46. Support Vector Machines (SVM) - Regression

Support Vector Regression (SVR) is a regression technique that uses SVM for modellling and predicting continuous outcomes.

- Opposite of SVC
- Here distance between decision b/w support vectors should be minimum

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
dataset = pd.read_csv(r'Data/placement.csv')
dataset.head(3)
```

Out[2]:

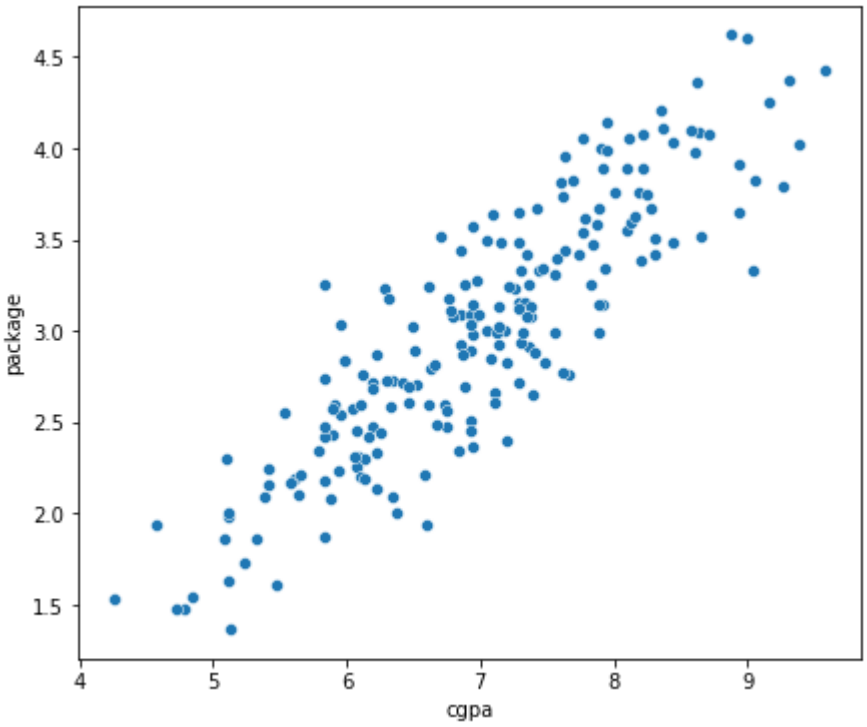
	cgpa	package
0	6.89	3.26
1	5.12	1.98
2	7.82	3.25

In [3]:
```
dataset.isnull().sum()
```

Out[3]:
```
cgpa       0
package    0
dtype: int64
```

Step 1: To check if the data is linearly/non-linearly separable data

In [27]:
```
plt.figure(figsize=(7,6))
sns.scatterplot(x='cgpa', y='package', data=dataset)
plt.show()
```



This graph represents that our data is linearly separable

Step 2: Separate dependent and independent variables

In [16]:
```
x = dataset[['cgpa']]
y =dataset['package']
```

Step 3: Split data into train and test data

In [17]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_st
```

Step 4: Train data through SVR Model

In [19]:
```
from sklearn.svm import SVR
```

In [22]:
```
'''kernel : {'linear', 'poly', 'rbf', 'sigmoid', 'precomputed'} or callable,
     Specifies the kernel type to be used in the algorithm.
     If none is given, 'rbf' will be used. If a callable is given it is
     used to precompute the kernel matrix.'''
sv = SVR(kernel='linear')
sv.fit(x_train, y_train)
```

