A Project Report

On

US STORES SALES

By Group - AKHANDA

WSU ID:	**NAME:**	**E-MAILS:**
K968X456	Vijaya Ramya CH	vxchilimutha@shockers.wichita.edu
F233Z974	Vinay Chowdari Mandava	vxmandava2@shockers.wichita.edu
Q684A655	Ravikiran Nallamothu	rxnallamothu@shockers.wichita.edu
J639H476	Lokesh Muppalla	lxmuppalla@shockers.wichita.edu
T779k442	Vijaykumar Ethakota	Vxethakota@wichita.edu
H664V993	Sai Manish Koganti	sxkoganti1@shockers.wichita.edu

Submitted to

ALDEN WILNER

CS746-Perspectives On Data Science

School of Computing Department
Wichita State University

ABSTRACT:

This dataset contains information about Sales Values in Dollars on American Stores between 2010 and 2011. This dataset contains:Stores' Area, State, Region and Size; Products' ID, Description, Type, Category and Sale Date; Accounting Info, such as Budget Margin, Profit, Total Expenses and Marking.

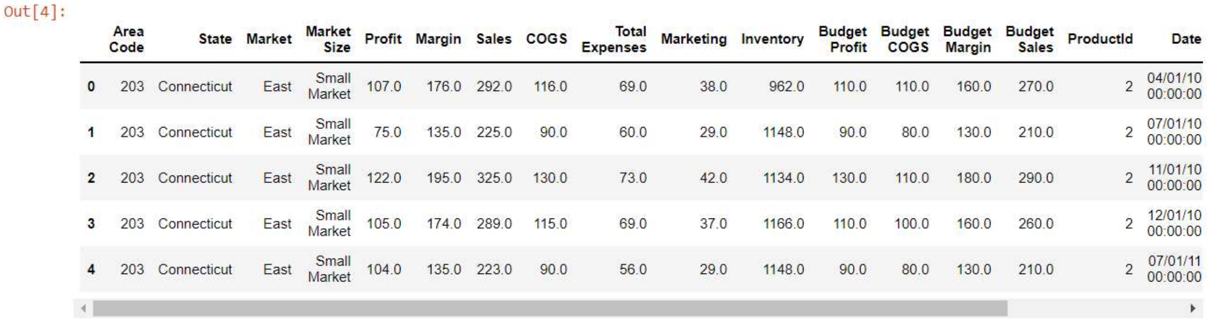
INTRODUCTION:

There are few terms to know before going to the actual project

Profit: Profit is the money you have left after paying for business expenses
Margin: margin is the difference between the price at which a product is sold and the costs associated with making or selling the product (or cost of goods sold).
Sales: sales refer to any transactions where money or value is exchanged for the ownership of a good or entitlement to a service.
COGS: Cost of goods sold (COGS) is the direct cost of making a company's products.
Total Expenses: A company's total expenses refer to the sum of its costs spent toward running the business.
Marketing: Marketing expense is comprised of those costs incurred to present an organization's goods and services to prospective customers
Inventory: Inventory is the raw materials used to produce goods as well as the goods that are available for sale.
Budget: Budget is the amount that we were expecting when we made a budget.

A. Data Description:

The chosen data set must be cleaned so that we can work on it for the processing. This is how the original data set looks like.



Out[4]:

	Area Code	State	Market	Market Size	Profit	Margin	Sales	COGS	Total Expenses	Marketing	Inventory	Budget Profit	Budget COGS	Budget Margin	Budget Sales	ProductId	Date
0	203	Connecticut	East	Small Market	107.0	176.0	292.0	116.0	69.0	38.0	962.0	110.0	110.0	160.0	270.0	2	04/01/10 00:00:00
1	203	Connecticut	East	Small Market	75.0	135.0	225.0	90.0	60.0	29.0	1148.0	90.0	80.0	130.0	210.0	2	07/01/10 00:00:00
2	203	Connecticut	East	Small Market	122.0	195.0	325.0	130.0	73.0	42.0	1134.0	130.0	110.0	180.0	290.0	2	11/01/10 00:00:00
3	203	Connecticut	East	Small Market	105.0	174.0	289.0	115.0	69.0	37.0	1166.0	110.0	100.0	160.0	260.0	2	12/01/10 00:00:00
4	203	Connecticut	East	Small Market	104.0	135.0	223.0	90.0	56.0	29.0	1148.0	90.0	80.0	130.0	210.0	2	07/01/11 00:00:00

The dimension of the dataset is 4119 rows × 19 columns. It contains data which has many data types in the table. It contains Strings, Serial Numbers, percentages, and currencies with their respective symbols before the numbers.

We cleaned the data so that it should be compatible for processing in finding a better model.

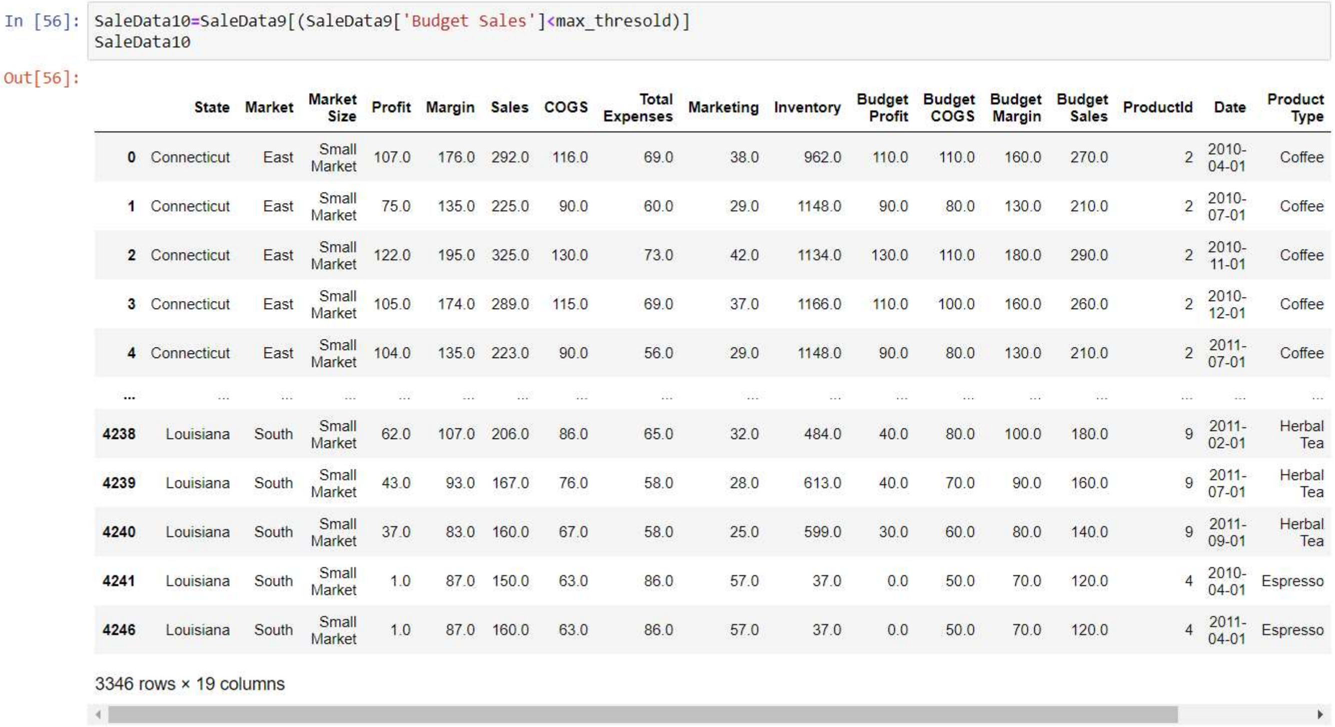
Statistical Overview:

