45. Support Vector Machines (SVM) - Classification (Practical)

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from mlxtend.plotting import plot_decision_regions
```

In [4]:

```
dataset = pd.read_csv(r'Data/placement_3.csv')
dataset.head(3)
```

Out[4]:

	cgpa	score	placed
0	7.19	26	1
1	7.46	38	1
2	7.54	40	1

In [5]:

```
dataset.isnull().sum()
```

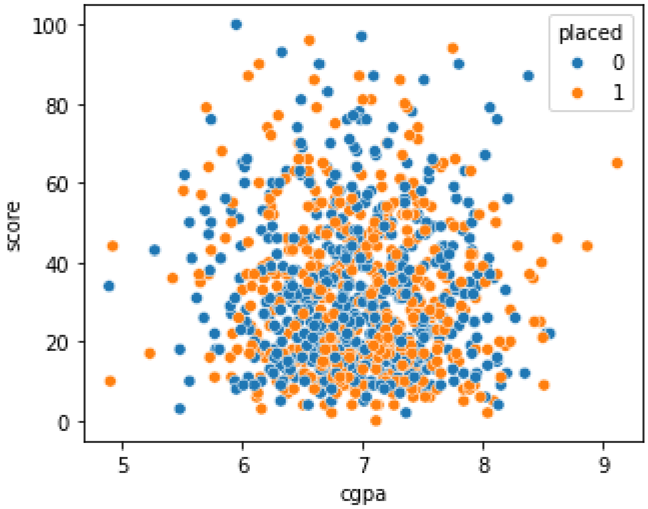
Out[5]:

```
cgpa      0
score     0
placed    0
dtype: int64
```

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

In [11]:

```
plt.figure(figsize=(5,4))
sns.scatterplot(x='cgpa', y='score', data=dataset, hue='placed')
plt.show()
```



Step 2: Separate dependent and independent variables

In [13]:
```
x = dataset.iloc[:,:-1]
y = dataset['placed']
```

Step 3: Split data into train and test data

In [14]:
```
from sklearn.model_selection import train_test_split
```

In [51]:
```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_st
```

Step 4: Train data through SVM Model

SVC: Support Vector Classifier. As our data output consist of 0 and 1, so we will apply classifier, SVC

In [52]:
```
from sklearn.svm import SVC
```

In [66]:
```
'''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 pre-compute the kernel matrix from data matrices; that matrix
    should be an array of shape ``(n_samples, n_samples)``.'''
# precomputed used for data that is one-encoded (in the form of 0 and 1)

sv = SVC(kernel='linear')
sv.fit(x_train, y_train)
```

Out[66]:
```
▾         SVC
SVC(kernel='linear')
```

Step 5: Check accuracy of SVM Model

In [63]:
```
sv.score(x_test, y_test)*100
```

Out[63]:
```
52.0
```

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

In [64]:
```
sv.score(x_train, y_train)*100
```

Out[64]:
```
49.875
```

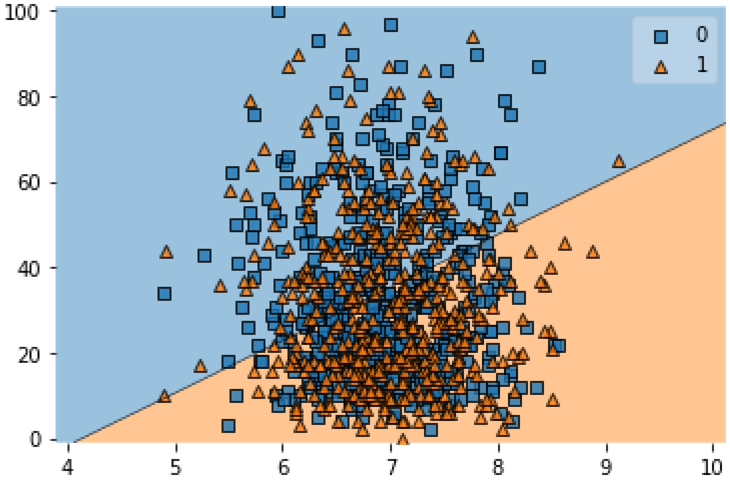
- Model is underfit

Step 7: Check boundaries of separation

In [65]:

```
plot_decision_regions(x.to_numpy(), y.to_numpy(), clf=sv)
plt.show()
```

```
C:\Users\rashi\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\bas
e.py:450: UserWarning: X does not have valid feature names, but SVC was fitted with
feature names
  warnings.warn(
```



- Lot of misclassifications

In [ ]: