Program 6

## Build KNN Classification model for a given dataset

## Screenshot:

5					DATE:	
-	KNN (K-Nowest-Neighbours)			participal grant contract		
4	Consider following dataset, for			K=3 and test data (	x, 35,100	) at
1	(person, Ag	, salary	ond y	neduct the stages	Turis Ex	(1)
1	Person	Age	Salary	Dirt & Pank	at taige	et
1	A	18	50	52.8	,	
1	B	23	55	7 46-6 Mar + and	4000 197	(2)
	Lasil At / C. Manit	24	70	31.9	and the	
1	D	41	600	40.4	met v	
	E	43	70	31-1	7 4	
	lotob matigues	38	40	60.1		
	my of section	BURN K A	NO MANY	is the operation of		
	Step 1: Drift	(d) = V	X2-X, 3-+(	42.40°		
30	a laboration 6x	(x) = (	35,100	not Show throat a		
	d, = 5(3	58-1852.	+ (100-10)	2 = 52.8		
				)2 = 466	3	

_		DATE: PAGE:		
_	KNN (K-Maryt-Neighboury)	se K=3 and test data (X, 35, 100) at		
_	Consider following dataset			
-	(person, top, salary) and	nedict the stages		
-	product of a 200	predict the tagget		
4	Person Age Salary	Dirt & Pank target		
4	A 18 50	52.8		
1		T 46-6 MAR - + 21th MIN 199 (3)		
1	C 24 70	2 31.9 2 2 N		
1	D 41 60	40.4 3 W		
	E 43 70	31.1		
	F 38 40	60.1		
	and another themes to have	This tend are annually on		
	step 1: Distrol = V(x2-x1,3-4	- (42-41)2		
1	(x2, x1) = (3r, 100	and Palent de la		
	d: 5(35-18)2 + (100-1	0)2 32 0 31		

## 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()
```