Inventory minimum values are negative: it's not possible to happen, because at the moment the inventory is equal to zero, the market doesn't have the product physically to sell. So, let's make a small Data Transformation to convert all negative Inventories to zero;

High Standard Deviation: take a look at 75% and max stats from **Sales** Feature, the difference between these two is 930.00 dollars, quite a big gap! Due to this the Standard Deviation is very high (148,.89 dollars), so, it'll be a problem in the future if we don't take an action to fix it;

B. Data Cleaning:

After clearing the outliers in all columns we get the final data set with 3346 rows × 19 columns dimensions.



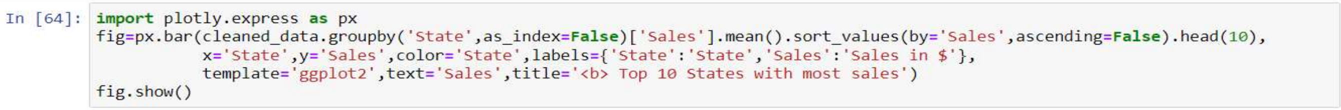
```
In [56]: SaleData10=SaleData9[(SaleData9['Budget Sales']<max_thresold)]
         SaleData10
```

Out[56]:

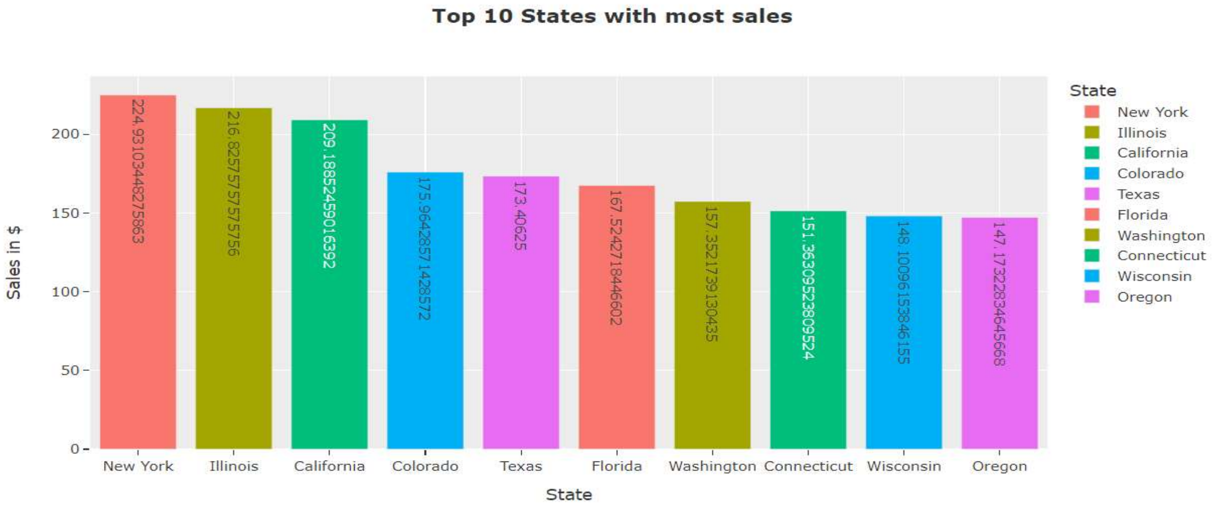
	State	Market	Market Size	Profit	Margin	Sales	COGS	Total Expenses	Marketing	Inventory	Budget Profit	Budget COGS	Budget Margin	Budget Sales	ProductId	Date	Product Type
0	Connecticut	East	Small Market	107.0	176.0	292.0	116.0	69.0	38.0	962.0	110.0	110.0	160.0	270.0	2	2010-04-01	Coffee
1	Connecticut	East	Small Market	75.0	135.0	225.0	90.0	60.0	29.0	1148.0	90.0	80.0	130.0	210.0	2	2010-07-01	Coffee
2	Connecticut	East	Small Market	122.0	195.0	325.0	130.0	73.0	42.0	1134.0	130.0	110.0	180.0	290.0	2	2010-11-01	Coffee
3	Connecticut	East	Small Market	105.0	174.0	289.0	115.0	69.0	37.0	1166.0	110.0	100.0	160.0	260.0	2	2010-12-01	Coffee
4	Connecticut	East	Small Market	104.0	135.0	223.0	90.0	56.0	29.0	1148.0	90.0	80.0	130.0	210.0	2	2011-07-01	Coffee
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4238	Louisiana	South	Small Market	62.0	107.0	206.0	86.0	65.0	32.0	484.0	40.0	80.0	100.0	180.0	9	2011-02-01	Herbal Tea
4239	Louisiana	South	Small Market	43.0	93.0	167.0	76.0	58.0	28.0	613.0	40.0	70.0	90.0	160.0	9	2011-07-01	Herbal Tea
4240	Louisiana	South	Small Market	37.0	83.0	160.0	67.0	58.0	25.0	599.0	30.0	60.0	80.0	140.0	9	2011-09-01	Herbal Tea
4241	Louisiana	South	Small Market	1.0	87.0	150.0	63.0	86.0	57.0	37.0	0.0	50.0	70.0	120.0	4	2010-04-01	Espresso
4246	Louisiana	South	Small Market	1.0	87.0	160.0	63.0	86.0	57.0	37.0	0.0	50.0	70.0	120.0	4	2011-04-01	Espresso

