Assignment 7

Problem 1)

Solution,

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report

df = pd.read_csv('data_banknote_authentication.csv')
x = df.drop(columns=['class'])
y = df['class']

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2,
random_state=20)

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

Y_pred_linear = svm_linear.predict(x_test)

confusion_matrix_linear = confusion_matrix(y_test, Y_pred_linear)
classification_report_linear = classification_report(y_test, Y_pred_linear)

print('confusion matrix (linear kernel):\n', confusion_matrix_linear)
print('classification_report (linear kernel):\n', classification_report_linear)



svm_rbf = SVC(kernel='rbf')
svm_rbf.fit(x_train, y_train)

y_pred_rbf = svm_rbf.predict(x_test)

confusion_matrix_rbf = confusion_matrix(y_test, y_pred_rbf)
classification_report_rbf = classification_report(y_test, y_pred_rbf)

print('confusion matrix (rbf kernel):\n', confusion_matrix_rbf)
print('classification_report (rbf kernel):\n', classification_report_rbf)
```

Comparison of Linear and RBF SVM Models:

The linear kernel performed well with high accuracy and precision.

The RBF kernel may capture more complex relationships in the data.

Comparing precision, recall, and F1-score will indicate which model is better suited for this dataset.

```
C:\Users\user\PycharmProjects\pythonProjectTest\.venv\Scripts\python.exe "C:\Users\user\Pyc
confusion matrix (linear kernel):
 [[152   2]
 [  0 121]]
classification_report (linear kernel):
              precision    recall  f1-score   support

           0       1.00      0.99      0.99       154
           1       0.98      1.00      0.99       121

    accuracy                           0.99       275
   macro avg       0.99      0.99      0.99       275
weighted avg       0.99      0.99      0.99       275

confusion matrix (rbf kernel):
 [[154   0]
 [  0 121]]
classification_report (rbf kernel):
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       154
           1       1.00      1.00      1.00       121

    accuracy                           1.00       275
   macro avg       1.00      1.00      1.00       275
weighted avg       1.00      1.00      1.00       275


Process finished with exit code 0
```

Project > AI with python > assingment 7 > 1.py

Problem 2

Solutipn>

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier

data = pd.read_csv('suv.csv')

x = data[['Age','EstimatedSalary']]
y = data['Purchased']

x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.2)

scaler = StandardScaler()
x_train=scaler.fit_transform(x_train)
```

```
x_test=scaler.transform(x_test)

dt_entropy = DecisionTreeClassifier(criterion='entropy', random_state=20)
dt_entropy = dt_entropy.fit(x_train,y_train)

y_pred_entropy = dt_entropy.predict(x_test)
confusion_matrix_entropy = confusion_matrix(y_test,y_pred_entropy)
classification_report_entropy = classification_report(y_test,y_pred_entropy)

print('Confusion Matrix(Entropy): \n', confusion_matrix_entropy)
print('Classification Report: \n', classification_report_entropy)

dt_gini = DecisionTreeClassifier(criterion='gini', random_state=20)
dt_gini = dt_gini.fit(x_train,y_train)

y_pred_gini = dt_gini.predict(x_test)
confusion_matrix_gini = confusion_matrix(y_test,y_pred_gini)
classification_report_gini = classification_report(y_test,y_pred_gini)

print('Confusion Matrix(Gini): \n', confusion_matrix_gini)
print('Classification Report: \n', classification_report_gini)
```

Comparison of Decision Tree Models:

The entropy-based model might be more sensitive to class distribution.

The gini-based model is often faster but might not split as optimally in some cases.

Checking precision, recall, and F1-score helps determine which model is better suited for this dataset.

```
C:\Users\user\PycharmProjects\pythonProjectTest\.venv\Scripts\python.exe "C:\Users\user\PycharmProjects\pytho
Confusion Matrix(Entropy):
 [[45 10]
 [ 3 22]]
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.82      0.87        55
           1       0.69      0.88      0.77        25

    accuracy                           0.84        80
   macro avg       0.81      0.85      0.82        80
weighted avg       0.86      0.84      0.84        80

Confusion Matrix(Gini):
 [[45 10]
 [ 6 19]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.82      0.85        55
           1       0.66      0.76      0.70        25

    accuracy                           0.80        80
   macro avg       0.77      0.79      0.78        80
weighted avg       0.81      0.80      0.80        80


Process finished with exit code 0
```