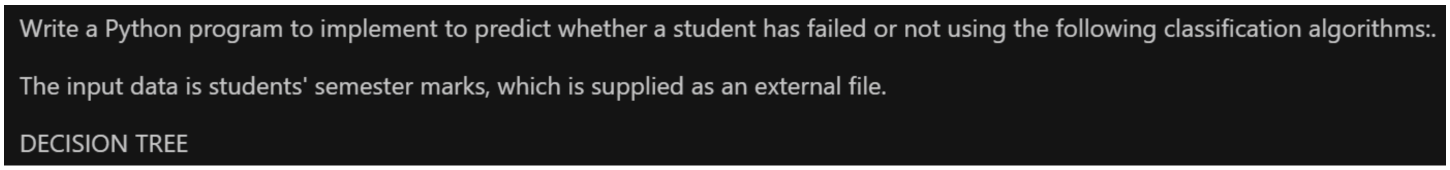
# SOFT COMPUTING ASSIGNMENT -2

**Perumalla Dharan**
**AP21110010201**

1.



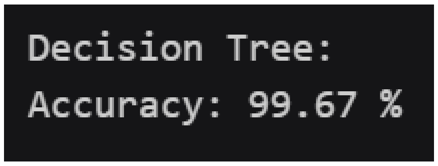
Write a Python program to implement to predict whether a student has failed or not using the following classification algorithms:.

The input data is students' semester marks, which is supplied as an external file.

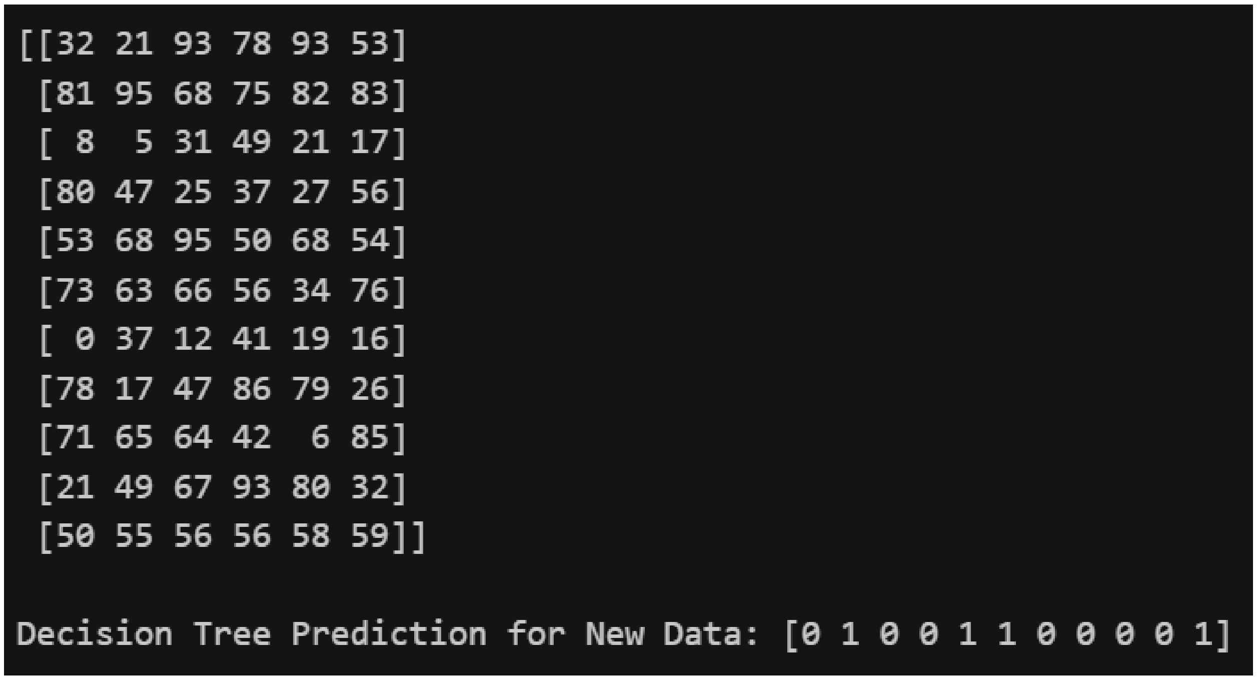
DECISION TREE

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score, classification_report


data = pd.read_csv('training_dataset_students(1000).csv')
y = data.iloc[:, -1].values
X = data.iloc[:, 1:-1].values