3346 rows × 19 columns

C. Exploratory Data Analysis:

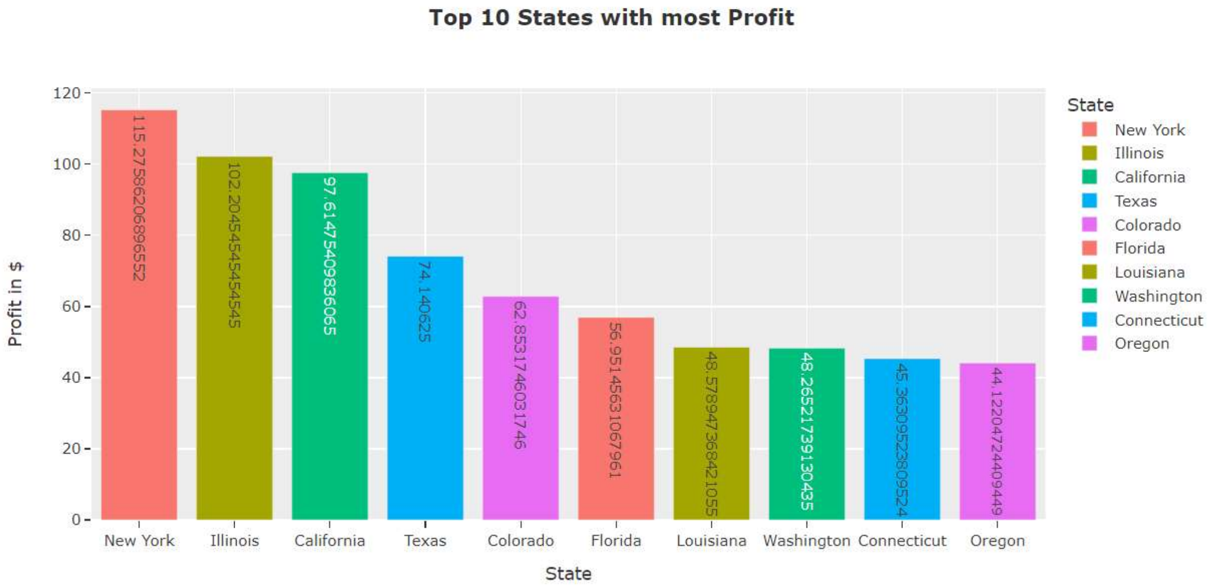


```
In [64]: import plotly.express as px
         fig=px.bar(cleaned_data.groupby('State',as_index=False)['Sales'].mean().sort_values(by='Sales',ascending=False).head(10),
                    x='State',y='Sales',color='State',labels={'State':'State','Sales':'Sales in $'},
                    template='ggplot2',text='Sales',title='<b> Top 10 States with most sales')
         fig.show()
```



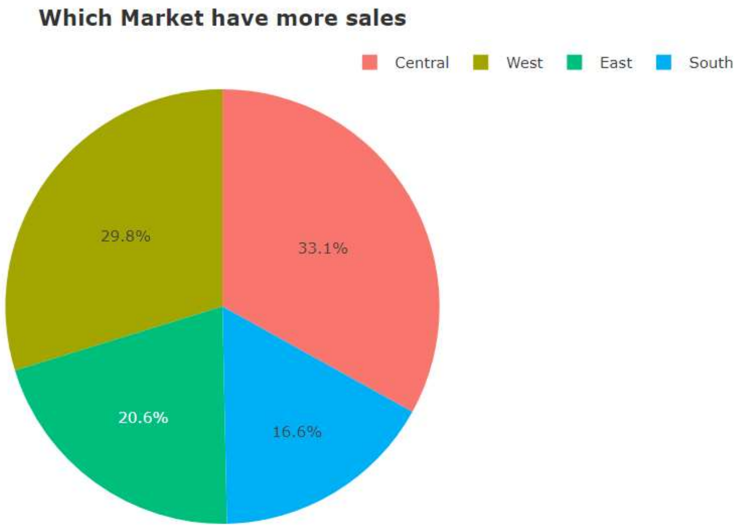
From the above plot we can see that states New York and Illinois have top sales.Wisconsin,Oregon have lease sales.

```
In [65]: fig=px.bar(cleaned_data.groupby('State',as_index=False)['Profit'].mean().sort_values(by='Profit',ascending=False).head(10),
                    x='State',y='Profit',color='State',labels={'State':'State','Profit':'Profit in $'},
                    template='ggplot2',text='Profit',title='<b> Top 10 States with most Profit')
         fig.show()
```

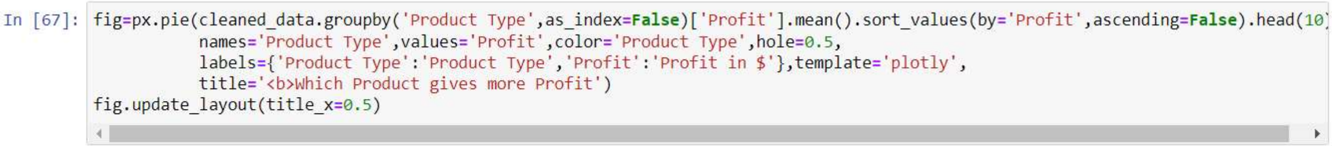


States New York and Illinois have high profits,whereas Connecticut and Oregon have least profits.

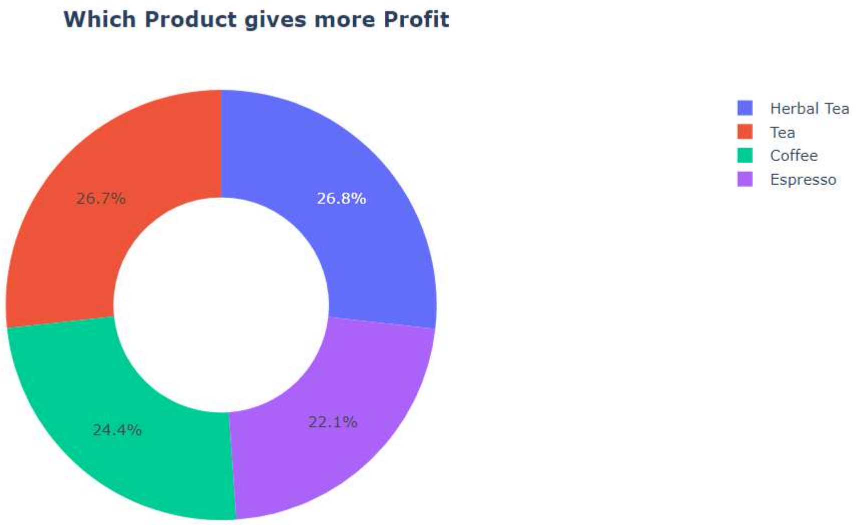
```
In [66]: fig=px.pie(cleaned_data.groupby('Market',as_index=False)['Sales'].count().sort_values(by='Sales',ascending=False).head(10),
                    names='Market',values='Sales',color='Market',
                    labels={'Market':'Market ','Sales':'Sales Count'},template='ggplot2',
                    title='<b>Which Market have more sales')
         fig.update_layout(title_x=0.5,legend=dict(orientation='h',yanchor='bottom',
                                                   y=1.02,xanchor='right',x=1))
```



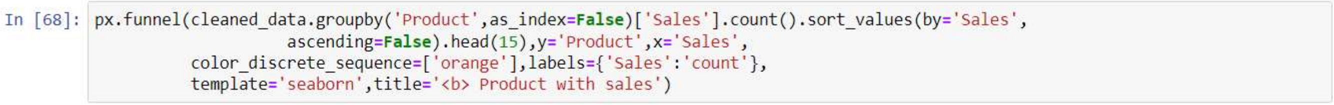
Central and West Markets have more sales,whereas East and South have comparatively low sales.



