Task 2 (Supervised Learning)

Importing Standard Libraries

In [1]:

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

Reading csv files

In [3]:

```
url = 'http://bit.ly/w-data'
data = pd.read_csv(url)
```

In [4]:
```
data
```

Out[4]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [5]:
```
data.head()
```

Out[5]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

In [6]:
```
data.shape
```

Out[6]:
```
(25, 2)
```

Calculating statistical data

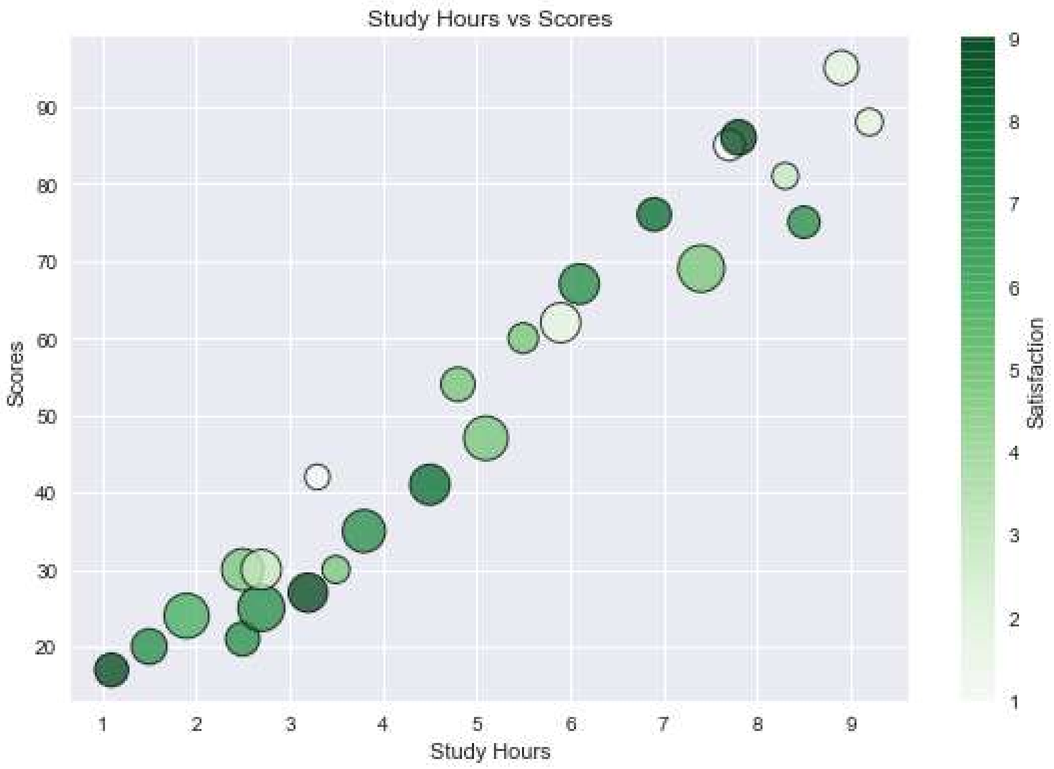
In [7]:
```
data.describe()
```

Out[7]:

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

Plotting a Scatterplot

In [8]:
```
plt.style.use('seaborn')
Hours = [2.5,5.1,3.2,8.5,3.5,1.5,9.2,5.5,8.3,2.7,7.7,5.9,4.5,3.3,1.1,8.9,2.5,1.9,6.1,7.4,2.7,4.8,3.8,6.9,7.8]
Scores = [21,47,27,75,30,20,88,60,81,25,85,62,41,42,17,95,30,24,67,69,30,54,35,76,86]
colors = [7,5,9,7,5,7,2,5,3,7,1,2,8,1,9,2,5,6,7,5,3,5,7,8,9]
sizes = [289,486,381,255,191,315,185,228,174,538,239,394,399,153,273,293,436,501,397,539,401,289,456,278,309]
plt.scatter(Hours, Scores, s=sizes, c=colors, cmap='Greens', edgecolor='Black',linewidth=1, alpha=0.75)
cbar = plt.colorbar()
cbar.set_label('Satisfaction')
plt.title('Study Hours vs Scores')
plt.xlabel('Study Hours')
plt.ylabel('Scores')
plt.tight_layout()
plt.show()
```



