Importing the libraries

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

```
df =
pd.read_csv('https://github.com/YBI-Foundation/Dataset/raw/main/Big
%20Sales%20Data.csv')
```

Get Information of Dataframe

```
df.head()
```

```
  Item_Identifier  Item_Weight Item_Fat_Content  Item_Visibility  \
0           FDT36         12.3          Low Fat         0.111448   
1           FDT36         12.3          Low Fat         0.111904   
2           FDT36         12.3               LF         0.111728   
3           FDT36         12.3          Low Fat         0.000000   
4           FDP12          9.8          Regular         0.045523   

      Item_Type  Item_MRP Outlet_Identifier  Outlet_Establishment_Year
\
0  Baking Goods   33.4874            OUT049                       1999

1  Baking Goods   33.9874            OUT017                       2007

2  Baking Goods   33.9874            OUT018                       2009

3  Baking Goods   34.3874            OUT019                       1985

4  Baking Goods   35.0874            OUT017                       2007


  Outlet_Size Outlet_Location_Type        Outlet_Type
Item_Outlet_Sales
0      Medium               Tier 1  Supermarket Type1
436.608721
1      Medium               Tier 2  Supermarket Type1
443.127721
2      Medium               Tier 3  Supermarket Type2
564.598400
3       Small               Tier 1      Grocery Store
1719.370000
```

```
4      Medium               Tier 2  Supermarket Type1
352.874000
```

```
df.info()   #gives column name, count,  not null category, D-type(data
type)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            14204 non-null  object 
 1   Item_Weight                11815 non-null  float64
 2   Item_Fat_Content           14204 non-null  object 
 3   Item_Visibility            14204 non-null  float64
 4   Item_Type                  14204 non-null  object 
 5   Item_MRP                   14204 non-null  float64
 6   Outlet_Identifier          14204 non-null  object 
 7   Outlet_Establishment_Year  14204 non-null  int64  
 8   Outlet_Size                14204 non-null  object 
 9   Outlet_Location_Type       14204 non-null  object 
 10  Outlet_Type                14204 non-null  object 
 11  Item_Outlet_Sales          14204 non-null  float64
dtypes: float64(4), int64(1), object(7)
memory usage: 1.3+ MB
```

```
df.describe()   #gives the linear relation of each column with another
column
```

```
        Item_Weight  Item_Visibility      Item_MRP
Outlet_Establishment_Year  \
count  11815.000000     14204.000000  14204.000000
14204.000000
mean      12.788355         0.065953    141.004977
1997.830681
std        4.654126         0.051459     62.086938
8.371664
min        4.555000         0.000000     31.290000
1985.000000
25%        8.710000         0.027036     94.012000
1987.000000
50%       12.500000         0.054021    142.247000
1999.000000
75%       16.750000         0.094037    185.855600
2004.000000
max       30.000000         0.328391    266.888400
2009.000000

       Item_Outlet_Sales
count       14204.000000
mean         2185.836320
```

```
std          1827.479550
min            33.290000
25%           922.135101
50%          1768.287680
75%          2988.110400
max         31224.726950
```

```
df.isnull().sum() #(df.isna().sum() gives same result)
#gives the sum of all null values columns-wise
```

```
Item_Identifier                 0
Item_Weight                  2389
Item_Fat_Content                0
Item_Visibility                 0
Item_Type                       0
Item_MRP                        0
Outlet_Identifier               0
Outlet_Establishment_Year       0
Outlet_Size                     0
Outlet_Location_Type            0
Outlet_Type                     0
Item_Outlet_Sales               0
dtype: int64
```

```
df.nunique()   #gives total no. of unique entries
```

```
Item_Identifier               1559
Item_Weight                    416
Item_Fat_Content                 5
Item_Visibility              13006
Item_Type                       16
Item_MRP                      8052
Outlet_Identifier               10
Outlet_Establishment_Year        9
Outlet_Size                      3
Outlet_Location_Type             3
Outlet_Type                      4
Item_Outlet_Sales             9144
dtype: int64
```

```
df.count()  #gives total no. of entries in columns it
```

```
Item_Identifier              14204
Item_Weight                  11815
Item_Fat_Content             14204
Item_Visibility              14204
Item_Type                    14204
Item_MRP                     14204
Outlet_Identifier            14204
Outlet_Establishment_Year    14204
Outlet_Size                  14204
```

```
Outlet_Location_Type         14204
Outlet_Type                  14204
Item_Outlet_Sales            14204
dtype: int64
```

```
df.columns  #give column names in the dataframe
```

```
Index(['Item_Identifier', 'Item_Weight', 'Item_Fat_Content',
'Item_Visibility',
       'Item_Type', 'Item_MRP', 'Outlet_Identifier',
       'Outlet_Establishment_Year', 'Outlet_Size',
'Outlet_Location_Type',
       'Outlet_Type', 'Item_Outlet_Sales'],
      dtype='object')
```

```
df.shape
```

```
(14204, 12)
```

```
df.dtypes
```

```
Item_Identifier               object
Item_Weight                  float64
Item_Fat_Content              object
Item_Visibility              float64
Item_Type                     object
Item_MRP                     float64
Outlet_Identifier             object
Outlet_Establishment_Year      int64
Outlet_Size                   object
Outlet_Location_Type          object
Outlet_Type                   object
Item_Outlet_Sales            float64
dtype: object
```