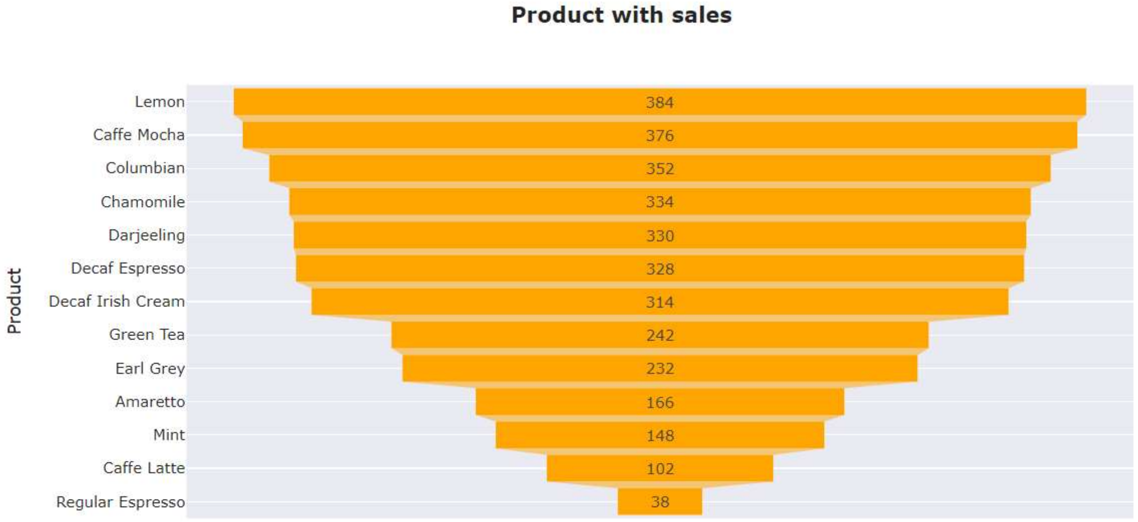
```
In [67]: fig=px.pie(cleaned_data.groupby('Product Type',as_index=False)['Profit'].mean().sort_values(by='Profit',ascending=False).head(10
                    names='Product Type',values='Profit',color='Product Type',hole=0.5,
                    labels={'Product Type':'Product Type','Profit':'Profit in $'},template='plotly',
                    title='<b>Which Product gives more Profit')
         fig.update_layout(title_x=0.5)
```



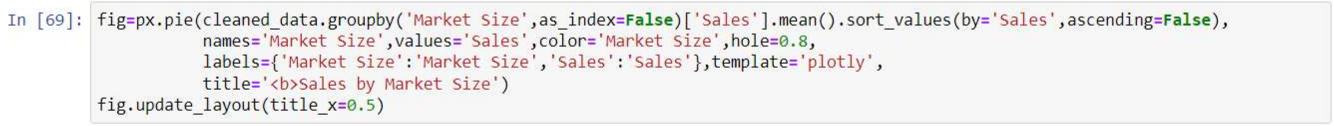
Herbal tea and tea gives more profit in US Sales Market



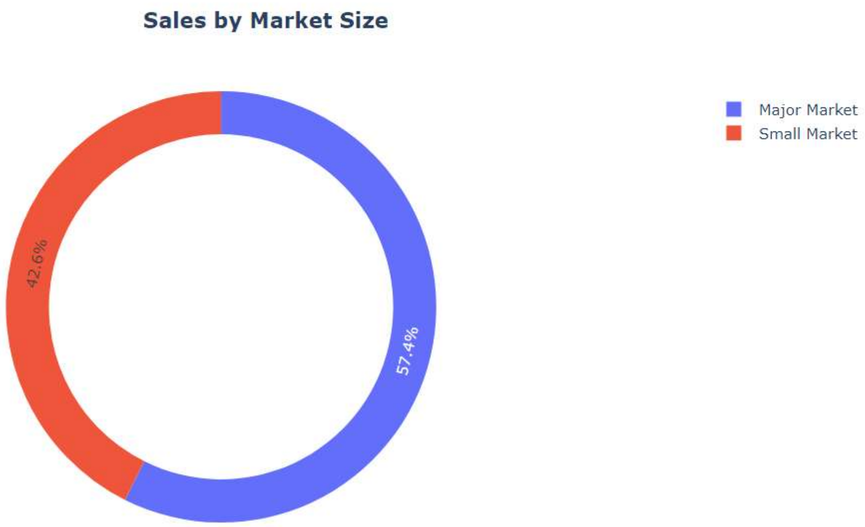
```
In [68]: px.funnel(cleaned_data.groupby('Product',as_index=False)['Sales'].count().sort_values(by='Sales',
                               ascending=False).head(15),y='Product',x='Sales',
                   color_discrete_sequence=['orange'],labels={'Sales':'count'},
                   template='seaborn',title='<b> Product with sales')
```

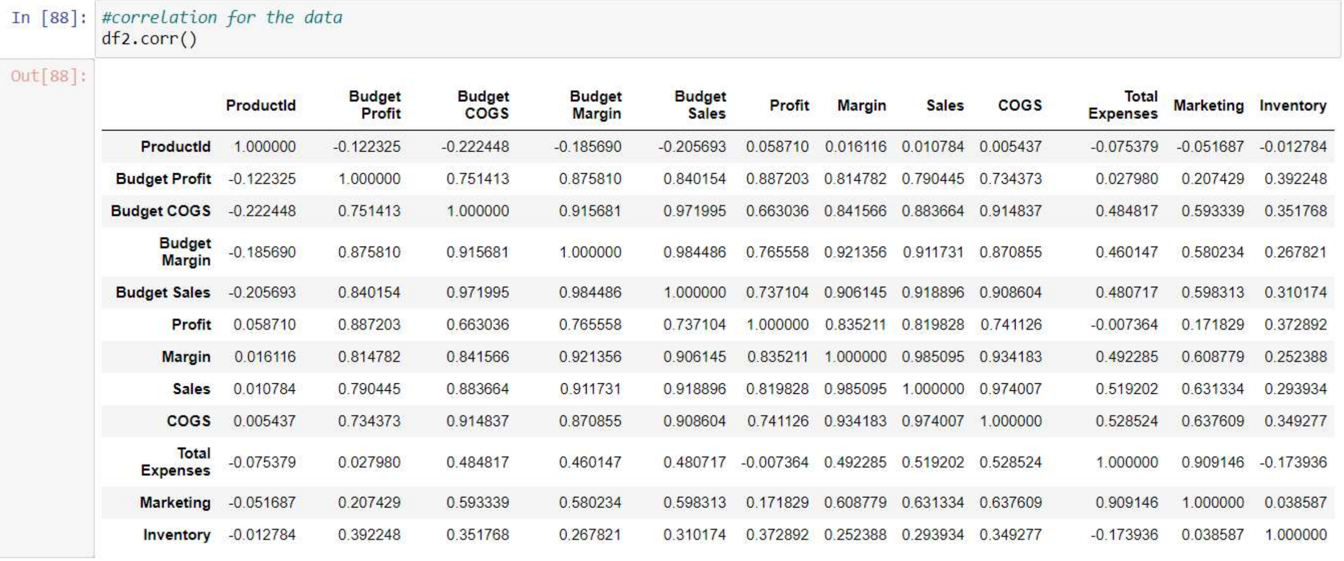


In products,lemon,caffe mocha, and Coulumbian have more sales.Whereas Mint, Caffe Latte and Regular Espresso have less product sales.



```
In [69]: fig=px.pie(cleaned_data.groupby('Market Size',as_index=False)['Sales'].mean().sort_values(by='Sales',ascending=False),
                    names='Market Size',values='Sales',color='Market Size',hole=0.8,
                    labels={'Market Size':'Market Size','Sales':'Sales'},template='plotly',
                    title='<b>Sales by Market Size')
         fig.update_layout(title_x=0.5)
```





```
In [88]: #correlation for the data
         df2.corr()
```