cleaning of data

```
In [12]: data.isnull().sum()
```

```
Out[12]: Hours     0
         Scores    0
         dtype: int64
```

```
In [13]: #split the data into explanatory and independent variables
         marks = data.drop("Scores",axis = "columns")
         duration = data.drop("Hours",axis = "columns")
```

```
In [14]: marks.shape
```

```
Out[14]: (25, 1)
```

```
In [15]: duration.shape
```

```
Out[15]: (25, 1)
```

train_test_split

```
In [16]: from sklearn.model_selection import train_test_split
         marks_train,marks_test,duration_train,duration_test=train_test_split(marks,duration,test_size=0.2,random_state=0
```

```
In [17]: marks_train.shape
```

```
Out[17]: (20, 1)
```

```
In [18]: duration_test.shape
```

```
Out[18]: (5, 1)
```

Visualization

```
In [20]: from sklearn.linear_model import LinearRegression
         reg = LinearRegression()
```

```
In [21]: reg.fit(marks_train,duration_train)
```

```
Out[21]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
In [22]: duration_pred = reg.predict(marks_test)
         duration_pred
```

```
Out[22]: array([[16.88414476],
                [33.73226078],
                [75.357018  ],
                [26.79480124],
                [60.49103328]])
```

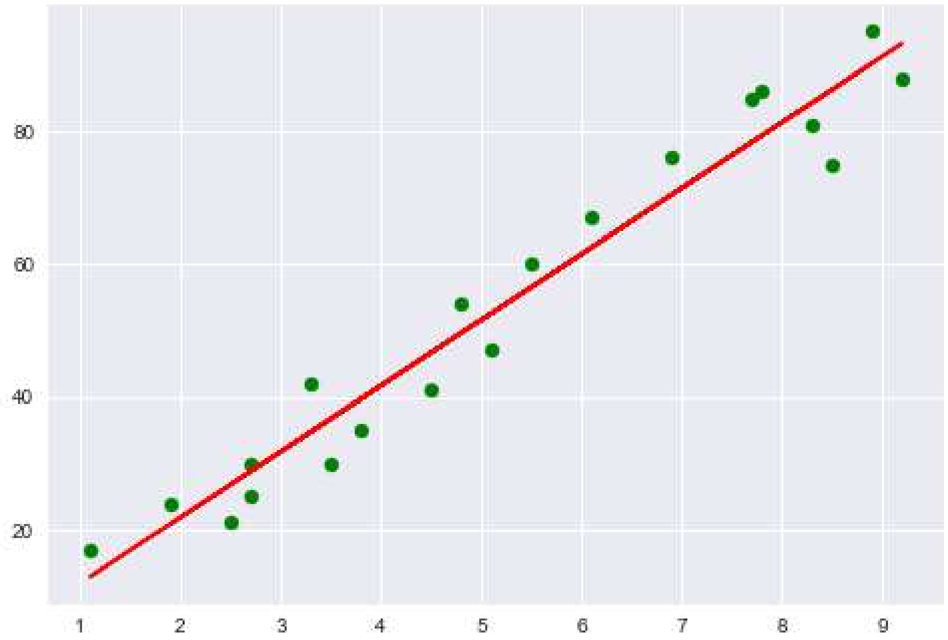
```
In [24]: duration_pred2 = reg.predict(marks_train)
```

In [26]:

```
plt.scatter(marks_train,duration_train,color='green')
plt.plot(marks_train,duration_pred2,color='red')
```

Out[26]:

```
[<matplotlib.lines.Line2D at 0x1dfe5e26108>]
```



In [29]:

```
from sklearn.metrics import mean_squared_error
score = mean_squared_error(duration_pred,duration_test)
print(score)
r_score = np.sqrt(mean_squared_error(duration_pred,duration_test))
print(r_score)
```

```
21.5987693072174
4.6474476121003665
```

In [30]:

```
duration_pred1 = reg.predict([[9.25]])
duration_pred1
```

Out[30]:

```
array([[93.69173249]])
```

In [ ]: