Karen Alvarado

Capstone Report

GitHub

In the below link you will be able to find all related data and information about this project:
https://github.com/KarenAlvarado1/Module-5-Task-4-

Python and all libraries needed to solve the problem

These are the libraries used for the project at hand:

```
#imports
#numpy, pandas, scipy, math, matplotlib
import numpy as np
import pandas as pd
import scipy
from math import sqrt
import matplotlib.pyplot as plt
#estimators
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.svm import SVR
from sklearn import linear_model
#model metrics
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.model_selection import cross_val_score
#cross validation
from sklearn.cross_validation import train_test_split
#For plots
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

http://localhost:8888/notebooks/BigData/Module%205/Task%204/Module%205%20-%20Task%204%20-%20Karen.ipynb

Exploratory Data Analysis

The data used for this project can be found in the link below. We feel comfortable using it since it is from a reliable source and the challenges found were according to the asks of this final project.

https://www.kaggle.com/mehdidag/black-friday

Some important clarifications to make for the better understanding of the data:

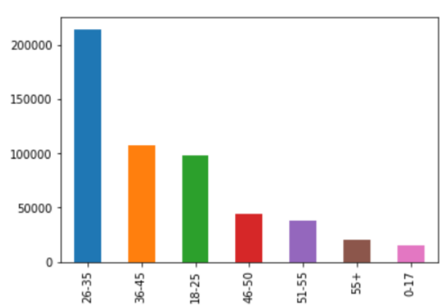
- Gender: 0 = M, 1= F
- Marital Status: 0 = single, 1 = Married
- Age: 1 = 0 to 17, 2= 18 to 25, 3= 26 to 35, 4= 36 to 45, 5= 46 to 50, 6= 51 to 55, 7= 55+
- Stay in Current City: 1= 1Yr, 2= 2Yrs, 3= 3Yrs, 4= 4+ years

The whole data set consists on 537,577 rows and 12 columns.
The most important variables present the following characteristics:

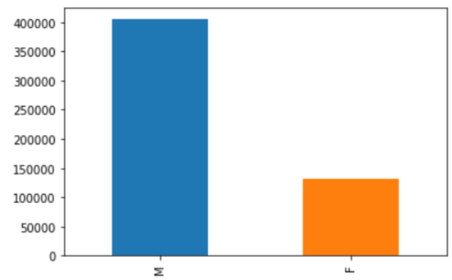
- Age:
  - 0-17: 14707
  - 18-25: 97634
  - 26-35: 214690
  - 36-45: 107499
  - 46-50: 44526
  - 51-55: 37618
  - 55+: 20903
- Gender
  - M: 405380
  - F: 132197
- Marital status:
  - 0: 317817
  - 1: 219760

We can visualize this data better with the following graphs:

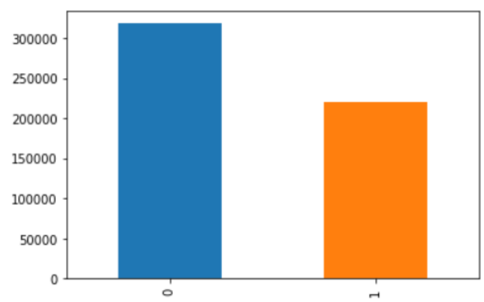
```
In [17]: Data.Age.value_counts().plot(kind='bar')
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x24ce185ecf8>
```



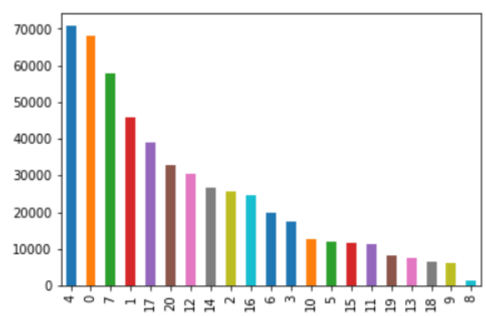
```
In [18]: Data.Gender.value_counts().plot(kind='bar')
Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x24ce48b8978>
```



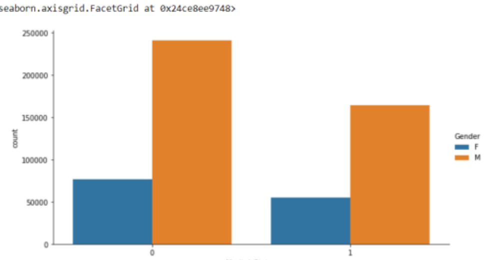