Taking care of missing data

```
df['Item_Weight'].fillna(df.groupby(['Item_Type'])
['Item_Weight'].transform('mean'),inplace=True)
```

```
df.info()
```

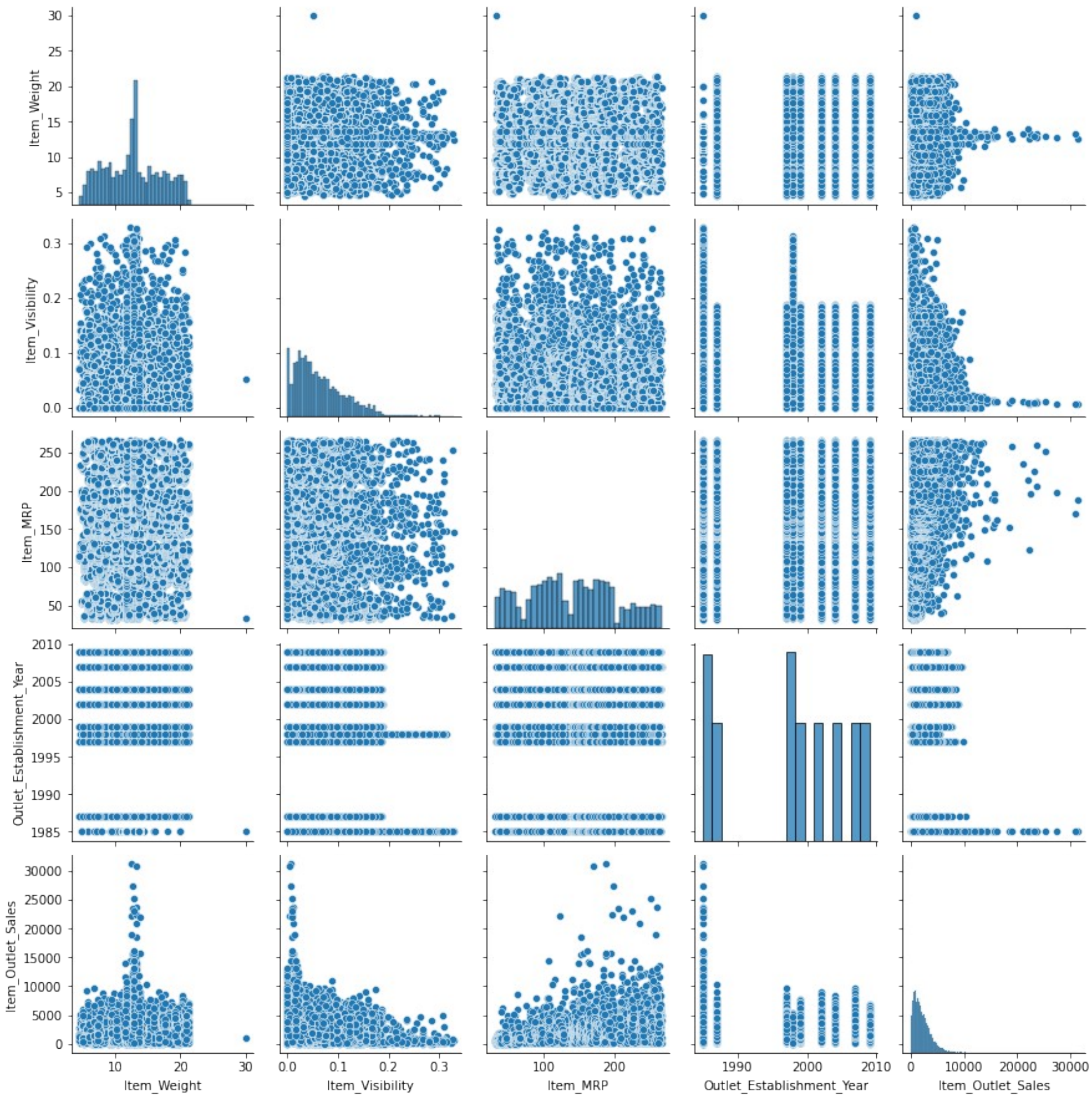
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            14204 non-null  object 
 1   Item_Weight                14204 non-null  float64
 2   Item_Fat_Content           14204 non-null  object 
 3   Item_Visibility            14204 non-null  float64
 4   Item_Type                  14204 non-null  object 
 5   Item_MRP                   14204 non-null  float64
 6   Outlet_Identifier          14204 non-null  object 
 7   Outlet_Establishment_Year  14204 non-null  int64  
 8   Outlet_Size                14204 non-null  object 
 9   Outlet_Location_Type       14204 non-null  object 
 10  Outlet_Type                14204 non-null  object 
 11  Item_Outlet_Sales          14204 non-null  float64
dtypes: float64(4), int64(1), object(7)
memory usage: 1.3+ MB
```

