Assortment_c	56.961419
19	CompetitionDistance	836.985071
20	CompetitionOpenSinceMonth	-32.288332
21	CompetitionOpenSinceYear	-983.871501
22	Promo2	362.714829
23	Promo2SinceWeek	552.193573
24	Promo2SinceYear	-724.692021
25	PromoInterval_Feb-Nov	-665.742354
26	PromoInterval_Jan-Oct	-528.311193
27	PromoInterval_Mar-Dec	-770.801104
28	PromoInterval_NA	190.678521

- Coefficient for Elastic Net Regression

- As the **Elastic Regularization** is the blend of both **Lasso and Ridge Regression** We will use this as our final model. Because **Lasso model** has improved a little bit than Simple linear regression in feature selection but **Ridge regression** having **alpha value 5** is also significantly contributing towards penalizing the weights, which could definately impact the feature selection. As the Elastic Net Regression has the alpha value 0.001 greater than Lasso regression alpha value 0.001 can have significant regularization effect on the trained model. Hence we will be using Elastic Net Regularization model as our final model.
- Note In all the models : r2_score is nearly the same... So will not be comparing the model on this basis.

Conclusion

1)When looking at key performance indicators across store Assortments and Store Types we see they follow similar monthly trends, but numbers can vary by Assortment and Store Type.

2)When looking at store Assortment we found that stores of Assortment Type B represent a small share of total sales, which isn't surprising considering less than 1% of stores are of the Type B assortment. However, when looking at average monthly sales we see the Assortment Type B stores out performing types A and C by a large margin. Looking at the number of units sold per transaction, UPT, we see Assortment Type A and C stores performing better than Type B.

3)A similar investigation into sales by Store Type also saw noticeable differences between Store Type B and Store Types A, C, and D. Again, Type B is the least common Store Type, representing only 1.8% of stores sampled from 2013-2014. And yet, similar to Assortment Type B, Store Type B also vastly outperforms other Store Types when looking at average monthly sales. Following the similarities, it also lags to the bottom when we compare UPT amongst Store Types.

Continued…….

4) In trying to further investigate that difference between average sales and average UPT for Store Type B, we looked at average customers by Store Type and found Type B to be well ahead of the others. This suggests that Type B stores on average have more customers visit, but they buy a smaller number of items. In looking at the average distance of the nearest competitor by store type we found that Type B stores were far closer to their nearest competitor store on average. This lead us to hypothesize that Type B stores may be concentrated in dense urban areas that see more foot traffic.

5) Further along that line of investigation, we looked for any correlations with competition distance, but found no strong correlations. Looking at correlations across our complete data set we only found the expected strong correlation between Customers and Sales, and smaller correlations between Promo and Sales.

6) We conclude that well Store Type B and store Assortment B represent a very small sample of the stores, they significantly outperform other store types and assortments in average sales, despite a lower UPT, and see a high volume of customers. It could be worthwhile investigating expanding into more Type B Stores and Assortments.

Continued…….

7)Finally we applied the various regression models , Simple Linear regression, Lasso,Ridge and Elastic net regression. After evaluating their metrics we finally chose to go with Elastic Net Regression.

Thank You!


Why Some Ideas Survive and Others Die
MADE
to
STICK
Chip Heath & Dan Heath