```
In [19]: Data.Marital_Status.value_counts().plot(kind='bar')
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x24ce490ccf8>
```



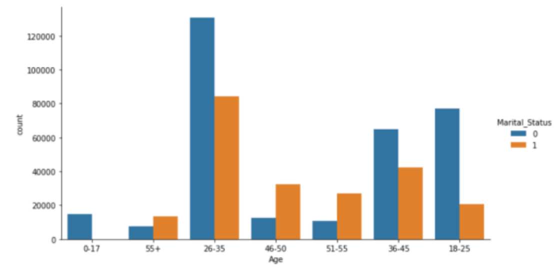

```
In [20]: Data.Occupation.value_counts().plot(kind='bar')
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x24ce4992240>
```



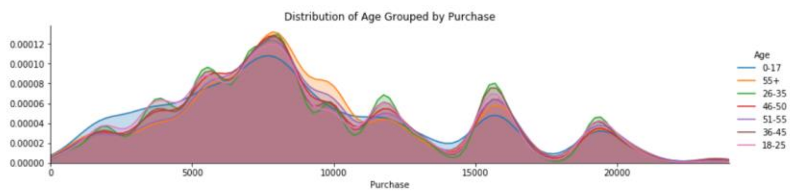

```
In [33]: # Using saeborn to group by Sex and marital status
g = sns.factorplot('Marital_Status', data=Data, hue='Gender', kind='count', aspect=1.75)
g.set_xlabels('Marital_Status')

C:\Users\kalvarado\AppData\Local\Continuum\anaconda3\lib\site-packages\seaborn\categorical.py:3666: UserWarning: The `factorpl
ot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. No
te that the default `kind` in `factorplot` (``'point'``) has changed ``'strip'`` in `catplot`.
  warnings.warn(msg)

Out[33]: <seaborn.axisgrid.FacetGrid at 0x24ce8ee9748>
```



```
In [34]: # Using saeborn to group by MARRIAGE and DEFAULT
         g = sns.factorplot('Age', data=Data, hue='Marital_Status', kind='count', aspect=1.75)
         g.set_xlabels('Age')

C:\Users\kalvarado\AppData\Local\Continuum\anaconda3\lib\site-packages\seaborn\categorical.py:3666: UserWarning: The `factorpl
ot` function has been renamed to `catplot`. The original name will be removed in a future release. Please update your code. No
te that the default `kind` in `factorplot` (`'point'`) has changed `'strip'` in `catplot`.
  warnings.warn(msg)

Out[34]: <seaborn.axisgrid.FacetGrid at 0x24ce8cd0358>
```




```
In [35]: #Grouped by Balance Limit
         fig = sns.FacetGrid(Data, hue='Age', aspect=4)
         fig.map(sns.kdeplot, 'Purchase', shade=True)
         oldest = Data['Purchase'].max()
         fig.set(xlim=(0,oldest))
         fig.set(title='Distribution of Age Grouped by Purchase')
         fig.add_legend()

Out[35]: <seaborn.axisgrid.FacetGrid at 0x24ce8d9f160>
```



In general, we can see that most buyers during Black Friday, at least in this data set, were men and single, between 26 and 35 years of age. This is quite interesting since people tend to perceive that women are more likely to spend money during this holiday; however, here we see that men spend more than 3 times the amount women spend.

Besides this initial contradiction to common thinking, we see that the rest of the data would be pretty logical since men between 26 and 35 years of age are likely in a higher professional position, and in this days more and more men are waiting more to get married; which leaves them with a higher financial independency for a longer amount of time; which would explain why there are so many males purchasing so much during this date.

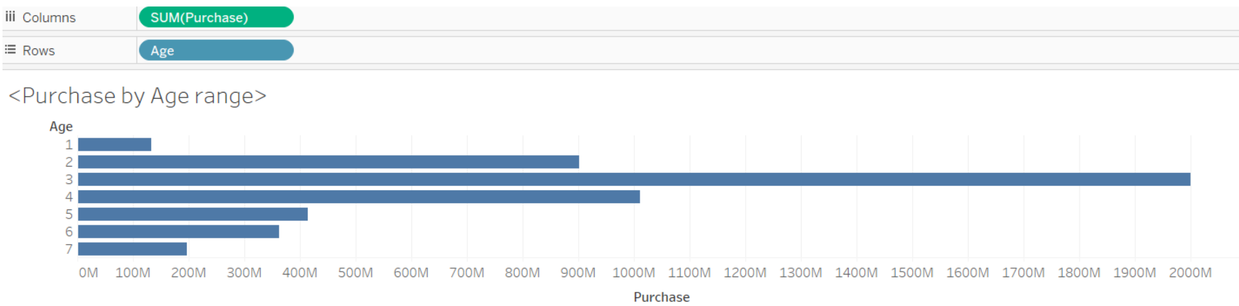
In [10]:
```
#Correlation Matrix
corrMat = Data.corr()
print(corrMat)
```

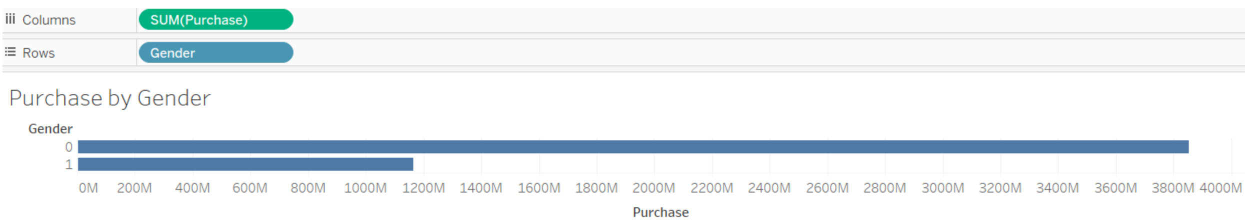
```
                            User_ID  Occupation  City_Category  \
User_ID                    1.000000   -0.023024       0.024107
Occupation                -0.023024    1.000000       0.033781
City_Category              0.024107    0.033781       1.000000
Stay_In_Current_City_Years -0.030655   0.031203       0.019948
Marital_Status             0.018732    0.024691       0.040173
Product_Category_1         0.003687   -0.008114      -0.027444
Product_Category_2         0.003663    0.006792       0.019535
Product_Category_3         0.003938    0.011941       0.037751
Purchase                   0.005389    0.021104       0.068507

                            Stay_In_Current_City_Years  Marital_Status  \
User_ID                                      -0.030655        0.018732
Occupation                                    0.031203        0.024691
City_Category                                 0.019948        0.040173
Stay_In_Current_City_Years                    1.000000       -0.012663
Marital_Status                               -0.012663        1.000000
Product_Category_1                           -0.004182        0.020546
Product_Category_2                            0.001244        0.001146
Product_Category_3                            0.001992       -0.004363
Purchase                                      0.005470        0.000129

