Implementation of Multiple Linear Regression using

Backward Elimination

CODE::

```
Spyder (Python 3.5)
      Edit
            Search
                     Source
                              Run
                                    Debug
                                            Consoles
                                                       Projects
                                                                Tools
                                                                       View
Editor - C:\Users\hp\Desktop\Machine Learning A-Z\Part 2 - Regression\Section 5 - Multiple Linear Regression\Multiple_Linear_Regression_div.py
Data_Preprocessing_div.py
                              Simple_Linear_Regression_div.py
                                                            Multiple_Linear_Regression_div.py
                                                                                           Multiple_linear_regress 4
                                                                                                             •
    7 # Importing the libraries
   8 import numpy as np
   9 import matplotlib.pyplot as plt
   10 import pandas as pd
   11
   12 # Importing the dataset
   13 dataset = pd.read_csv('50_Startups.csv')
   14 X = dataset.iloc[:, :-1].values
   15 y = dataset.iloc[:, 4].values
   16
   17 #handling missing data
   18 from sklearn.preprocessing import Imputer
   19 imputer=Imputer(missing_values="NaN",strategy="mean")
   20 imputer=imputer.fit(X[:,0:3])
   21 X[:,0:3]=imputer.transform(X[:,0:3])
   23 # Encoding categorical data
   24 # Encoding the Independent Variable
   25 from sklearn.preprocessing import LabelEncoder, OneHotEncoder
   26 labelencoder X = LabelEncoder()
   27 X[:, 3] = labelencoder_X.fit_transform(X[:, 3])
   28 onehotencoder = OneHotEncoder(categorical features = [3])
   29 X = onehotencoder.fit_transform(X).toarray()
   30
   31 X=X[:,1:]
   32
   33 # Splitting the dataset into the Training set and Test set
   34 from sklearn.cross validation import train test split
   35 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 45)
```

Spyder (Python 3.5)

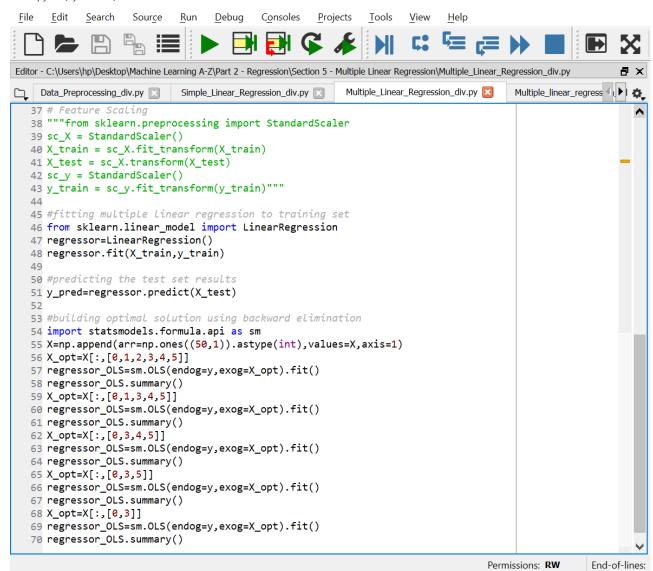




Fig1:: dataset=pd.read_csv('50_Startups.csv')

```
[[165349.2 136897.8 471784.1 'New York']
 [162597.7 151377.59 443898.53 'California']
 [153441.51 101145.55 407934.54 'Florida']
 [144372.41 118671.85 383199.62 'New York']
 [142107.34 91391.77 366168.42 'Florida']
 [131876.9 99814.71 362861.36 'New York']
 [134615.46 147198.87 127716.82 'California']
 [130298.13 145530.06 323876.68 'Florida']
 [120542.52 148718.95 311613.29 'New York']
 [123334.88 108679.17 304981.62 'California']
 [101913.08 110594.11 229160.95 'Florida']
 [100671.96 91790.61 249744.55 'California']
 [93863.75 127320.38 249839.44 'Florida']
 [91992.39 135495.07 252664.93 'California']
 [119943.24 156547.42 256512.92 'Florida']
 [114523.61 122616.84 261776.23 'New York']
 [78013.11 121597.55 264346.06 'California']
 [94657.16 145077.58 282574.31 'New York']
 [91749.16 114175.79 294919.57 'Florida']
 [86419.7 153514.11 0.0 'New York']
 [76253.86 113867.3 298664.47 'California']
 78389.47 153773.43 299737.29 'New York']
 [73994.56 122782.75 303319.26 'Florida']
 [67532.53 105751.03 304768.73 'Florida']
 [77044.01 99281.34 140574.81 'New York']
 [64664.71 139553.16 137962.62 'California']
 [75328.87 144135.98 134050.07 'Florida']
```

Fig2:: x=dataset.iloc[:,0:4].values

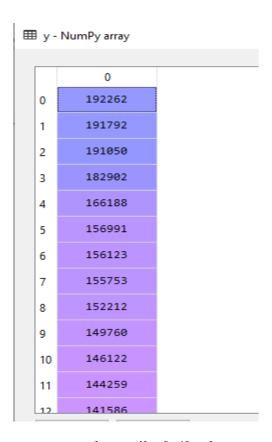


Fig3:: y=dataset.iloc[:,4].values

```
[[165349.2 136897.8 471784.1 2]
 [162597.7 151377.59 443898.53 0]
 [153441.51 101145.55 407934.54 1]
 [144372.41 118671.85 383199.62 2]
 [142107.34 91391.77 366168.42 1]
 [131876.9 99814.71 362861.36 2]
 [134615.46 147198.87 127716.82 0]
 [130298.13 145530.06 323876.68 1]
 [120542.52 148718.95 311613.29 2]
 [123334.88 108679.17 304981.62 0]
 [101913.08 110594.11 229160.95 1]
 [100671.96 91790.61 249744.55 0]
 [93863.75 127320.38 249839.44 1]
 [91992.39 135495.07 252664.93 0]
 [119943.24 156547.42 256512.92 1]
 [114523.61 122616.84 261776.23 2]
 [78013.11 121597.55 264346.06 0]
 [94657.16 145077.58 282574.31 2]
 [91749.16 114175.79 294919.57 1]
 [86419.7 153514.11 0.0 2]
 [76253.86 113867.3 298664.47 0]
 [78389.47 153773.43 299737.29 2]
 [73994.56 122782.75 303319.26 1]
 [67532.53 105751.03 304768.73 1]
 [77044.01 99281.34 140574.81 2]
 [64664.71 139553.16 137962.62 0]
```

Fig4:: x[:,3]=labelencoder.fit_transform(x[:,3])

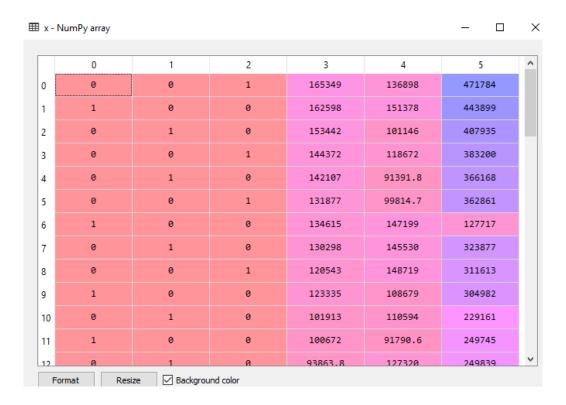


Fig5:: x=onehotencoder.fit_transform(x).toarray()



Fig6:: x=x[:,1:]

Dataset after the removal of dummy variable