Multiple Linear Regression

```
In [1]:  import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         from sklearn import linear_model
```

```
In [2]:  user=pd.read_csv("E:\\Assignment Regression\\User_Data.csv")
         user
```

Out[2]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0
...	...	...	...	...	...
395	15691863	Female	46	41000	1
396	15706071	Male	51	23000	1
397	15654296	Female	50	20000	1
398	15755018	Male	36	33000	0
399	15594041	Female	49	36000	1

400 rows × 5 columns

```
In [19]:  #Allocate the needful Column data
          x = user.iloc[:,:-1]
          y = user.iloc[:,4]
```

In [20]:
```python
#Drop Unnecessary Columm
x=x.drop('Gender',axis='columns') ,
x=x.drop('User ID',axis='columns')
x
```

Out[20]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...	...	...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

In [21]:
```python
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4,
                                                    random_state=1)
```

In [22]:
```python
reg = linear_model.LinearRegression()
reg.fit(x_train, y_train)
```

Out[22]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)

In [29]:
```python
reg.coef_
```

Out[29]: array([2.44572890e-02, 4.78219567e-06])

In [31]:
```python
reg.score(x_test, y_test)
```

Out[31]: 0.435068929354876

In [32]:
```python
reg.predict([[30,40000]])
```

Out[32]: array([0.02985473])

In [33]:
```python
## plotting residual errors in training data
plt.scatter(reg.predict(x_train), reg.predict(x_train) - y_train,
            color = "green", s = 10, label = 'Train data')

## plotting residual errors in test data
plt.scatter(reg.predict(x_test), reg.predict(x_test) - y_test,
            color = "blue", s = 10, label = 'Test data')

## plotting line for zero residual error
plt.hlines(y = 0, xmin = 0, xmax = 50, linewidth = 2)

## plotting legend
plt.legend(loc = 'upper right')

## plot title
plt.title("Residual errors")

## function to show plot
plt.show()
```
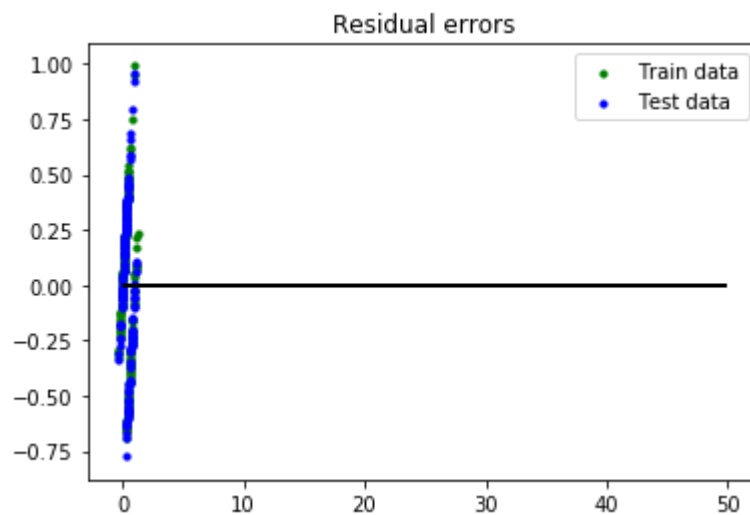

In [ ]:

In [ ]: