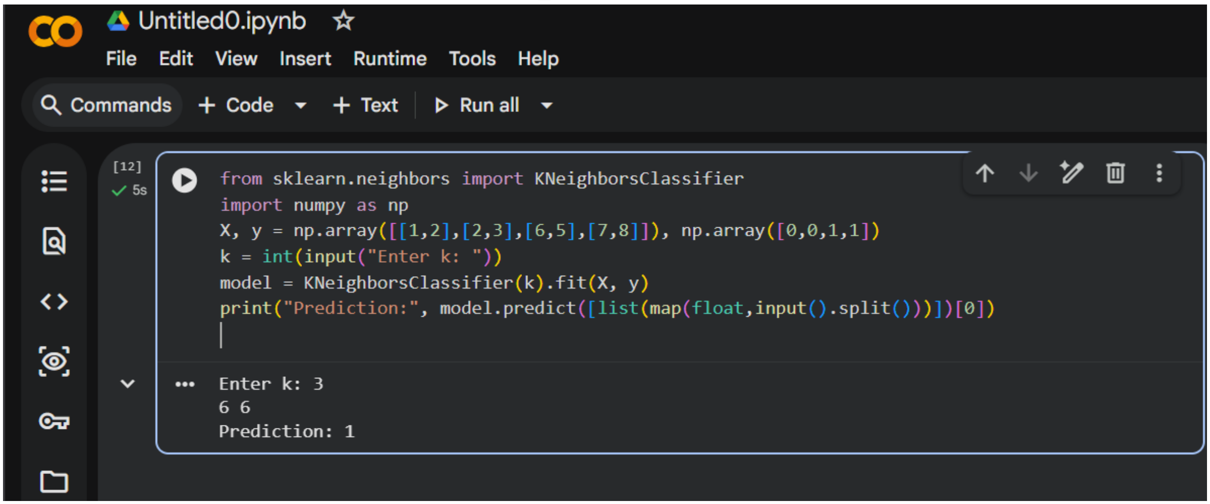
DAY – 6

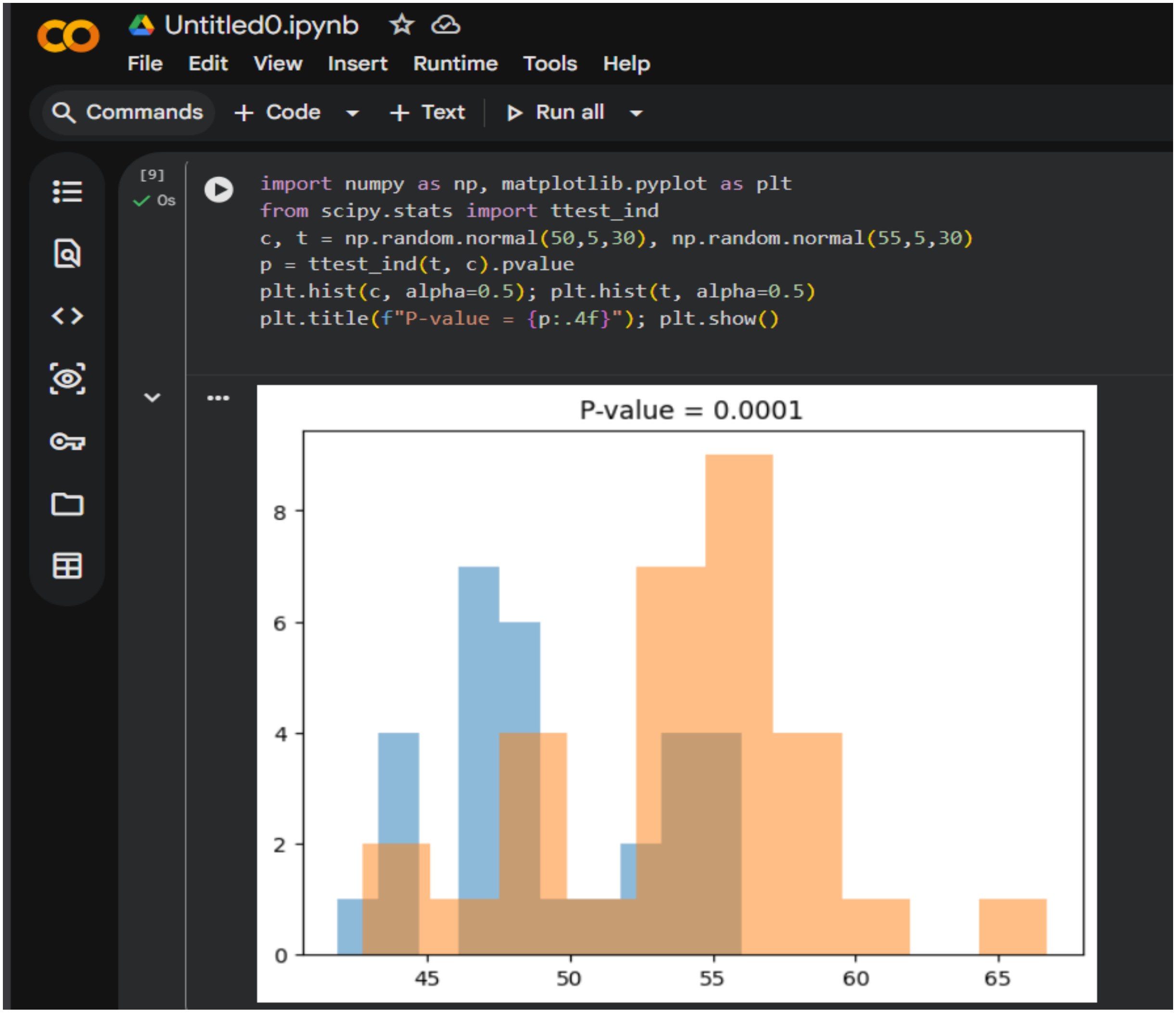
[10.02.2025]

Exp : 21-25

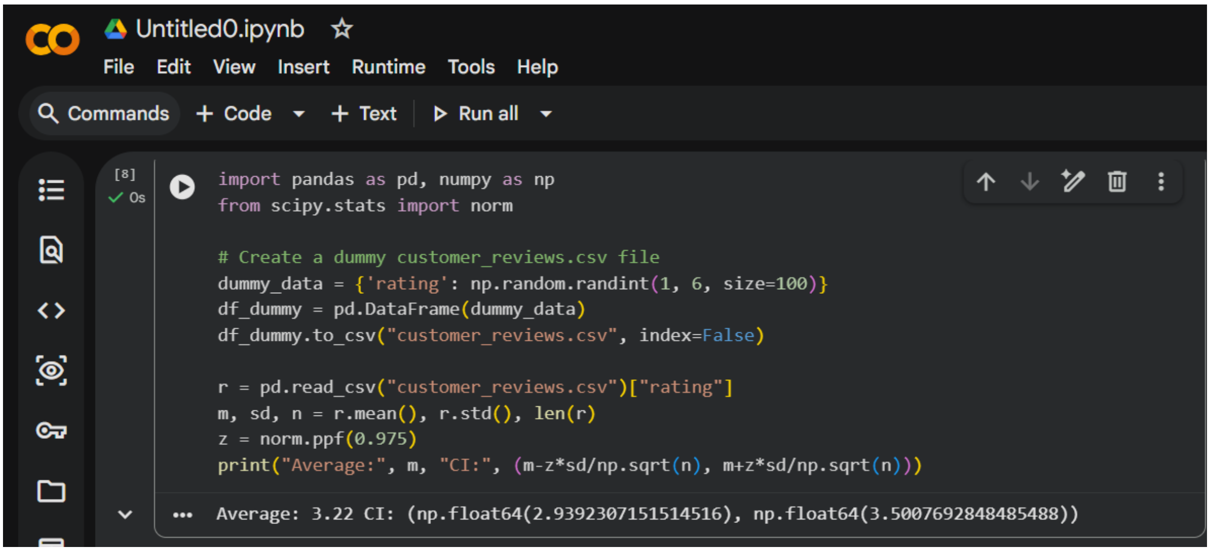


```
from sklearn.neighbors import KNeighborsClassifier
import numpy as np
X, y = np.array([[1,2],[2,3],[6,5],[7,8]]), np.array([0,0,1,1])
k = int(input("Enter k: "))
model = KNeighborsClassifier(k).fit(X, y)
print("Prediction:", model.predict([list(map(float,input().split()))])[0])
```

```
Enter k: 3
6 6
Prediction: 1
```



```
import numpy as np, matplotlib.pyplot as plt
from scipy.stats import ttest_ind