X_train,   X_test,   y_train,   y_test   =   train_test_split(X,   y,
 test_size=0.3, random_state=42)


dt_model = DecisionTreeClassifier()
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)
print("Decision Tree:")
print("Accuracy:",                     round(accuracy_score(y_test,
 y_pred_dt)*100,2),"%")
```

```python
data = pd.read_csv('students_testing.csv')
new_data = data.iloc[:, 1:-1].values

print(new_data)

pred_dt_new = dt_model.predict(new_data)
print("\nDecision Tree Prediction for New Data:", pred_dt_new)
```



```
Decision Tree:
Accuracy: 99.67 %
```



```
[[32 21 93 78 93 53]
 [81 95 68 75 82 83]
 [ 8  5 31 49 21 17]
 [80 47 25 37 27 56]
 [53 68 95 50 68 54]
 [73 63 66 56 34 76]
 [ 0 37 12 41 19 16]
 [78 17 47 86 79 26]
 [71 65 64 42  6 85]
 [21 49 67 93 80 32]
 [50 55 56 56 58 59]]

Decision Tree Prediction for New Data: [0 1 0 0 1 1 0 0 0 0 1]
```

2.

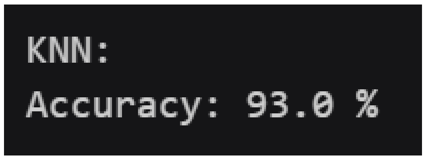
Write a Python program to implement to predict whether a student has failed or not using the following classification algorithms:.

The input data is students' semester marks, which is supplied as an external file.

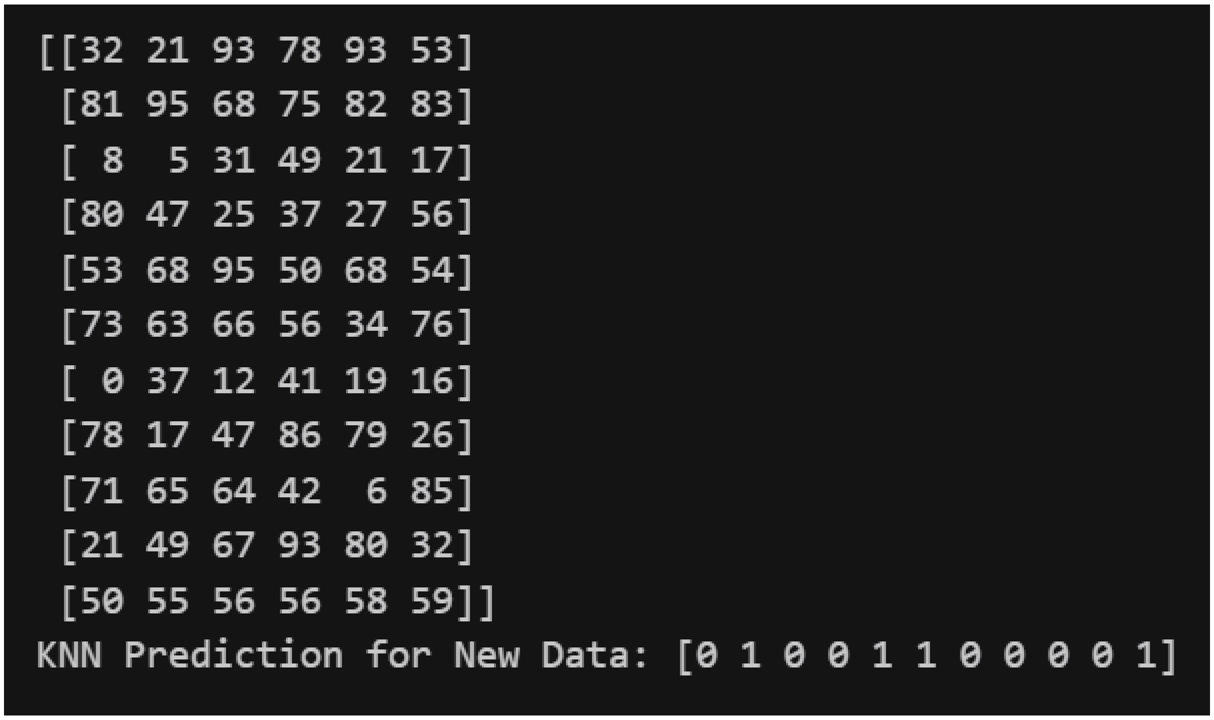
KNN

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score, classification_report


data = pd.read_csv('training_dataset_students(1000).csv')
y = data.iloc[:, -1].values
X = data.iloc[:, 1:-1].values

X_train, X_test, y_train, y_test = train_test_split(X, y,
 test_size=0.3, random_state=42)


knn_model = KNeighborsClassifier()
knn_model.fit(X_train, y_train)
y_pred_knn = knn_model.predict(X_test)
print("\nKNN:")
print("Accuracy:", round(accuracy_score(y_test,
 y_pred_knn)*100,2),"%")


data = pd.read_csv('students_testing.csv')
new_data = data.iloc[:, 1:-1].values

print(new_data)
```

```
pred_knn_new = knn_model.predict(new_data)
print("KNN Prediction for New Data:", pred_knn_new)
```