Out[88]:

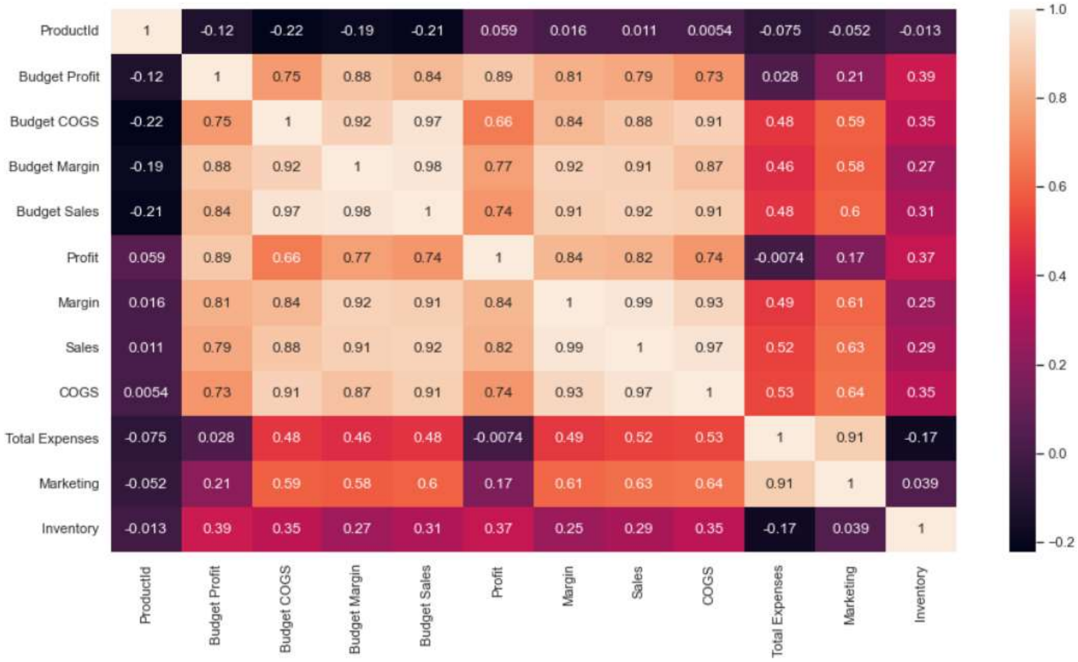
	ProductId	Budget Profit	Budget COGS	Budget Margin	Budget Sales	Profit	Margin	Sales	COGS	Total Expenses	Marketing	Inventory
ProductId	1.000000	-0.122325	-0.222448	-0.185690	-0.205693	0.058710	0.016116	0.010784	0.005437	-0.075379	-0.051687	-0.012784
Budget Profit	-0.122325	1.000000	0.751413	0.875810	0.840154	0.887203	0.814782	0.790445	0.734373	0.027980	0.207429	0.392248
Budget COGS	-0.222448	0.751413	1.000000	0.915681	0.971995	0.663036	0.841566	0.883664	0.914837	0.484817	0.593339	0.351768
Budget Margin	-0.185690	0.875810	0.915681	1.000000	0.984486	0.765558	0.921356	0.911731	0.870855	0.460147	0.580234	0.267821
Budget Sales	-0.205693	0.840154	0.971995	0.984486	1.000000	0.737104	0.906145	0.918896	0.908604	0.480717	0.598313	0.310174
Profit	0.058710	0.887203	0.663036	0.765558	0.737104	1.000000	0.835211	0.819828	0.741126	-0.007364	0.171829	0.372892
Margin	0.016116	0.814782	0.841566	0.921356	0.906145	0.835211	1.000000	0.985095	0.934183	0.492285	0.608779	0.252388
Sales	0.010784	0.790445	0.883664	0.911731	0.918896	0.819828	0.985095	1.000000	0.974007	0.519202	0.631334	0.293934
COGS	0.005437	0.734373	0.914837	0.870855	0.908604	0.741126	0.934183	0.974007	1.000000	0.528524	0.637609	0.349277
Total Expenses	-0.075379	0.027980	0.484817	0.460147	0.480717	-0.007364	0.492285	0.519202	0.528524	1.000000	0.909146	-0.173936
Marketing	-0.051687	0.207429	0.593339	0.580234	0.598313	0.171829	0.608779	0.631334	0.637609	0.909146	1.000000	0.038587
Inventory	-0.012784	0.392248	0.351768	0.267821	0.310174	0.372892	0.252388	0.293934	0.349277	-0.173936	0.038587	1.000000

Linear Relationships:

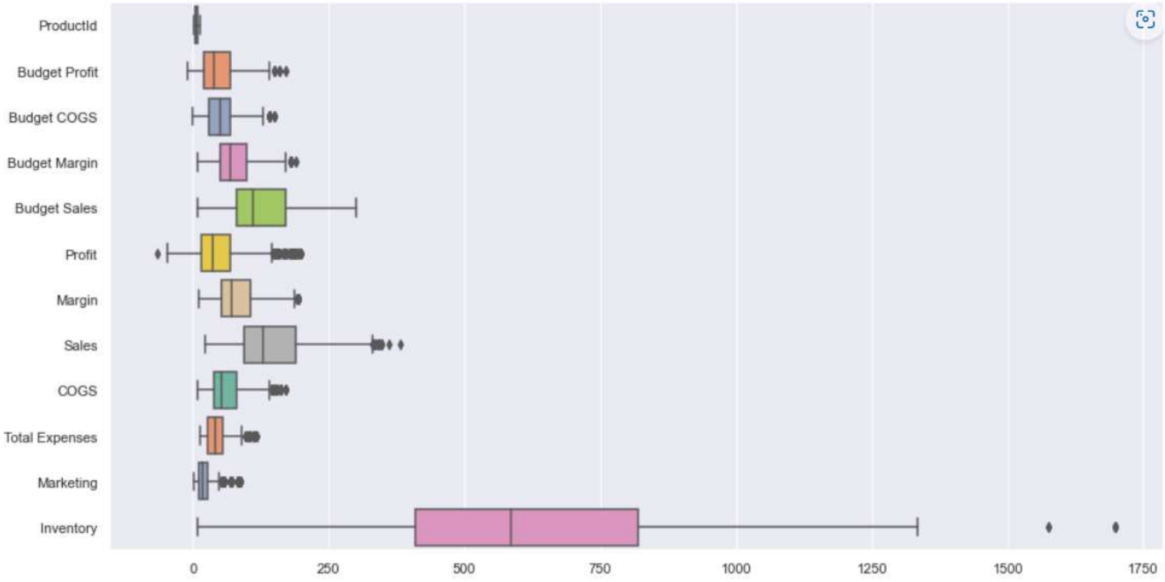
Marketing - Inventory (38.58%): as more marketing the market does for a specific as higher will be the Product Inventory. One of the reasons here is due to the belief that more clients will be interested to buy the shared products and then, the market will more units of the products to sell;