c, t = np.random.normal(50,5,30), np.random.normal(55,5,30)
p = ttest_ind(t, c).pvalue
plt.hist(c, alpha=0.5); plt.hist(t, alpha=0.5)
plt.title(f"P-value = {p:.4f}"); plt.show()
```

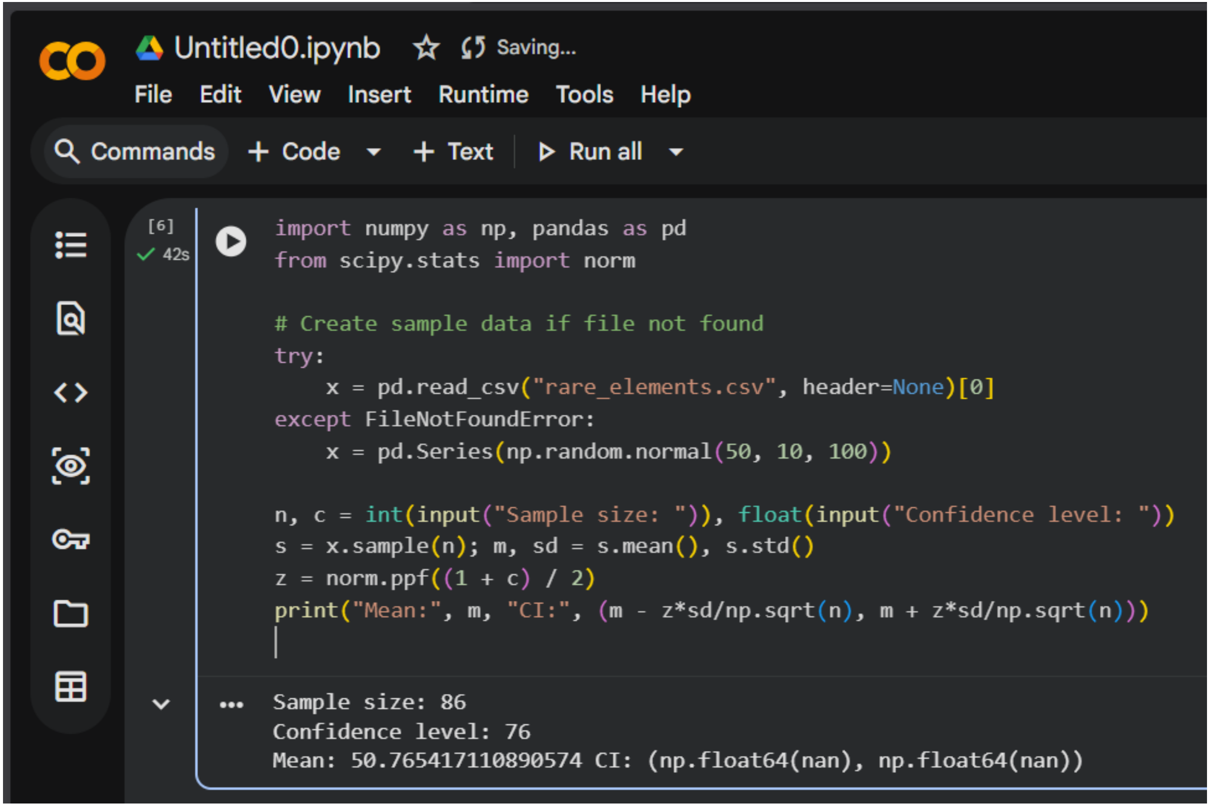


```
import pandas as pd, numpy as np
from scipy.stats import norm

# Create a dummy customer_reviews.csv file
dummy_data = {'rating': np.random.randint(1, 6, size=100)}
df_dummy = pd.DataFrame(dummy_data)
df_dummy.to_csv("customer_reviews.csv", index=False)

r = pd.read_csv("customer_reviews.csv")["rating"]