                            Product_Category_1  Product_Category_2  \
User_ID                               0.003687            0.003663
Occupation                           -0.008114            0.006792
City_Category                        -0.027444            0.019535
Stay_In_Current_City_Years           -0.004182            0.001244
Marital_Status                        0.020546            0.001146
Product_Category_1                    1.000000           -0.040730
Product_Category_2                   -0.040730            1.000000
Product_Category_3                   -0.389048            0.090284
Purchase                             -0.314125            0.038395
```

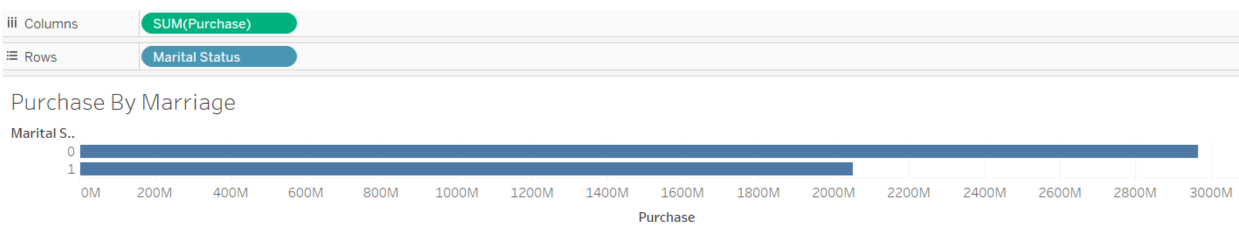
In [11]:
```
#Covariance
covMat = Data.cov()
print(covMat)
```

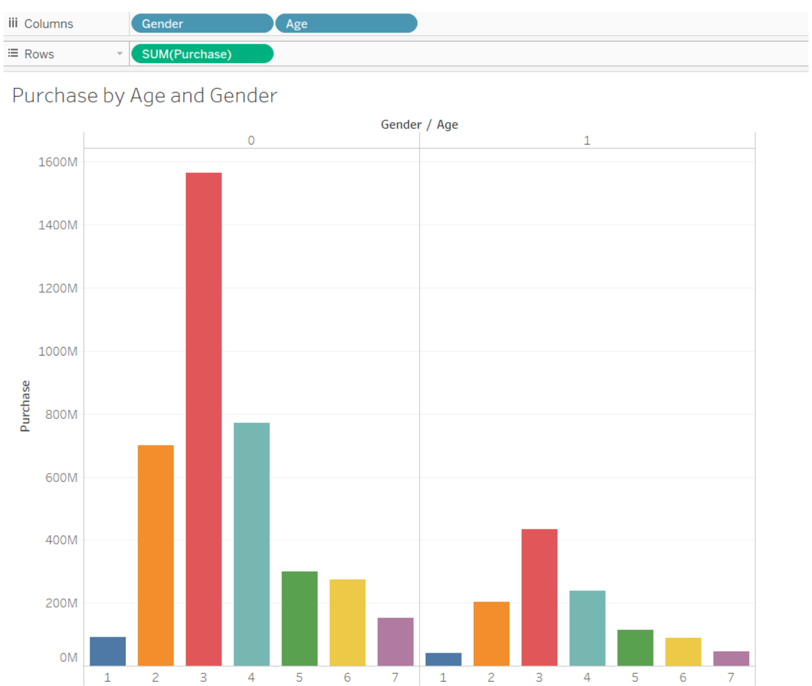
```
                                 User_ID  Occupation  City_Category  \
User_ID                     2.939142e+06 -257.522212      31.394316
Occupation                 -2.575222e+02   42.564139       0.167413
City_Category               3.139432e+01    0.167413       0.577033
Stay_In_Current_City_Years -6.778627e+01    0.262569       0.019545
Marital_Status              1.578743e+01    0.079192       0.015002
Product_Category_1          2.370829e+01   -0.198560      -0.078190
Product_Category_2          3.900920e+01    0.275248       0.092178
Product_Category_3          4.230483e+01    0.488144       0.179689
Purchase                    4.602301e+04  685.823205     259.212384

                            Stay_In_Current_City_Years  Marital_Status  \
User_ID                                     -67.786271       15.787429
Occupation                                    0.262569        0.079192
City_Category                                 0.019545        0.015002
Stay_In_Current_City_Years                    1.663656       -0.008030
Marital_Status                               -0.008030        0.241683
Product_Category_1                           -0.020231        0.037884
Product_Category_2                            0.009968        0.003499
Product_Category_3                            0.016099       -0.013441
Purchase                                     35.140495        0.315931