Total Expenses - Marketing (90.91%): as higher is Total Expenses as higher is the costs in Marketing. It means that markets with more costs gain more money and, consequently, can apply more money in the Marketing.

```
In [90]: #the correlation values of heatmap
         sns.set(rc={'figure.figsize':(15,8)})
         sns.heatmap(df2.corr(), annot=True)
         plt.show()
```

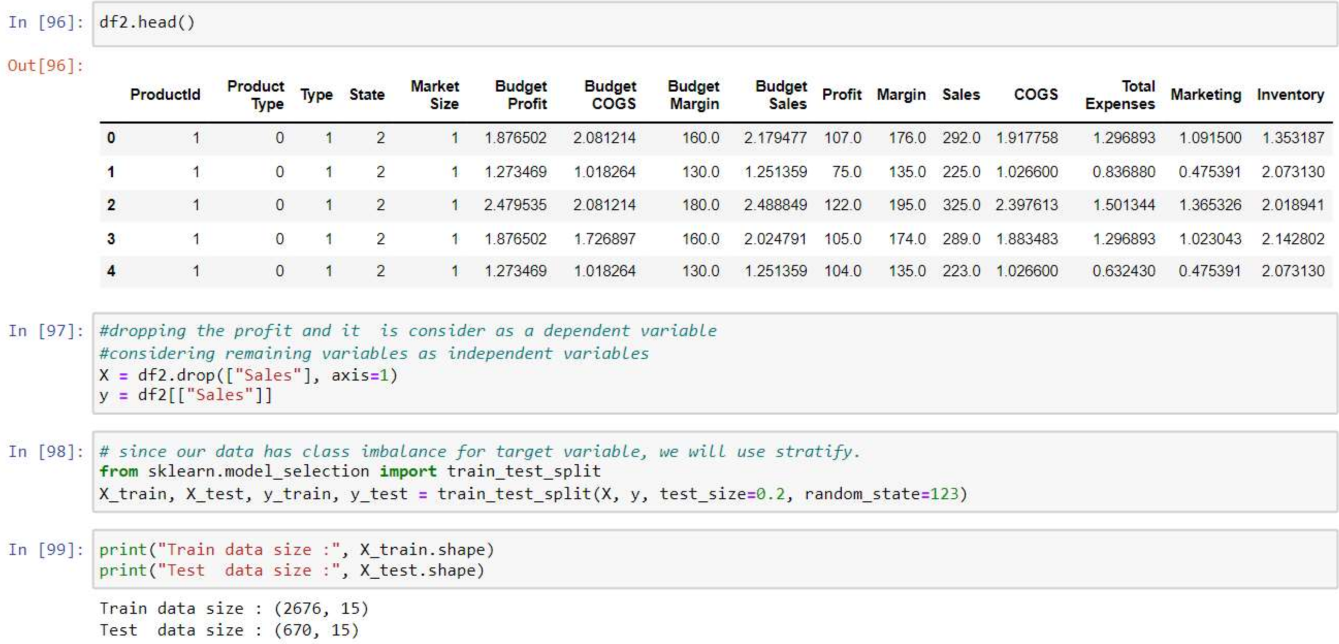


```
In [91]: #IDentifying the outliners and will deals with them in the model 2
         sns.boxplot(data=df2, orient="h", palette="Set2")

Out[91]: <AxesSubplot:>
```



Both productid and Budget sales dont have outliers.Whereas all other columns have outliers.

D. Regression:



```
In [96]: df2.head()
```

Out[96]:

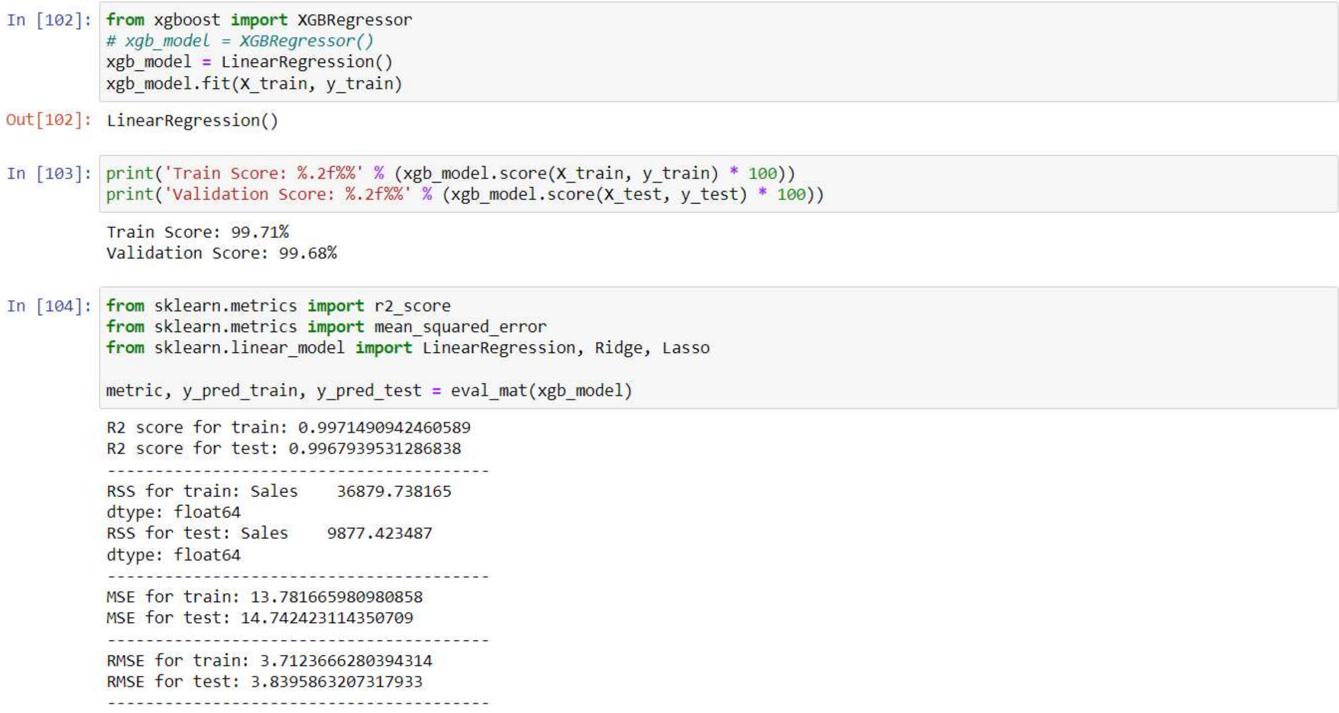
	ProductId	Product Type	Type	State	Market Size	Budget Profit	Budget COGS	Budget Margin	Budget Sales	Profit	Margin	Sales	COGS	Total Expenses	Marketing	Inventory
0	1	0	1	2	1	1.876502	2.081214	160.0	2.179477	107.0	176.0	292.0	1.917758	1.296893	1.091500	1.353187
1	1	0	1	2	1	1.273469	1.018264	130.0	1.251359	75.0	135.0	225.0	1.026600	0.836880	0.475391	2.073130
2	1	0	1	2	1	2.479535	2.081214	180.0	2.488849	122.0	195.0	325.0	2.397613	1.501344	1.365326	2.018941
3	1	0	1	2	1	1.876502	1.726897	160.0	2.024791	105.0	174.0	289.0	1.883483	1.296893	1.023043	2.142802
4	1	0	1	2	1	1.273469	1.018264	130.0	1.251359	104.0	135.0	223.0	1.026600	0.632430	0.475391	2.073130

```
In [97]: #dropping the profit and it  is consider as a dependent variable
         #considering remaining variables as independent variables
         X = df2.drop(["Sales"], axis=1)
         y = df2[["Sales"]]
```

```
In [98]: # since our data has class imbalance for target variable, we will use stratify.
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=123)
```

```
In [99]: print("Train data size :", X_train.shape)
         print("Test  data size :", X_test.shape)

         Train data size : (2676, 15)
         Test  data size : (670, 15)
```

In regression analysis, we considered the target variable as sales and remaining variables as independent variables.Classified our data into train and test data for further prediction.



```
In [102]: from xgboost import XGBRegressor
          # xgb_model = XGBRegressor()
          xgb_model = LinearRegression()
          xgb_model.fit(X_train, y_train)

Out[102]: LinearRegression()
```

```
In [103]: print('Train Score: %.2f%%' % (xgb_model.score(X_train, y_train) * 100))
          print('Validation Score: %.2f%%' % (xgb_model.score(X_test, y_test) * 100))

          Train Score: 99.71%
          Validation Score: 99.68%
```

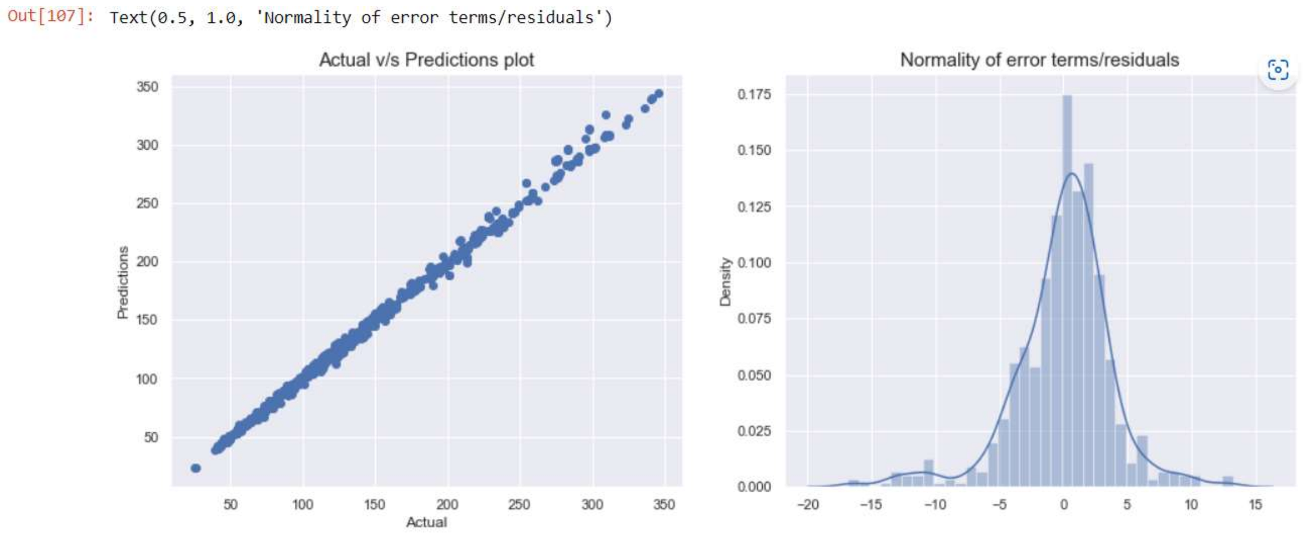
```
In [104]: from sklearn.metrics import r2_score
          from sklearn.metrics import mean_squared_error
          from sklearn.linear_model import LinearRegression, Ridge, Lasso

          metric, y_pred_train, y_pred_test = eval_mat(xgb_model)

          R2 score for train: 0.9971490942460589
          R2 score for test: 0.9967939531286838
          ----------------------------------------
          RSS for train: Sales    36879.738165
          dtype: float64
          RSS for test: Sales    9877.423487
          dtype: float64
          ----------------------------------------
          MSE for train: 13.781665980980858
          MSE for test: 14.742423114350709
          ----------------------------------------
          RMSE for train: 3.7123666280394314
          RMSE for test: 3.8395863207317933
          ----------------------------------------
```

R-Squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. In the linear regression, the model generalized or fitted in the 99.71% training data.

The **Residual Sum of Squares** (RSS) measures the level of variance in the error term, or residuals, of a regression model. The smaller the residual sum of squares, the better your model fits your data; the greater the residual sum of squares, the poorer your model fits your data. From the results higher RSS. So the model is poor fit to the data.

The **Mean Squared Error** measures how close a regression line is to a set of data points. It is a risk function corresponding to the expected value of the squared error loss. Mean square error is calculated by taking the average, specifically the mean, of errors squared from data as it relates to a function. From the results, it has the lower MSE value it indicates that data points are dispersed closely around its mean. It reflects the centralized distribution of our data values.

The **RMSE** is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values. Whereas R-squared is a relative measure of fit, RMSE is an absolute measure of fit. As the square root of a variance, RMSE can be interpreted as the standard deviation of the unexplained variance. From the results, RMSE has the low values, So, the model is better fit and it's good prediction for the response variable.



```
Out[107]: Text(0.5, 1.0, 'Normality of error terms/residuals')
```

From the above plots,

1. Actual v/s Predictions plot, it shows more accuracy with less deviation. Which indicates the model mostly defines the relationship between dependent and independent variables.
2. Normality of error terms/residuals, here the distribution line clearly shows the data is normally distributed.

