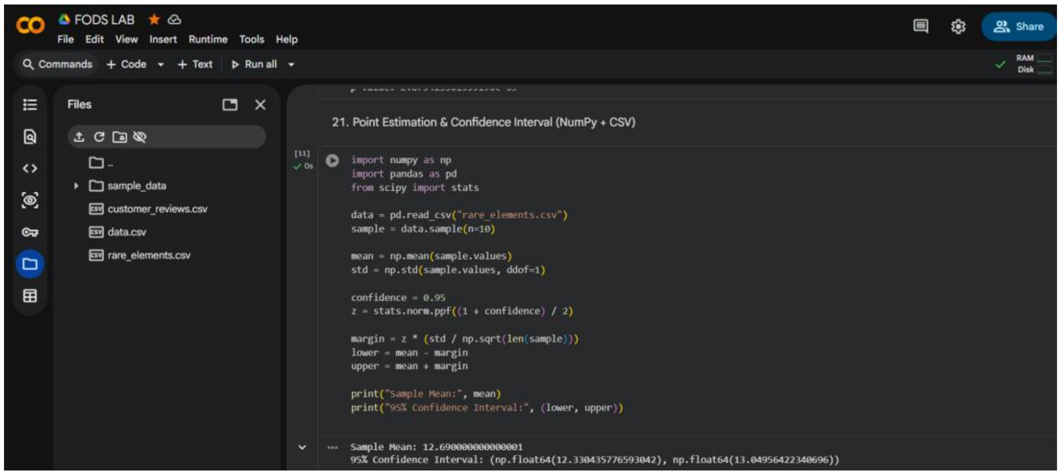
R.Sathish DATE:10/02/2026
192324204

EXP:21

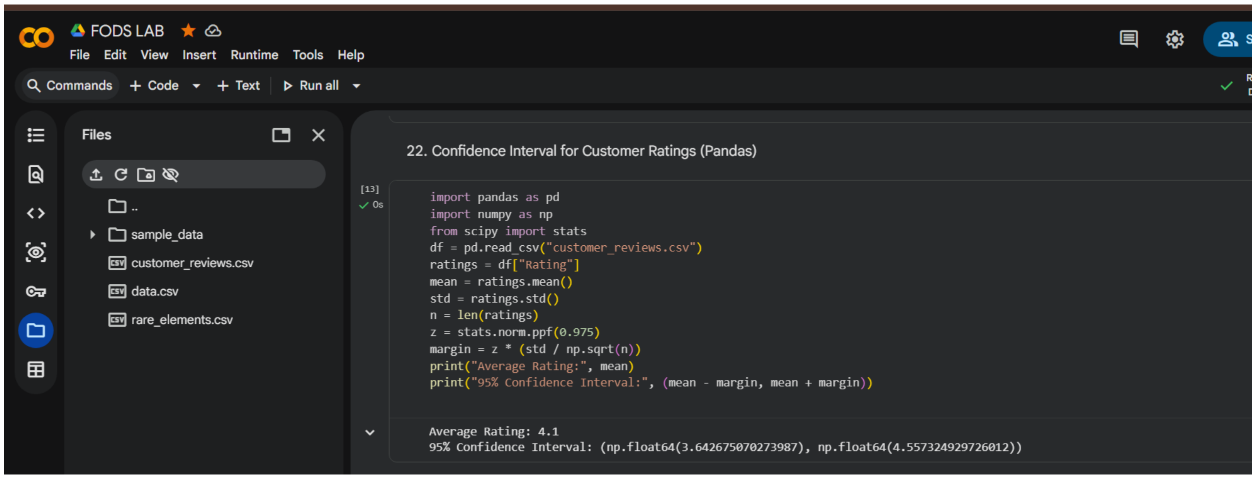


21. Point Estimation & Confidence Interval (NumPy + CSV)

```
import numpy as np
import pandas as pd
from scipy import stats

data = pd.read_csv("rare_elements.csv")
sample = data.sample(n=10)

mean = np.mean(sample.values)
std = np.std(sample.values, ddof=1)

confidence = 0.95
z = stats.norm.ppf((1 + confidence) / 2)

margin = z * (std / np.sqrt(len(sample)))
lower = mean - margin
upper = mean + margin

print("Sample Mean:", mean)
print("95% Confidence Interval:", (lower, upper))
```

```
Sample Mean: 12.690000000000001
95% Confidence Interval: (np.float64(12.330435776593042), np.float64(13.049564223406696))
```