```
df.describe()
```

```
        Item_Weight  Item_Visibility      Item_MRP 
Outlet_Establishment_Year  \
count  14204.000000     14204.000000  14204.000000 
14204.000000   
mean      12.790642         0.065953    141.004977 
1997.830681   
std        4.251186         0.051459     62.086938 
8.371664   
min        4.555000         0.000000     31.290000 
1985.000000   
25%        9.300000         0.027036     94.012000 
1987.000000   
50%       12.800000         0.054021    142.247000 
1999.000000   
75%       16.000000         0.094037    185.855600 
2004.000000   
max       30.000000         0.328391    266.888400 
2009.000000   

       Item_Outlet_Sales  
count       14204.000000  
mean         2185.836320  
std          1827.479550  
min            33.290000  
25%           922.135101  
50%          1768.287680  
75%          2988.110400  
max         31224.726950  
```

```
import seaborn as sns
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f853f00bcd0>
```



et Categories and Counts of Categorical Variables

```
df[['Item_Identifier']].value_counts()
```

```
Item_Identifier
FDQ08              10
FDO24              10
FDQ19              10
FDQ28              10
FDQ31              10
                   ..
FDM52               7
FDM50               7
FDL50               7
FDM10               7
```

```
FDR51               7
Length: 1559, dtype: int64
```

```
df[['Item_Fat_Content']].value_counts()
```

```
Item_Fat_Content
Low Fat             8485
Regular             4824
LF                   522
reg                  195
low fat              178
dtype: int64
```

```
df.replace({'Item_Fat_Content':{'LF':'Low Fat','reg':'Regular','low
fat':'Low Fat'}},inplace=True)
```

```
df['Item_Fat_Content'].value_counts()
```

```
Low Fat    9185
Regular    5019
Name: Item_Fat_Content, dtype: int64
```

```
df.replace({'Item_Fat_Content':{'Low Fat':0,'Regular':1}},
inplace=True)
```

```
df[['Item_Type']].value_counts()
```

```
Item_Type
Fruits and Vegetables    2013
Snack Foods              1989
Household                1548
Frozen Foods             1426
Dairy                    1136
Baking Goods             1086
Canned                   1084
Health and Hygiene        858
Meat                      736
Soft Drinks               726
Breads                    416
Hard Drinks               362
Others                    280
Starchy Foods             269
Breakfast                 186
Seafood                    89
dtype: int64
```

```
df.replace({'Item_Type':{'Fruits and Vegetables':0, 'Snack
Foods':0,'Household':1,
                         'Frozen Foods':0,'Diary':0,'Baking
Goods':0,'Canned':0, 'Health and Hygiene':1,
                         'Heat':0, 'Soft Drinks':0,'Breads':0,'Hard
Drinks':0,'Others':2,'Starchy Foods':0, 'Breakfast':0,
'Seafood':0}},inplace=True)
```

```
df[['Item_Type']].value_counts()
```

```
Item_Type
0            9646
1            2406
Dairy        1136
Meat          736
2             280
dtype: int64
```

```
df[['Outlet_Identifier']].value_counts()
```

```
Outlet_Identifier
OUT027               1559
OUT013               1553
OUT035               1550
OUT046               1550
OUT049               1550
OUT045               1548
OUT018               1546
OUT017               1543
OUT010                925
OUT019                880
dtype: int64
```

