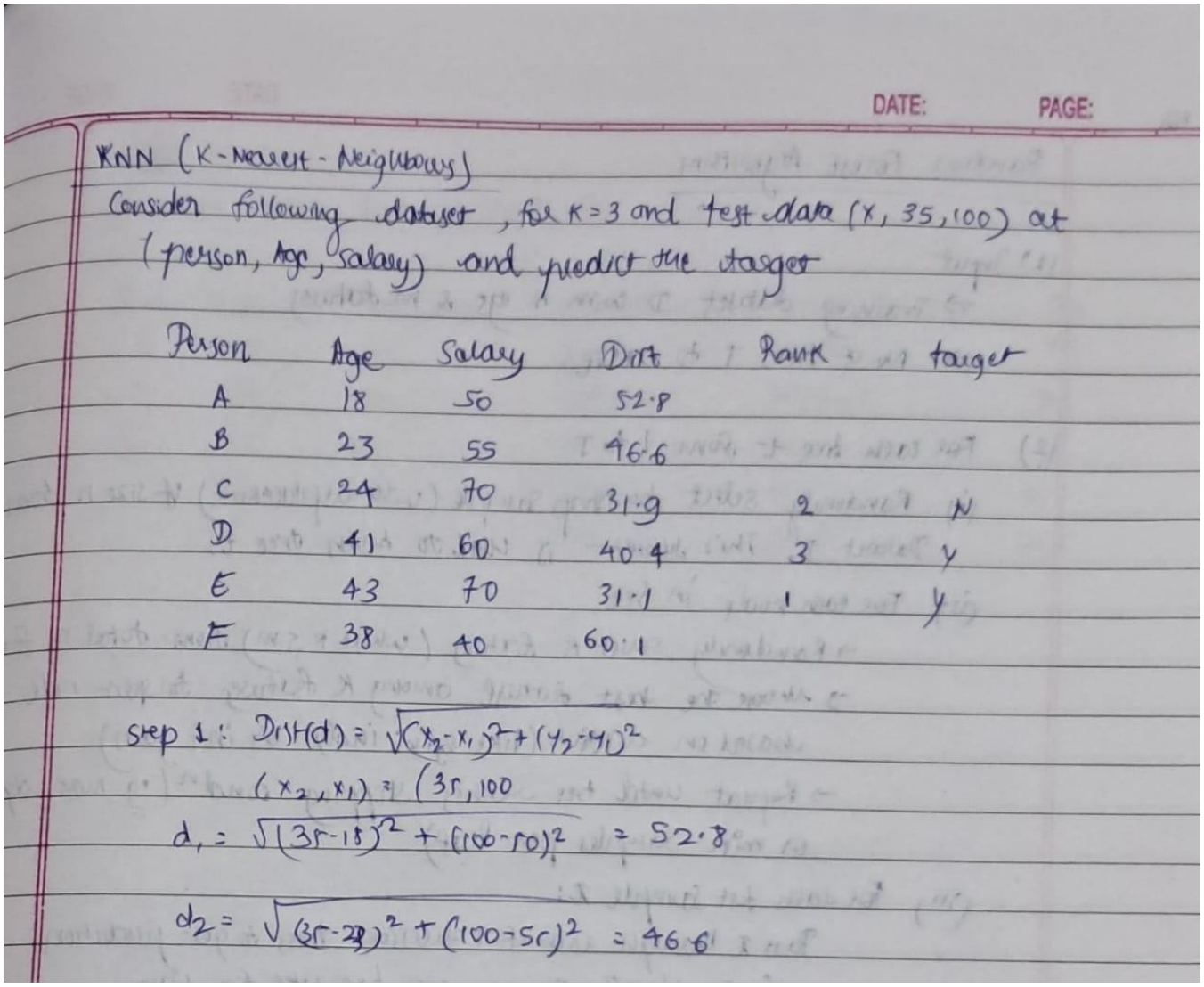
Program 6

Build KNN Classification model for a given dataset

Screenshot:



DATE: PAGE:

KNN (K-Nearest-Neighbours)

Consider following dataset, for K=3 and test data (x, 35, 100) at (person, Age, Salary) and predict the target