EXP:22

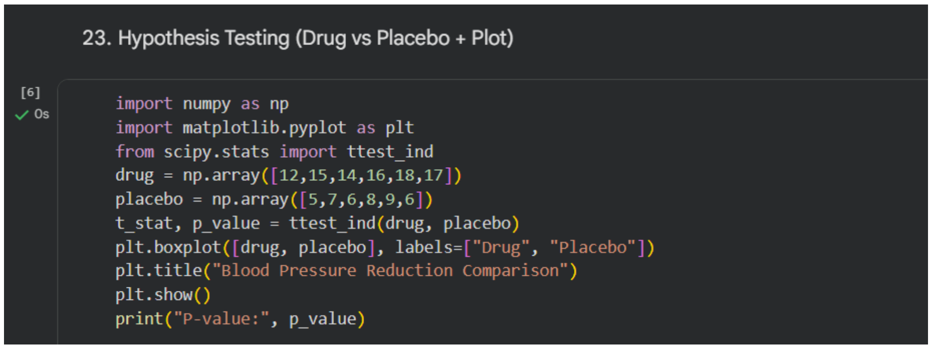

FODS LAB
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Files
sample_data
customer_reviews.csv
data.csv
rare_elements.csv

22. Confidence Interval for Customer Ratings (Pandas)

```
import pandas as pd
import numpy as np
from scipy import stats
df = pd.read_csv("customer_reviews.csv")
ratings = df["Rating"]
mean = ratings.mean()
std = ratings.std()
n = len(ratings)
z = stats.norm.ppf(0.975)
margin = z * (std / np.sqrt(n))
print("Average Rating:", mean)
print("95% Confidence Interval:", (mean - margin, mean + margin))
```

```
Average Rating: 4.1
95% Confidence Interval: (np.float64(3.642675070273987), np.float64(4.557324929726012))
```

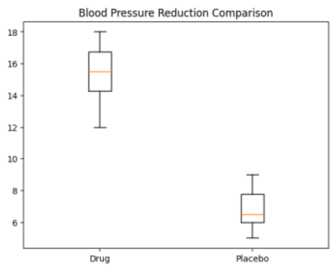
EXP:23



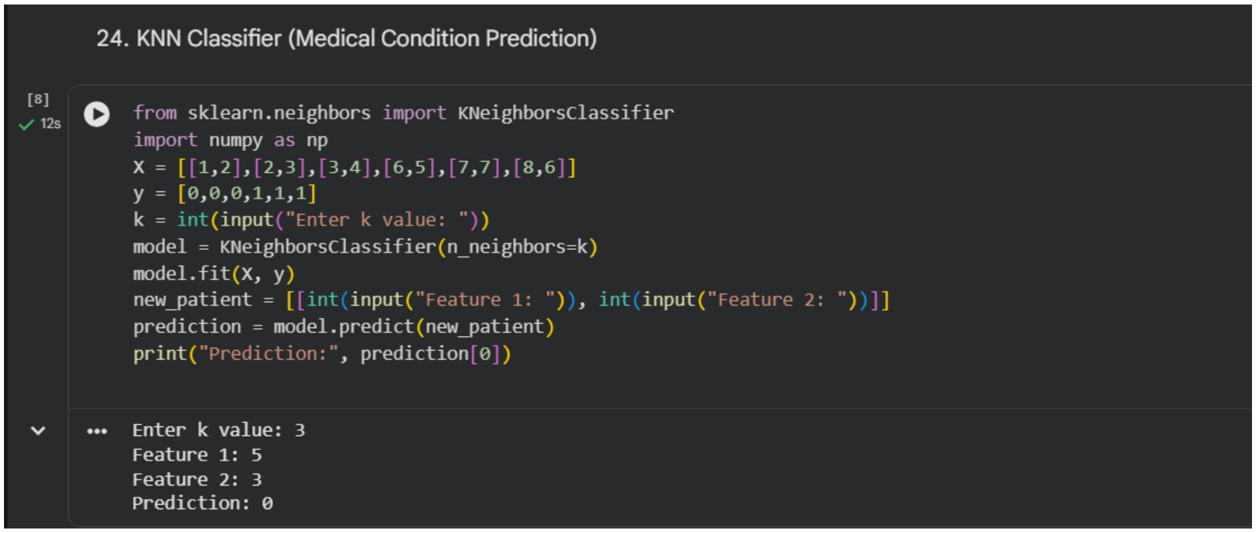
23. Hypothesis Testing (Drug vs Placebo + Plot)