m, sd, n = r.mean(), r.std(), len(r)
z = norm.ppf(0.975)
print("Average:", m, "CI:", (m-z*sd/np.sqrt(n), m+z*sd/np.sqrt(n)))
```

```
Average: 3.22 CI: (np.float64(2.9392307151514516), np.float64(3.5007692848485488))
```

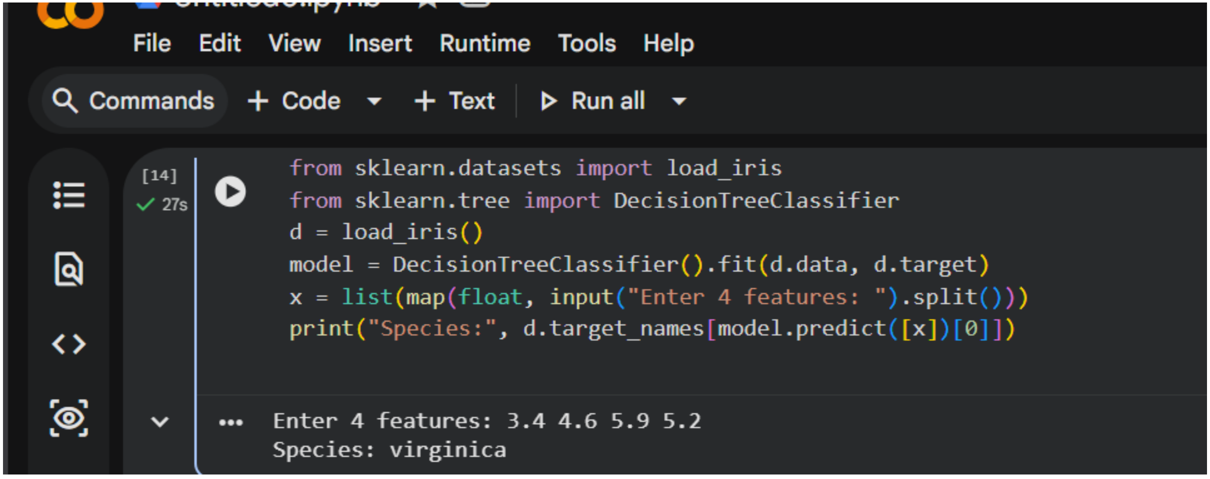


```
import numpy as np, pandas as pd
from scipy.stats import norm

# Create sample data if file not found
try:
    x = pd.read_csv("rare_elements.csv", header=None)[0]
except FileNotFoundError:
    x = pd.Series(np.random.normal(50, 10, 100))

n, c = int(input("Sample size: ")), float(input("Confidence level: "))
s = x.sample(n); m, sd = s.mean(), s.std()
z = norm.ppf((1 + c) / 2)
print("Mean:", m, "CI:", (m - z*sd/np.sqrt(n), m + z*sd/np.sqrt(n)))
```

```
Sample size: 86
Confidence level: 76
Mean: 50.765417110890574 CI: (np.float64(nan), np.float64(nan))
```



```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
d = load_iris()
model = DecisionTreeClassifier().fit(d.data, d.target)
x = list(map(float, input("Enter 4 features: ").split()))
print("Species:", d.target_names[model.predict([x])[0]])
```

```
Enter 4 features: 3.4 4.6 5.9 5.2
Species: virginica
```