                            Product_Category_1  Product_Category_2  \
User_ID                              23.708294           39.009204
Occupation                           -0.198560            0.275248
City_Category                        -0.078190            0.092178
Stay_In_Current_City_Years           -0.020231            0.009968
Marital_Status                        0.037884            0.003499
Product_Category_1                   14.067758           -0.948914
Product_Category_2                   -0.948914           38.584204
Product_Category_3                   -9.143311            3.513997
Purchase                          -5868.580224         1187.951501
```

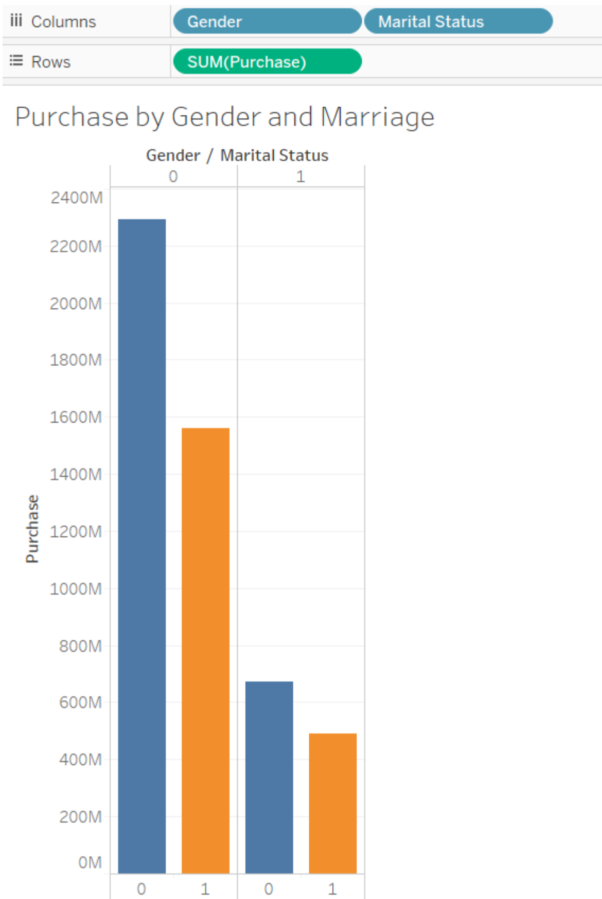
Data Visualization

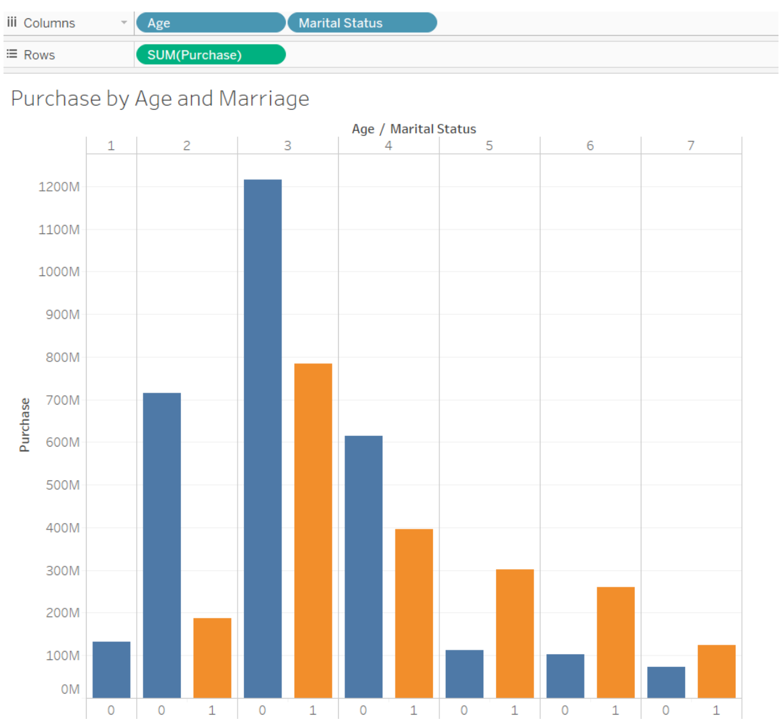









Columns
Gender
Marital Status
Rows
SUM(Purchase)
Purchase by Gender and Marriage
Gender / Marital Status
0
1
Purchase
2400M
2200M
2000M
1800M
1600M
1400M
1200M
1000M
800M
600M
400M
200M
0M
0
1
0
1


Columns
Age
Marital Status
Rows
SUM(Purchase)
Purchase by Age and Marriage
Age / Marital Status
1
2
3
4
5
6
7
Purchase
1200M
1100M
1000M
900M
800M
700M
600M
500M
400M
300M
200M
100M
0M
0
0
1
0
1
0
1
0
1
0
1
0
1

Data collection, pre-processing and feature engineering

https://www.kaggle.com/mehdidag/black-friday

In [7]:
```
#features
features = DataNew.iloc[:,2:14]
print('Summary of feature sample')
features.head()
```