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind
drug = np.array([12,15,14,16,18,17])
placebo = np.array([5,7,6,8,9,6])
t_stat, p_value = ttest_ind(drug, placebo)
plt.boxplot([drug, placebo], labels=["Drug", "Placebo"])
plt.title("Blood Pressure Reduction Comparison")
plt.show()
print("P-value:", p_value)
```

OP:



EXP:24

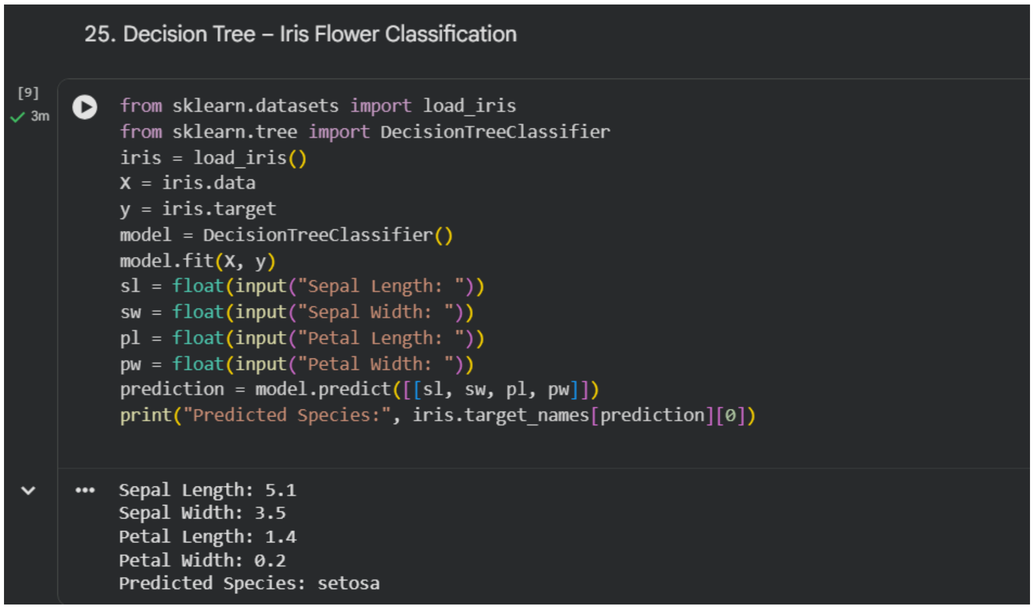

24. KNN Classifier (Medical Condition Prediction)

[8]

```
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
X = [[1,2],[2,3],[3,4],[6,5],[7,7],[8,6]]
y = [0,0,0,1,1,1]
k = int(input("Enter k value: "))
model = KNeighborsClassifier(n_neighbors=k)
model.fit(X, y)
new_patient = [[int(input("Feature 1: ")), int(input("Feature 2: "))]]
prediction = model.predict(new_patient)
print("Prediction:", prediction[0])
```

```
Enter k value: 3
Feature 1: 5
Feature 2: 3
Prediction: 0
```

EXP:25


25. Decision Tree – Iris Flower Classification

[9]

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
iris = load_iris()
X = iris.data
y = iris.target
model = DecisionTreeClassifier()
model.fit(X, y)
sl = float(input("Sepal Length: "))
sw = float(input("Sepal Width: "))
pl = float(input("Petal Length: "))
pw = float(input("Petal Width: "))
prediction = model.predict([[sl, sw, pl, pw]])
print("Predicted Species:", iris.target_names[prediction][0])
```

```
Sepal Length: 5.1
Sepal Width: 3.5
Petal Length: 1.4
Petal Width: 0.2
Predicted Species: setosa
```