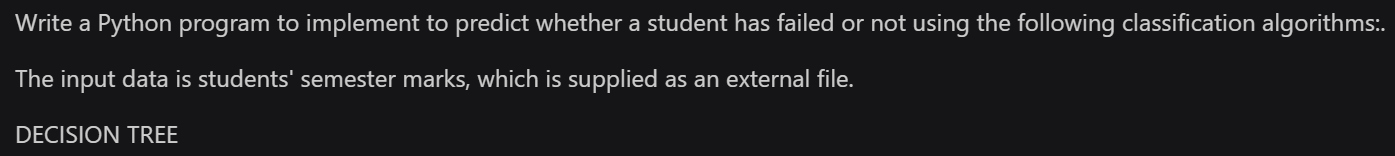
**SOFT COMPUTING**

**ASSIGNMENT -2**

**Perumalla Dharan**

**AP21110010201**



*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),"%")

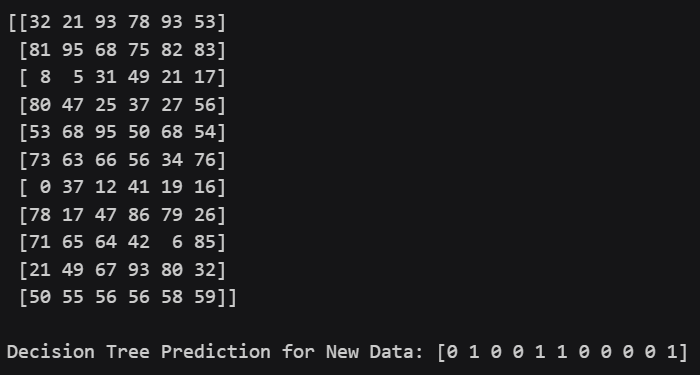
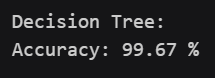
data = pd.read\_csv('students\_testing.csv')

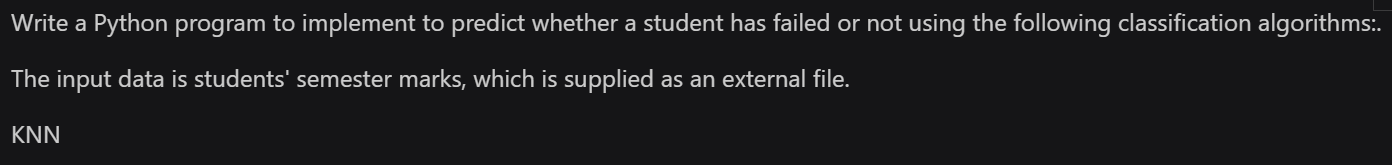
new\_data = data.iloc[:, 1:-1].values

print(new\_data)

pred\_dt\_new = dt\_model.predict(new\_data)

print("\nDecision Tree Prediction for New Data:", pred\_dt\_new)





*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),"%")

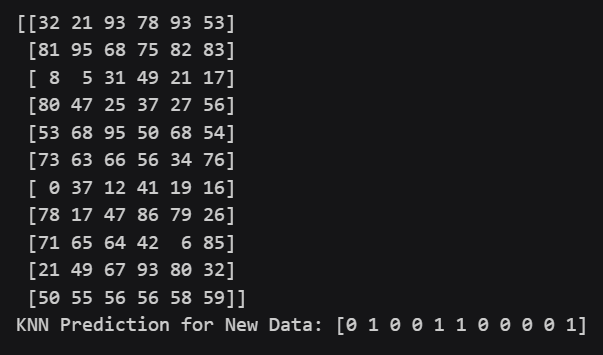
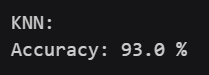
data = pd.read\_csv('students\_testing.csv')

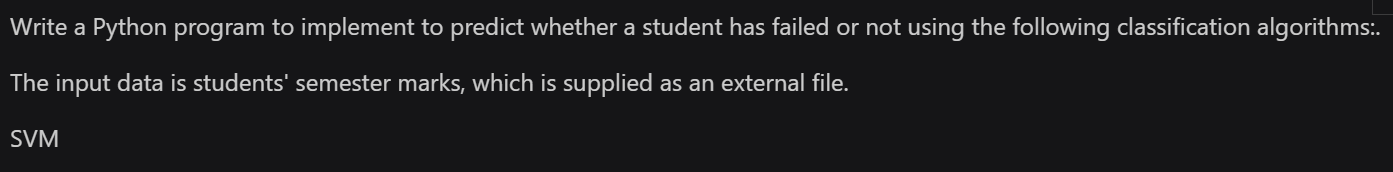
new\_data = data.iloc[:, 1:-1].values

print(new\_data)

pred\_knn\_new = knn\_model.predict(new\_data)

print("KNN Prediction for New Data:", pred\_knn\_new)





*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)

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),"%")

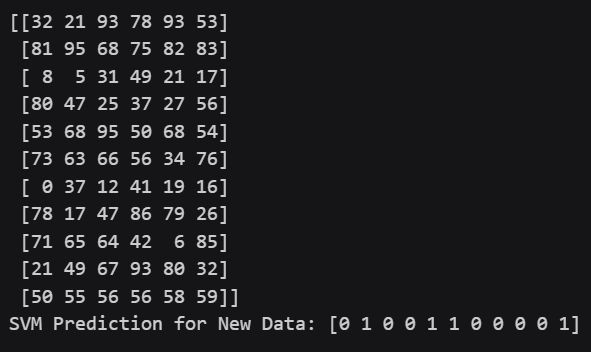
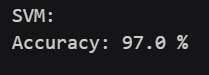
data = pd.read\_csv('students\_testing.csv')

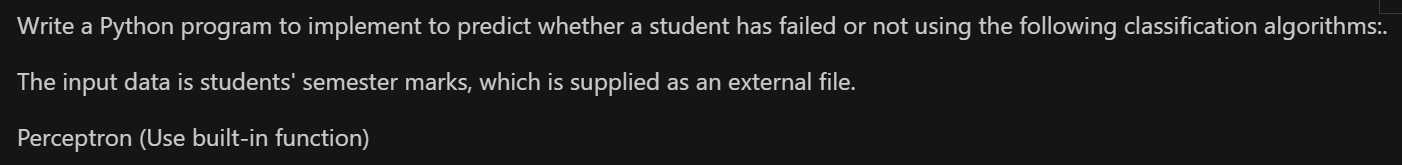
new\_data = data.iloc[:, 1:-1].values

print(new\_data)

pred\_svm\_new = svm\_model.predict(new\_data)

print("SVM Prediction for New Data:", pred\_svm\_new)





*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)