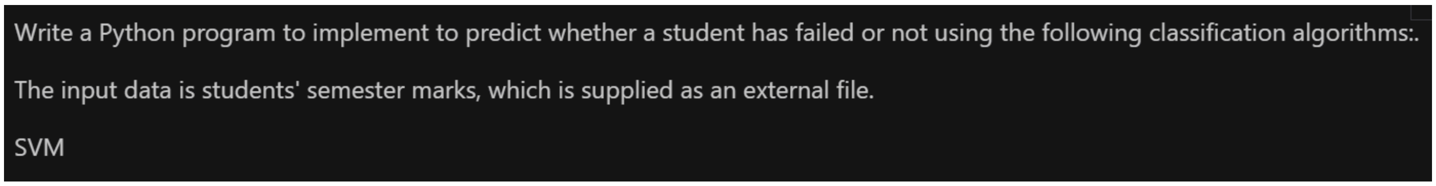
```
KNN:
Accuracy: 93.0 %
```



```
[[32 21 93 78 93 53]
 [81 95 68 75 82 83]
 [ 8  5 31 49 21 17]
 [80 47 25 37 27 56]
 [53 68 95 50 68 54]
 [73 63 66 56 34 76]
 [ 0 37 12 41 19 16]
 [78 17 47 86 79 26]
 [71 65 64 42  6 85]
 [21 49 67 93 80 32]
 [50 55 56 56 58 59]]
KNN Prediction for New Data: [0 1 0 0 1 1 0 0 0 0 1]
```

---

3.



Write a Python program to implement to predict whether a student has failed or not using the following classification algorithms:.

The input data is students' semester marks, which is supplied as an external file.

SVM

```
import pandas as pd
from sklearn.model_selection import train_test_split
```

```python
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score, classification_report


data = pd.read_csv('training_dataset_students(1000).csv')
y = data.iloc[:, -1].values
X = data.iloc[:, 1:-1].values

X_train, X_test, y_train, y_test = train_test_split(X, y,
 test_size=0.3, random_state=42)


svm_model = SVC()
svm_model.fit(X_train, y_train)
y_pred_svm = svm_model.predict(X_test)
print("\nSVM:")
print("Accuracy:", round(accuracy_score(y_test,
 y_pred_svm)*100,2),"%")


data = pd.read_csv('students_testing.csv')
new_data = data.iloc[:, 1:-1].values

print(new_data)

pred_svm_new = svm_model.predict(new_data)
print("SVM Prediction for New Data:", pred_svm_new)
```

```
SVM:
Accuracy: 97.0 %
```

```
[[32 21 93 78 93 53]
 [81 95 68 75 82 83]
 [ 8  5 31 49 21 17]
 [80 47 25 37 27 56]
 [53 68 95 50 68 54]
 [73 63 66 56 34 76]
 [ 0 37 12 41 19 16]
 [78 17 47 86 79 26]
 [71 65 64 42  6 85]
 [21 49 67 93 80 32]
 [50 55 56 56 58 59]]
SVM Prediction for New Data: [0 1 0 0 1 1 0 0 0 0 1]
```

---

4.

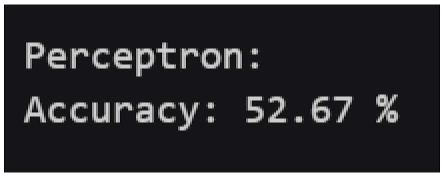
Write a Python program to implement to predict whether a student has failed or not using the following classification algorithms:.