```
df.replace({'Outlet_Identifier':{'OUT027':0, 'OUT013':1,'OUT049':2,
                         'OUT046':3,'OUT035':4,'OUT045':5,'OUT018':6,
'OUT017':7,
                         'OUT010':8, 'OUT019':9}},inplace=True)
```

```
df[['Outlet_Size']].value_counts()
```

```
Outlet_Size
Medium         7122
Small          5529
High           1553
dtype: int64
```

```
df.replace({'Outlet_Size': {'Small':0, 'Medium':1, 'High': 2}},
inplace=True)
```

```
df[['Outlet_Location_Type']].value_counts()
```

```
Outlet_Location_Type
Tier 3                  5583
Tier 2                  4641
Tier 1                  3980
dtype: int64
```

```
df.replace({'Outlet_Location_Type':{'Tier 1':0,'Tier 2':1, 'Tier
3':2}},inplace=True)
```

```
df[['Outlet_Type']].value_counts()
```

```
Outlet_Type
Supermarket Type1    9294
Grocery Store        1805
Supermarket Type3    1559
Supermarket Type2    1546
dtype: int64
```

```
df.replace({'Outlet_Type':{'Grocery Store':0, 'Supermarket Type1':1,
'Supermarket Type2':2,'Supermarket Type3':3}},inplace=True)
```

```
df.replace({'Item_Type':{'Dairy':3,'Meat':4}},inplace=True)
```

```
df.head()
```

```
  Item_Identifier  Item_Weight  Item_Fat_Content  Item_Visibility
Item_Type  \
0           FDT36         12.3                 0         0.111448
0
1           FDT36         12.3                 0         0.111904
0
2           FDT36         12.3                 0         0.111728
0
3           FDT36         12.3                 0         0.000000
0
4           FDP12          9.8                 1         0.045523
0

   Item_MRP  Outlet_Identifier  Outlet_Establishment_Year  Outlet_Size
\
0   33.4874                  2                       1999            1

1   33.9874                  7                       2007            1

2   33.9874                  6                       2009            1

3   34.3874                  9                       1985            0

4   35.0874                  7                       2007            1


   Outlet_Location_Type  Outlet_Type  Item_Outlet_Sales
0                     0            1         436.608721
1                     1            1         443.127721
2                     2            2         564.598400
3                     0            0        1719.370000
4                     1            1         352.874000
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14204 entries, 0 to 14203
Data columns (total 12 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Item_Identifier            14204 non-null  object 
 1   Item_Weight                14204 non-null  float64
 2   Item_Fat_Content           14204 non-null  int64  
 3   Item_Visibility            14204 non-null  float64
 4   Item_Type                  14204 non-null  int64  
 5   Item_MRP                   14204 non-null  float64
 6   Outlet_Identifier          14204 non-null  int64  
 7   Outlet_Establishment_Year  14204 non-null  int64  
 8   Outlet_Size                14204 non-null  int64  
 9   Outlet_Location_Type       14204 non-null  int64  
 10  Outlet_Type                14204 non-null  int64  
 11  Item_Outlet_Sales          14204 non-null  float64
dtypes: float64(4), int64(7), object(1)
memory usage: 1.3+ MB
```

#Define y

```
y=df['Item_Outlet_Sales']
```

```
y.shape
```

```
(14204,)
```

```
y
```

```
0         436.608721
1         443.127721
2         564.598400
3        1719.370000
4         352.874000
            ...     
14199    4984.178800
14200    2885.577200
14201    2885.577200
14202    3803.676434
14203    3644.354765
Name: Item_Outlet_Sales, Length: 14204, dtype: float64
```

```
df.columns
```

```
Index(['Item_Identifier', 'Item_Weight', 'Item_Fat_Content',
'Item_Visibility',
       'Item_Type', 'Item_MRP', 'Outlet_Identifier',
       'Outlet_Establishment_Year', 'Outlet_Size',
'Outlet_Location_Type',
       'Outlet_Type', 'Item_Outlet_Sales'],
      dtype='object')
```

```
X=df.drop(['Item_Identifier','Item_Outlet_Sales'],axis=1)
```

```
X
```

```
       Item_Weight  Item_Fat_Content  Item_Visibility  Item_Type
Item_MRP  \
0        12.300000                 0         0.111448          0
33.4874
1        12.300000                 0         0.111904          0
33.9874
2        12.300000                 0         0.111728          0
33.9874
3        12.300000                 0         0.000000          0
34.3874
4         9.800000                 1         0.045523          0
35.0874
...            ...               ...              ...        ...
...
14199    12.800000                 0         0.069606          0
261.9252
14200    12.800000                 0         0.070013          0
262.8252
14201    12.800000                 0         0.069561          0
263.0252
14202    13.659758                 0         0.069282          0
263.5252
14203    12.800000                 0         0.069727          0
263.6252

       Outlet_Identifier  Outlet_Establishment_Year  Outlet_Size  \
0                      2                       1999            1
1                      7                       2007            1
2                      6                       2009            1
3                      9                       1985            0
4                      7                       2007            1
...                  ...                        ...          ...
14199                  4                       2004            0
14200                  7                       2007            1
14201                  1                       1987            2
14202                  0                       1985            1
14203                  2                       1999            1

       Outlet_Location_Type  Outlet_Type
0                         0            1
1                         1            1
2                         2            2
3                         0            0
4                         1            1
```

```
...                      ...          ...
14199                      1            1
14200                      1            1
14201                      2            1
14202                      2            3
14203                      0            1

[14204 rows x 10 columns]
```

#Get X Variables Standardized

```
from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

X_std=
df[['Item_Weight','Item_Visibility','Item_MRP','Outlet_Establishment_Y
ear']]

X_std=sc.fit_transform(X_std)

X_std
```

```
array([[-0.11541705,  0.88413635, -1.73178716,  0.13968068],
       [-0.11541705,  0.89300616, -1.72373366,  1.09531886],
       [-0.11541705,  0.88958331, -1.72373366,  1.3342284 ],
       ...,
       [ 0.00220132,  0.07011952,  1.96538148, -1.29377659],
       [ 0.20444792,  0.06469366,  1.97343499, -1.53268614],
       [ 0.00220132,  0.07334891,  1.97504569,  0.13968068]])
```

```
X[['Item_Weight','Item_Visibility','Item_MRP','Outlet_Establishment_Ye
ar']] = pd.DataFrame(X_std,
columns=[['Item_Weight','Item_Visibility','Item_MRP','Outlet_Establish
ment_Year']])

X
```

```
       Item_Weight  Item_Fat_Content  Item_Visibility  Item_Type
Item_MRP  \
0        -0.115417                 0         0.884136          0 -
1.731787
1        -0.115417                 0         0.893006          0 -
1.723734
2        -0.115417                 0         0.889583          0 -
1.723734
3        -0.115417                 0        -1.281712          0 -
1.717291
4        -0.703509                 1        -0.397031          0 -
1.706016
...            ...               ...              ...        ...
...
```

```
14199     0.002201                 0         0.070990         0
1.947664
14200     0.002201                 0         0.078898         0
1.962160
14201     0.002201                 0         0.070120         0
1.965381
14202     0.204448                 0         0.064694         0
1.973435
14203     0.002201                 0         0.073349         0
1.975046

       Outlet_Identifier  Outlet_Establishment_Year  Outlet_Size  \
0                      2                   0.139681            1
1                      7                   1.095319            1
2                      6                   1.334228            1
3                      9                  -1.532686            0
4                      7                   1.095319            1
...                  ...                        ...          ...
14199                  4                   0.736955            0
14200                  7                   1.095319            1
14201                  1                  -1.293777            2
14202                  0                  -1.532686            1
14203                  2                   0.139681            1

       Outlet_Location_Type  Outlet_Type
0                         0            1
1                         1            1
2                         2            2
3                         0            0
4                         1            1
...                     ...          ...
14199                     1            1
14200                     1            1
14201                     2            1
14202                     2            3
14203                     0            1

[14204 rows x 10 columns]
```

Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.3, random_state = 1)
```

```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

```
((9942, 10), (4262, 10), (9942,), (4262,))
```

```
df['Item_Type'].value_counts()
```

```
0    9646
1    2406
3    1136
4     736
2     280
Name: Item_Type, dtype: int64
```

```
X_train
```

```
       Item_Weight  Item_Fat_Content  Item_Visibility  Item_Type
Item_MRP  \
12210    -1.484495                 0        -0.620225          0
0.099906
12156     1.284242                 0        -0.228613          0
0.032382
1797     -0.105812                 1        -0.230054          0 -
1.329063
10348    -1.787950                 1        -0.420379          4 -
0.597667
2505     -0.105812                 1        -0.186842          0
0.702572
...            ...               ...              ...        ...
...
905       1.166623                 1         0.037543          0
0.746061
5192     -0.824656                 1        -0.324191          0
1.657564
12172     0.872577                 1         1.834909          0
0.046836
235       1.225432                 0        -1.115763          0 -
0.962284
13349     0.519722                 0        -0.328706          0 -
1.217500

       Outlet_Identifier  Outlet_Establishment_Year  Outlet_Size  \
12210                  6                   1.334228            1
12156                  1                  -1.293777            2
1797                   0                  -1.532686            1
10348                  2                   0.139681            1
2505                   9                  -1.532686            0
...                  ...                        ...          ...
905                    7                   1.095319            1
5192                   4                   0.736955            0
12172                  6                   1.334228            1
235                    4                   0.736955            0
13349                  1                  -1.293777            2

       Outlet_Location_Type  Outlet_Type
12210                     2            2
```

```
12156                    2            1
1797                     2            3
10348                    0            1
2505                     0            0
...                    ...          ...
905                      1            1
5192                     1            1
12172                    2            2
235                      1            1
13349                    2            1

[9942 rows x 10 columns]
```

Model

```
from sklearn.ensemble import RandomForestRegressor
rfg=RandomForestRegressor()
rfg.fit(X_train,y_train)

RandomForestRegressor()
```

Model Prediction

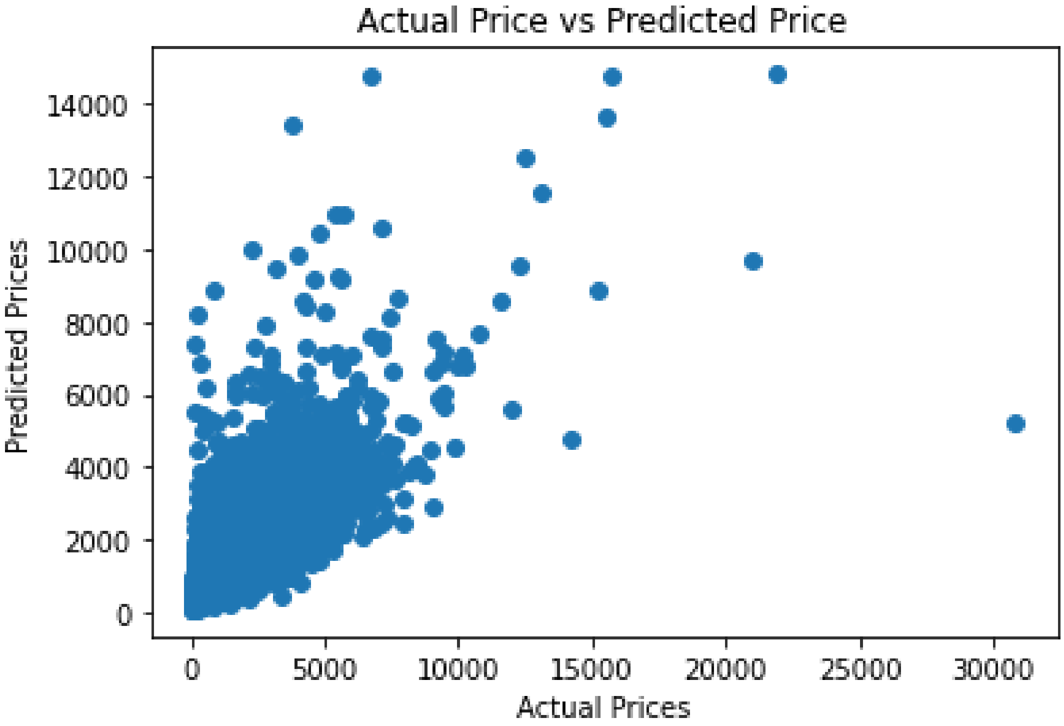
```
y_pred=rfg.predict(X_test)

y_pred.shape

(4262,)

y_pred

array([2946.01017703, 2029.51391296, 1134.53111942, ...,
2332.65014151,
       2471.72123057, 1382.6964982 ])
```

Model Evaluation

```
from sklearn.metrics import mean_squared_error, mean_absolute_error,
mean_absolute_percentage_error, r2_score

mean_squared_error(y_test,y_pred)

1636851.7710929627

mean_absolute_percentage_error(y_test,y_pred)

0.7260326369580009

r2_score(y_test,y_pred)

0.48070477430390246
```

#Get Visualization of Actual Vs Predicted Results

```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual Price vs Predicted Price')
plt.show()
```


Actual Price vs Predicted Price
Predicted Prices
Actual Prices