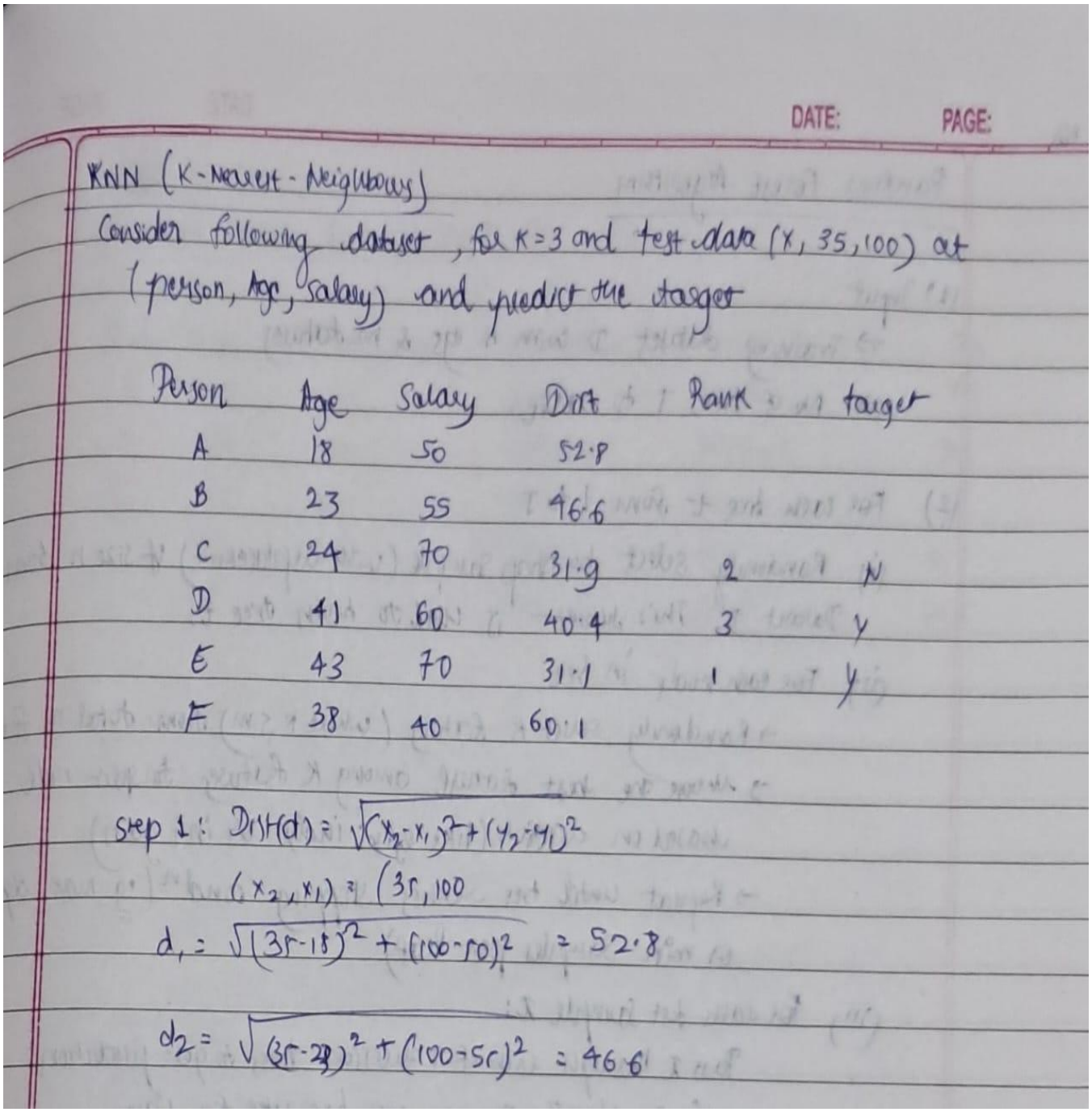
Person	Age	Salary	Dist	Rank	target
A	18	50	52.8		
B	23	55	46.6		
C	24	70	31.9	2	N
D	41	60	40.4	3	Y
E	43	70	31.1	1	Y
F	38	40	60.1		

step 1: $Dist(d) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

$(x_2, x_1) = (35, 100$

$d_1 = \sqrt{(35-18)^2+(100-50)^2} = 52.8$

$d_2 = \sqrt{(35-23)^2+(100-55)^2} = 46.6$



DATE: PAGE:

KNN (K-Nearest-Neighbours)

Consider following dataset, for K=3 and test data (x, 35, 100) at (person, Age, Salary) and predict the target

Person	Age	Salary	Dist	Rank	target
A	18	50	52.8		
B	23	55	46.6		
C	24	70	31.9	2	N
D	41	60	40.4	3	Y
E	43	70	31.1	1	Y
F	38	40	60.1		

step 1: $Dist(d) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$

$(x_2, x_1) = (35, 100$

$d_1 = \sqrt{(35-18)^2+(100-50)^2} = 52.8$

$d_2 = \sqrt{(35-23)^2+(100-55)^2} = 46.6$

Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
import matplotlib.pyplot as plt

import seaborn as sns


# Load the Iris dataset

iris = pd.read_csv("/content/iris (2).csv")


# Prepare the data

X = iris.drop('species', axis='columns')

y = iris.species


# Split the data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Initialize the KNN classifier with k=3

knn = KNeighborsClassifier(n_neighbors=3)  # You can experiment with different k values


# Train the classifier

knn.fit(X_train, y_train)


# Make predictions

y_pred = knn.predict(X_test)


# Evaluate the model

accuracy = accuracy_score(y_test, y_pred)
```

```
print(f"Accuracy: {accuracy}")


print("\nClassification Report:\n", classification_report(y_test, y_pred))


cm = confusion_matrix(y_test, y_pred)
print("\nConfusion Matrix:\n", cm)


# Visualize the confusion matrix (optional)
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues",
        xticklabels=["Setosa", "Versicolor", "Virginica"],
        yticklabels=["Setosa", "Versicolor", "Virginica"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
```