The input data is students' semester marks, which is supplied as an external file.

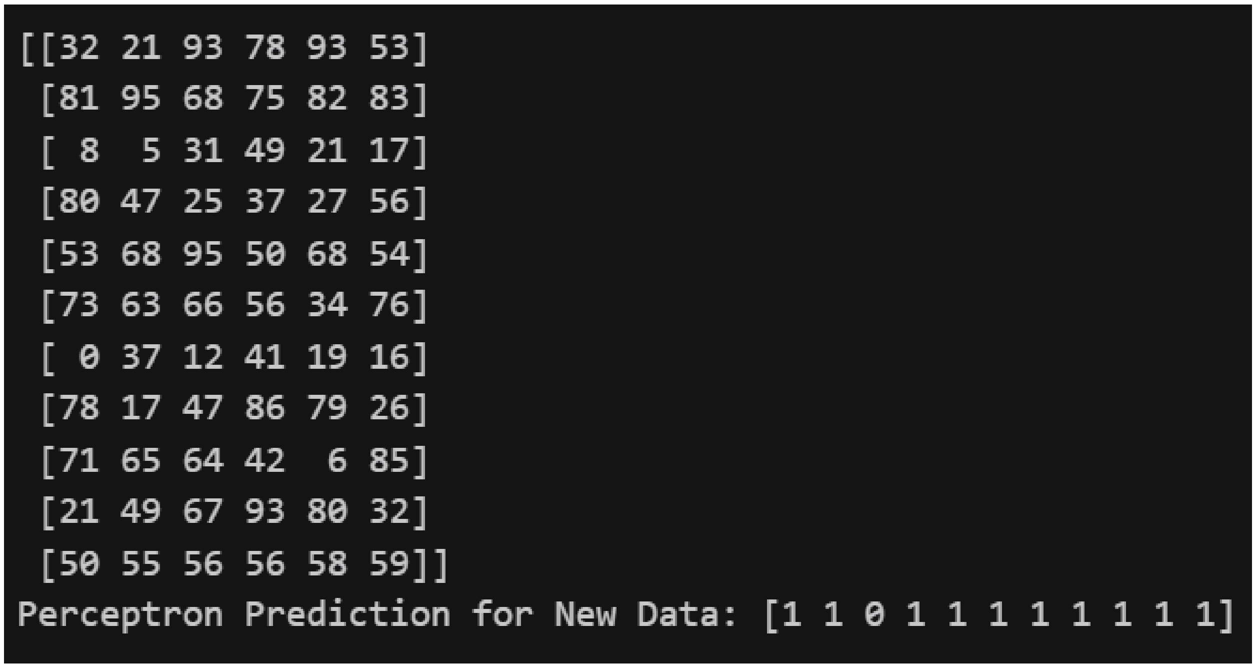
Perceptron (Use built-in function)

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score, classification_report
```

```
data = pd.read_csv('training_dataset_students(1000).csv')
y = data.iloc[:, -1].values
X = data.iloc[:, 1:-1].values

X_train, X_test, y_train, y_test = train_test_split(X, y,
 test_size=0.3, random_state=42)


perceptron_model = Perceptron()
perceptron_model.fit(X_train, y_train)
y_pred_perceptron = perceptron_model.predict(X_test)
print("\nPerceptron:")
print("Accuracy:", round(accuracy_score(y_test,
 y_pred_perceptron)*100,2),"%")


data = pd.read_csv('students_testing.csv')
new_data = data.iloc[:, 1:-1].values

print(new_data)

pred_perceptron_new = perceptron_model.predict(new_data)
print("Perceptron Prediction for New Data:", pred_perceptron_new)
```



```
Perceptron:
Accuracy: 52.67 %
```



```
[[32 21 93 78 93 53]
 [81 95 68 75 82 83]
 [ 8  5 31 49 21 17]
 [80 47 25 37 27 56]
 [53 68 95 50 68 54]
 [73 63 66 56 34 76]
 [ 0 37 12 41 19 16]
 [78 17 47 86 79 26]
 [71 65 64 42  6 85]
 [21 49 67 93 80 32]
 [50 55 56 56 58 59]]
Perceptron Prediction for New Data: [1 1 0 1 1 1 1 1 1 1 1]
```