College of Electrical & Mechanical Engineering, NUST

NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

44 CE A

AI & Decision Support System

Lab 9

NAME	REG NO	Syndicate
Syed Shaheer Suhaib	429292	A

Table of Contents

Questions

Task1:

Age	Loan	Class (Defaulter)
25	40000	0
35	60000	0
45	80000	0
20	20000	0
35	120000	0
52	18000	0
23	95000	1
40	62000	1
60	100000	1
48	220000	1
33	150000	1
48	142000	?

CODE:

```
import math
import numpy as np

def naive_bayes(data, classes, sample):


    sample = np.array(sample)

    values = list(data.values())
    total = len(values)

    print("\n===  1: prior probabilites ===")
    priors = []
    for c in classes:
        count = values.count(c)
        p = count / total
        priors.append(p)
        print(f"P(class={c}) = {p}")


    posteriors = []

    print("\n===  2: CLASS parameters & likelihood ===")
    for i, c in enumerate(classes):

        class_points = np.array([key for key, val in data.items() if val == c])
        print(f"\nClass {c} data = {class_points}")

        mean_vec = np.mean(class_points, axis=0)
        var_vec  = np.var(class_points, axis=0)

        print(f"Mean = {mean_vec}")
        print(f"Variance = {var_vec}")


        likelihood_dims = (1 / np.sqrt(2*np.pi*var_vec)) * np.exp(-((sample -
mean_vec)**2) / (2*var_vec))
        likelihood = np.prod(likelihood_dims)
```

```
        print(f"Likelihood = {likelihood}")

        post = likelihood * priors[i]
        posteriors.append(post)
        print(f"Unnormalized posterior = {post}")


    evidence = sum(posteriors)

    print("\n=== STEP 3: NORMALIZED POSTERIORS ===")
    for i, c in enumerate(classes):
        posteriors[i]=  posteriors[i] / evidence
        print(posteriors[i])
        print( )


    prediction = classes[np.argmax(posteriors)]
    print("\n=== FINAL DECISION ===")
    print(f"Predicted class = {prediction}")
    return prediction

data = {
    (25, 40000): 0,
    (35, 60000): 0,
    (45, 80000): 0,
    (20, 20000): 0,
    (35, 120000): 0,
    (52, 18000): 0,
    (23, 95000): 1,
    (40, 62000): 1,
    (60, 100000): 1,
    (48, 220000): 1,
    (33, 150000): 1,

}

naive_bayes(data,[0,1],(48,142000))
```

Result:

```
=== STEP 5: NORMALIZED POSTERIORS ===
0.07096075899894243

0.9290392410010576


=== FINAL DECISION ===
Predicted class = 1
PS D:\DE 44 CE 2022\mySemData\sem 7\AI>
chpad  ⊗ 1 ⚠ 0
```

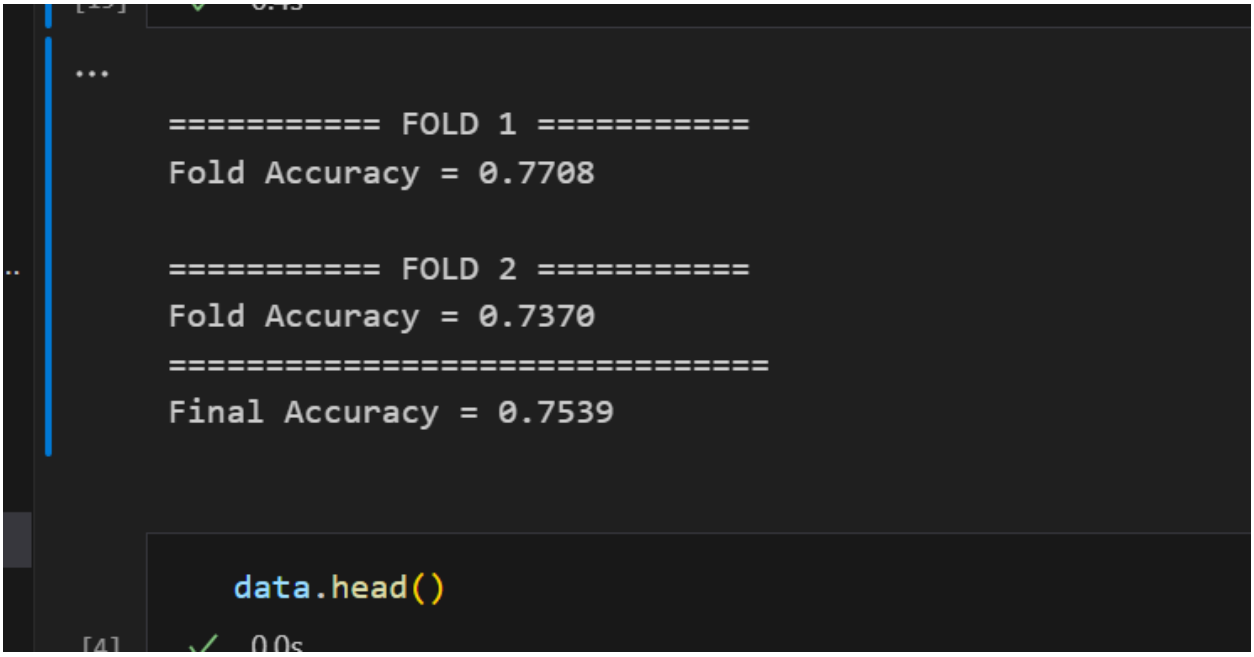
Task2 script:

```
import pandas as pd
import numpy as np

from ask1 import naivebayes



data = pd.read_csv("diabetes.csv")


X = data.drop("Outcome", axis=1).values
y = data["Outcome"].values

X

k = 2
fold_size = len(data) // k

accuracies = []
```

```
for fold in range(k):

    print(f"\n=========== FOLD {fold+1} ===========")

    if fold == 0:
        X_train = X[:fold_size]
        y_train = y[:fold_size]
        X_test  = X[fold_size:]
        y_test  = y[fold_size:]
    else:
        X_test  = X[:fold_size]
        y_test  = y[:fold_size]
        X_train = X[fold_size:]
        y_train = y[fold_size:]


    train_data = {}
    for i in range(len(X_train)):
        train_data[tuple(X_train[i])] = y_train[i]

    # classify
    correct = 0

    for i in range(len(X_test)):
        pred = naive_bayes(train_data, [0, 1], X_test[i])
        if pred == y_test[i]:
            correct += 1

    accuracy = correct / len(X_test)
    accuracies.append(accuracy)

    print(f"Fold Accuracy = {accuracy:.4f}")

print("================================")
print(f"Final Accuracy = {np.mean(accuracies):.4f}")
```

Result:

Syed Shaheer Suhaib CMS ID 429292



Question 3:

Clearly explain the difference between Naïve Bayes and the Multivariate Bayesian Classification Model. Discuss how the results may differ when both models are applied to the same dataset.

Naïve Bayes assumes all features are independent, so it calculates class probabilities by multiplying the individual feature probabilities. The Multivariate Bayesian classifier does not assume independence and instead uses the full covariance matrix, capturing relationships between features. When applied to the same dataset, Naïve Bayes may give simpler but less accurate results if features are correlated, while the Multivariate model usually performs better by modeling those dependencies.