```
Summary of feature sample
```

Out[7]:

	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchas
0	1	1	10	1	2	0	3	0	0	837
1	1	1	10	1	2	0	1	6	14	1520
2	1	1	10	1	2	0	12	0	0	142
3	1	1	10	1	2	0	12	14	0	105
4	0	7	16	3	4	0	8	0	0	796

Data science process: Best practices

As part the data science process I first had to define the process that we were going follow to thoroughly analyze the data found in the Black Friday dataset. I choose the following framework.
Define the goal:

- Why do the stakeholders want to do the project?

This project is important for stake holders as they will be able to visualize a forecast of potential purchases for future years and wit will allow them to estimate a potential amount of dollars they can expect to sell.

- What do they need from it?

They need an analysis to describe potential fixed customers that are likely to buy from them in Black Friday.

- Why is their current solution inadequate?

Currently it seems that there are no solutions for their problem so this will solve at least their initial inquiries.

- What resources do you need?

We need the correct libraries to upload into python, we would also need the description of all the variables in the data set; however, we weren't able to identify all of them, so we will work with the information we have to date.

- How will the result of your project be deployed?

The result of the project will be deployed in marketing campaigns and forecast to stake holders, as of potential gains from this celebration.

Collect and manage data

- What data is available?

We were able to collect a whole data set with more than 500K rows and 12 columns, from which we created 2 other columns from the previous ones that were already there.

- Will it help to solve the problem? Is it enough?

I believe this Will be enough to solve the initial problem, as we are performing a complete analysis to answer the most important questions we can infer from the data.

- Is the data quality good enough?

We had to perform several modifications to the data; however, the quality of it is good enough to perform the required analysis.

Build the model

- Which techniques might I apply to build the model?

In order to build the model, I started by analyzing all the variables, then performed the correlation and covariance matrices to identify important correlations and interdependencies. After defining the features, I define the dependent variable, and partition the data into the training and testing environments and finally determine the types of models I will use. As the dependent variable is continuous, we would need a regression analysis using regression techniques.

- How many techniques should I apply?

In this case I will use 3 regression techniques: Random Forest, Support Vector Regression and Linear Regression.

Evaluate and critique the model

- Is the model accurate enough to meet the stakeholders' needs?

According to the accuracy metrics gathered, we can say that yes, the model is accurate enough.

- Does it perform better than "the obvious guess" and any techniques being used currently?

Yes, as the accuracy is higher than 50% (actually almost 100%), it is safe to say that the technique is way better than the obvious guess.

- Do the results of the model make sense in the context of the real-world problem domain?

Yes, the results make sense as they were compared to the real data and the results showed to be accurate.

Present results and document

- How should stakeholders interpret the model?

The model is quite easy to understand and interpret, the graphic interfaces used both in python and in tableau will allow us to explain the models clearly in a way that is easy for them to derive decisions from it.

- How confident should they be in its predictions?

They should be pretty confident in the predictions as the accuracy is high; however, using a different data set the results may vary as the model may be over fitted.

<u>Deploy and maintain the model</u>

- How is the model to be handed off to "production"?

The model is to be used with the features specified in python, the person who will run the code will only have to input the new data set, all the steps that follow will allow him/her to create the new variables and choose the correct features, so the "production" part should be fairly easy.

- How often, and under which circumstances, should the model be revised?

Whenever there's new data, since this is an annual event, I would expect to have new data every year.

Predictive Modeling and Evaluation (the whole process)

The models chosen were Support Vector Regression, Random Forest and Linear Regression. The whole process can be seen in the python notebook.

```
In [15]: #Models
         modelSVR = SVR()
         modelRF = RandomForestRegressor()
         modelLR = LinearRegression()
```

Model selection

By accuracy we chose to use the RF model.

```
In [18]: modelRF.score(X_train,y_train)
Out[18]: 0.9999999986901742

In [21]: modelLR.score(X_train,y_train)
Out[21]: 1.0
```

Cross validation

In [28]:
```
print(cross_val_score(modelRF, X_train, y_train))
```
```
[ 0.99999997  0.99999998  0.99999998]
```

In [16]:
```
print(cross_val_score(modelSVR, X_train, y_train))
```
```
[-0.06504179 -0.06526591 -0.06333199]
```

In [17]:
```
print(cross_val_score(modelLR, X_train, y_train))
```
```
[ 1.  1.  1.]
```

In [18]:
```
#Model Fitting
modelRF.fit(X_train,y_train)
print(cross_val_score(modelRF, X_train, y_train))
modelRF.score(X_train,y_train)
```
```
[ 0.99999999  0.99999997  0.99999999]
```

Out[18]: 0.99999999891233371