E. Variable Selection Methods:

1. Filter Methods:

Using Filter methods, we took Sales as Dependent variables and the remaining variables as independent variables.

From the results, The predictors explain about 99% of the variation in our response variable

1. Standard Errors assume that the covariance matrix of the errors is correctly specified.
2. The smallest eigenvalue is 1.43e-28. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

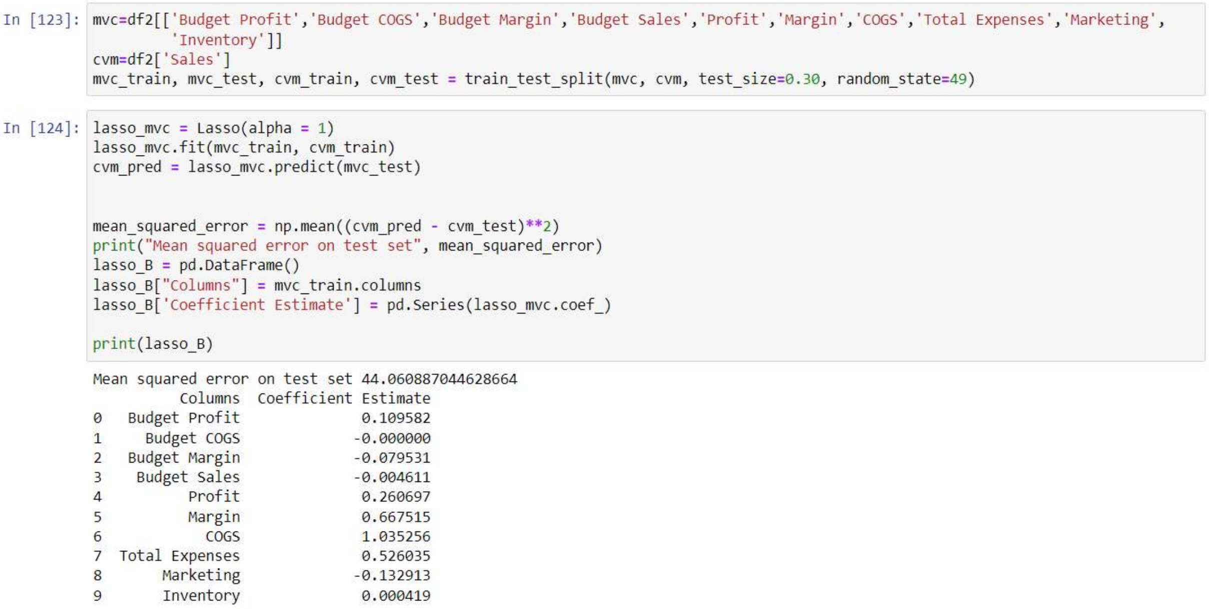
2. Wrapper Methods:

Using Wrapper Methods, here also we taken as the sales as target variable and the remaining variables as independent variables

From the results, the wrapper method also has the 99.8% variation by predictors to our response variable.

1. Standard Errors assume that the covariance matrix of the errors is correctly specified.
2. The smallest eigenvalue is 2.99e-24. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

3. Embedded Methods:



```
In [123]: mvc=df2[['Budget Profit','Budget COGS','Budget Margin','Budget Sales','Profit','Margin','COGS','Total Expenses','Marketing',
                   'Inventory']]
          cvm=df2['Sales']
          mvc_train, mvc_test, cvm_train, cvm_test = train_test_split(mvc, cvm, test_size=0.30, random_state=49)
```

```
In [124]: lasso_mvc = Lasso(alpha = 1)
          lasso_mvc.fit(mvc_train, cvm_train)
          cvm_pred = lasso_mvc.predict(mvc_test)


          mean_squared_error = np.mean((cvm_pred - cvm_test)**2)
          print("Mean squared error on test set", mean_squared_error)
          lasso_B = pd.DataFrame()
          lasso_B["Columns"] = mvc_train.columns
          lasso_B['Coefficient Estimate'] = pd.Series(lasso_mvc.coef_)

          print(lasso_B)
```

```
Mean squared error on test set 44.060887044628664
          Columns  Coefficient Estimate
0   Budget Profit              0.109582
1     Budget COGS             -0.000000
2   Budget Margin             -0.079531
3    Budget Sales             -0.004611
4          Profit              0.260697
5          Margin              0.667515
6            COGS              1.035256
7  Total Expenses              0.526035
8       Marketing             -0.132913
9       Inventory              0.000419
```

Using Embedded methods, the variables are the same as the previous methods and from the results we see the coefficient estimates for the columns.

Here the Budget COGS, Budget Margin, Budget Sales and Marketing has the negative coefficient estimates and the remaining columns have the positive Coefficient estimates.

F. CLASSIFICATION TYPES:

The various estimates of accuracy are listed below, and the performance of the corresponding classifiers in terms of Accuracy, Precision, F-measure, Recall, and Error Rate values is listed in the table below. The table below shows various performance values for all classification algorithms computed using various metrics. From the table below, it has been determined that **K - Nearest Neighbors** provides the greatest accuracy. In contrast to other classifiers, the **K-Nearest Neighbors** machine learning classifier will therefore accurately forecast the greatest sales in the Market Size. since it provides higher accuracy when compared to alternative classification algorithms, with an Associate in measuring accuracy of 83.32% and it has the 0.84 precision, Recall is 0.89 with F-Measure is 0.86 and here the model failure rate is 16.66 to low compared to remaining classification algorithms.

Classifications Algorithms	***Accuracy (%)***	***Precision***	***Recall***	***F-Measure***	***Error Rate (%)***
Logistic Regression	72.54	0.74	0.85	0.79	27.46
Naive Bayes	67.37	0.68	0.89	0.77	32.63
K - Nearest Neighbors	**83.32**	**0.84**	**0.89**	**0.86**	**16.66**
SVM	73.86	0.73	0.90	0.80	26.14

CONCLUSION:

As was shown in the beginning, major market stores tend to have higher profits. The ceiling for profits is also much higher for major market stores.

Major markets are often able to capitalize on factors that require spending money in a way that small markets are not. These stores are able to increase their COGS and total expenses in selling goods because they have the resources and/or demand to make it worthwhile.

Companies with more marketing gain more profits.

While the potential for profits in major market stores is great, the potential for big losses is also increased. There is only a small group of small market data points that had severely negative profits.

Considering the profits gained by stores we can choose lemons, caffe mint instead of Mint, Caffe Latte and Regular Espresso.

We should always choose Central and West Markets to shop since they have more sales.

Which part of the US has more sales?

Central and West Markets have more sales,whereas East and South have comparatively low sales.

Which state has more sales but gains less profits and vice versa?

Texas has more sales but less profit compared to Colorado which gains more profits with less sales than texas.

Which product gives more sales?

Herbal tea and tea gives more profit in US Sales Market

Effect of marketing and sales on profits?

Sales has a higher impact on gaining profits than what marketing does.