Out[22]:
```
SVR
SVR(kernel='linear')
```

Step 5: Check accuracy of SVM Model

In [23]:
```
sv.score(x_test, y_test)*100
```

Out[23]: 77.06668029575103

Step 6: Check whether SVM Model is over/under-fit

In [24]:
```
sv.score(x_test, y_test)*100
```

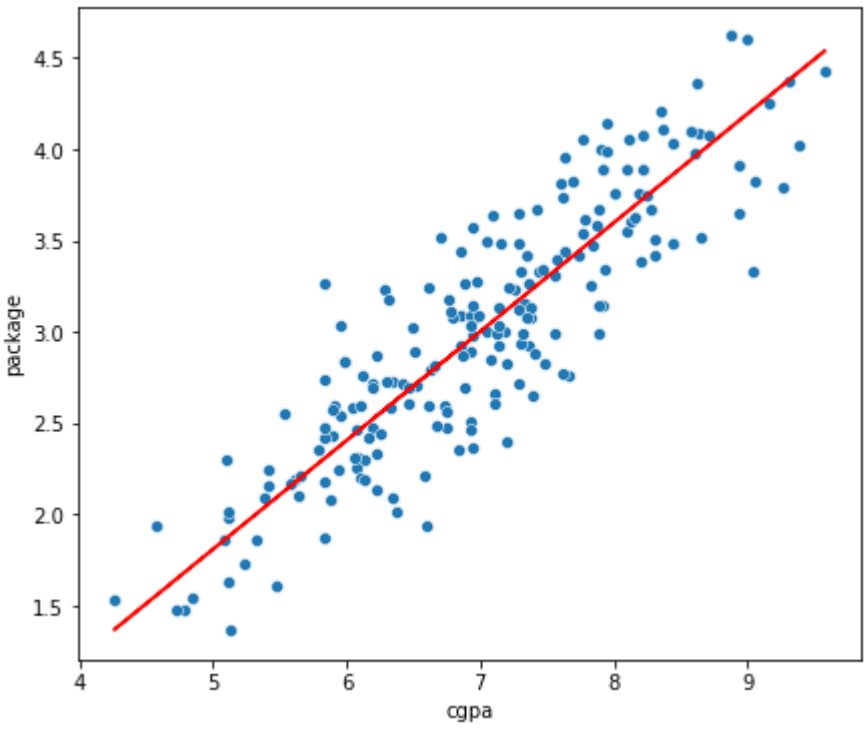
Out[24]: 77.06668029575103

In [25]:
```
sv.score(x_train, y_train)*100
```

Out[25]: 77.45351616879739

Step 7: Draw Prediction Line

In [29]:
```
plt.figure(figsize=(7,6))
sns.scatterplot(x='cgpa', y='package', data=dataset)
plt.plot(dataset['cgpa'], sv.predict(x), color='red')
plt.show()
```



Train data through SVR Model - Kernel: poly

In [33]:
```
sv1 = SVR(kernel='poly', degree=3)
sv1.fit(x_train, y_train)
```

Out[33]:
```
SVR(kernel='poly')
```

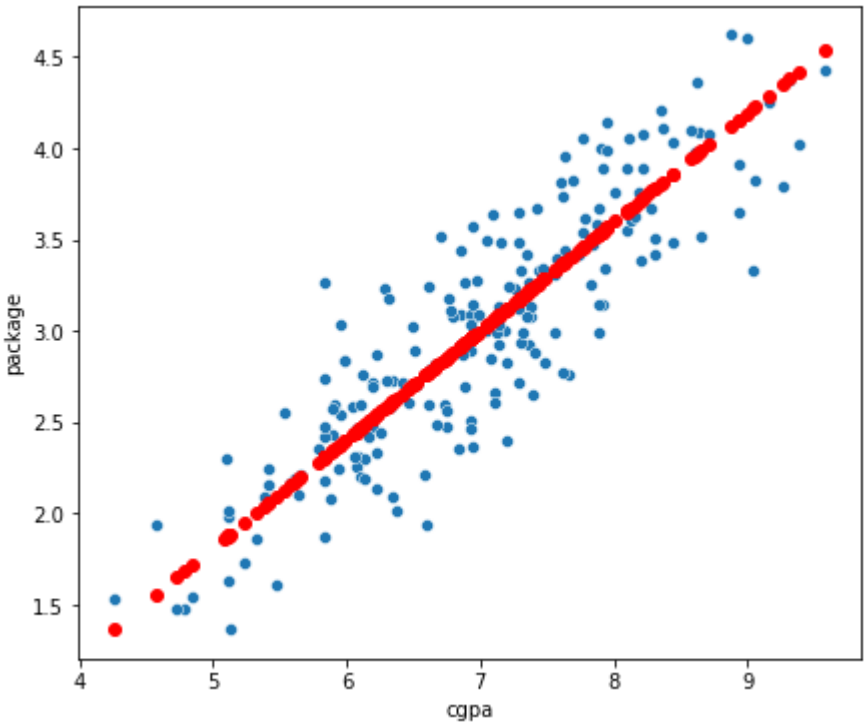
In [ ]:

In [34]:
```
sv.score(x_test, y_test)*100
```

Out[34]:
```
77.06668029575103
```

In [ ]:

In [35]:
```
plt.figure(figsize=(7,6))
sns.scatterplot(x='cgpa', y='package', data=dataset)
plt.scatter(dataset['cgpa'], sv.predict(x), color='red')
plt.show()
```



In [ ]: