I Pen-and-paper

1.

	y_1	y_2	Class	Classification
x_1	A	0	P	P
x_2	B	1	P	N
x_3	A	1	P	N
x_4	A	0	P	P
x_5	B	0	N	-
x_6	B	0	N	-
x_7	A	1	N	-
x_8	B	1	N	-

x_1)

x_i	x_2	x_3	x_4	x_5	x_6	x_7	x_8
$d(x_1, x_i)$	2.5	1.5	0.5	1.5	1.5	1.5	2.5
Class	P	P	P	N	N	N	N

neighbours = $\{x_3, x_4, x_5, x_6, x_7\}$
P: $\frac{1}{1.5} + \frac{1}{0.5} = 2,\overline{6}$
N: $\frac{1}{1.5} + \frac{1}{1.5} + \frac{1}{1.5} = 2$
$\hat{z}_{x_1} = \text{P} \implies x_1$ is a true positive.

x_2)

x_i	x_1	x_3	x_4	x_5	x_6	x_7	x_8
$d(x_2, x_i)$	2.5	1.5	2.5	1.5	1.5	1.5	0.5
Class	P	P	P	N	N	N	N

neighbours = $\{x_3, x_5, x_6, x_7, x_8\}$
P: $\frac{1}{1.5} = 0,\overline{6}$
N: $\frac{1}{1.5} + \frac{1}{1.5} + \frac{1}{1.5} + \frac{1}{0.5} = 4$
$\hat{z}_{x_2} = \text{N} \implies x_2$ is a false negative.

x_3)

x_i	x_1	x_2	x_4	x_5	x_6	x_7	x_8
$d(x_3, x_i)$	1.5	1.5	1.5	2.5	2.5	0.5	1.5
Class	P	P	P	N	N	N	N

neighbours = $\{x_1, x_2, x_4, x_7, x_8\}$
P: $\frac{1}{1.5} + \frac{1}{1.5} + \frac{1}{1.5} = 2$
N: $\frac{1}{0.5} + \frac{1}{1.5} = 2,\overline{6}$
$\hat{z}_{x_3} = \text{N} \implies x_3$ is a false negative.

x_4)

x_i	x_1	x_2	x_3	x_5	x_6	x_7	x_8
$d(x_4, x_i)$	0.5	2.5	1.5	1.5	1.5	1.5	2.5
Class	P	P	P	N	N	N	N

neighbours = $\{x_1, x_3, x_5, x_6, x_7\}$
P: $\frac{1}{0.5} + \frac{1}{1.5} = 2,\overline{6}$
N: $\frac{1}{1.5} + \frac{1}{1.5} + \frac{1}{1.5} = 2$
$\hat{z}_{x_4} = \text{P} \implies x_4$ is a true positive.

$$\text{recall} = \frac{\text{TP}}{\text{FP} + \text{FN}} = \frac{2}{2+2} = 0.5$$

2. $y_1 \mid \mathrm{P} = \{\mathrm{A}, \mathrm{A}, \mathrm{A}, \mathrm{B}, \mathrm{B}\}$
   $y_1 \mid \mathrm{N} = \{\mathrm{A}, \mathrm{B}, \mathrm{B}, \mathrm{B}\}$
   $y_2 \mid \mathrm{P} = \{0, 0, 0, 1, 1\}$
   $y_2 \mid \mathrm{N} = \{0, 0, 1, 1\}$
   $y_3 \mid \mathrm{P} = \{1.2, 0.8, 0.5, 0.9, 0.8\}$
   $y_3 \mid \mathrm{N} = \{1, 0.9, 1.2, 0.8\}$

$$P(\text{class} = \mathrm{P}) = \frac{5}{9}$$
$$P(\text{class} = \mathrm{N}) = 1 - P(\text{class} = \mathrm{P}) = 1 - \frac{5}{9} = \frac{4}{9}$$
$$P(y_1 = A, y_2 = 0) = \frac{2}{9}$$
$$P(y_1 = A, y_2 = 1) = \frac{2}{9}$$
$$P(y_1 = B, y_2 = 0) = \frac{3}{9}$$
$$P(y_1 = B, y_2 = 1) = \frac{2}{9}$$
$$P(y_1 = A, y_2 = 0 \mid \text{class} = P) = \frac{2}{5}$$
$$P(y_1 = A, y_2 = 1 \mid \text{class} = P) = \frac{1}{5}$$
$$P(y_1 = B, y_2 = 0 \mid \text{class} = P) = \frac{1}{5}$$
$$P(y_1 = B, y_2 = 1 \mid \text{class} = P) = \frac{1}{5}$$
$$P(y_1 = A, y_2 = 0 \mid \text{class} = N) = \frac{0}{4}$$
$$P(y_1 = A, y_2 = 1 \mid \text{class} = N) = \frac{1}{4}$$
$$P(y_1 = B, y_2 = 0 \mid \text{class} = N) = \frac{2}{4}$$
$$P(y_1 = B, y_2 = 1 \mid \text{class} = N) = \frac{1}{4}$$
$$P(y_3) = \mathcal{N}(y_3 \mid \mu, \sigma^2), \text{where } \mu = 0.9 \text{ and } \sigma^2 = 0.0475$$
$$P(y_3 \mid \text{class} = P) = \mathcal{N}(y_3 \mid \mu_\mathrm{P}, \sigma^2_\mathrm{P}), \text{where } \mu_\mathrm{P} = 0.84 \text{ and } \sigma^2_\mathrm{P} = 0.063$$
$$P(y_3 \mid \text{class} = N) = \mathcal{N}(y_3 \mid \mu_\mathrm{N}, \sigma^2_\mathrm{N}), \text{where } \mu_\mathrm{N} = 0.975 \text{ and } \sigma^2_\mathrm{N} = 0.02916667$$

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \times e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$\mu = \frac{\sum_{i=1}^{n} x_i}{n}$$
$$\sigma^2 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})}{n-1}$$

3.

$$P(\text{class} = \text{P} \mid y_1 = \text{A}, y_2 = 1, y_3 = 0.8) =$$

$$= \frac{P(\text{class} = \text{P}) \times P(y_1 = \text{A}, y_2 = 1, y_3 = 0.8 \mid \text{class} = \text{P})}{P(y_1 = \text{A}, y_2 = 1, y_3 = 0.8)} =$$

$$= \frac{P(\text{class} = \text{P}) \times P(y_1 = \text{A}, y_2 = 1 \mid \text{class} = \text{P}) \times P(y_3 = 0.8 \mid \text{class} = \text{P})}{P(y_1 = \text{A}, y_2 = 1) \times P(y_3 = 0.8)} =$$

$$= \frac{\frac{5}{9} \times \frac{1}{5} \times 1.569369}{\frac{2}{9} \times 1.647586} = 0.4762631$$

$$P(\text{class} = \text{P} \mid y_1 = \text{B}, y_2 = 1, y_3 = 1) =$$

$$= \frac{P(\text{class} = \text{P}) \times P(y_1 = \text{B}, y_2 = 1, y_3 = 1 \mid \text{class} = \text{P})}{P(y_1 = \text{B}, y_2 = 1, y_3 = 1)} =$$

$$= \frac{P(\text{class} = \text{P}) \times P(y_1 = \text{B}, y_2 = 1 \mid \text{class} = \text{P}) \times P(y_3 = 1 \mid \text{class} = \text{P})}{P(y_1 = \text{B}, y_2 = 1) \times P(y_3 = 1)} =$$

$$= \frac{\frac{5}{9} \times \frac{1}{5} \times 1.297186}{\frac{2}{9} \times 1.647586} = 0.3936626$$

$$P(\text{class} = \text{P} \mid y_1 = \text{B}, y_2 = 0, y_3 = 0.9) =$$

$$= \frac{P(\text{class} = \text{P}) \times P(y_1 = \text{B}, y_2 = 0, y_3 = 0.9 \mid \text{class} = \text{P})}{P(y_1 = \text{B}, y_2 = 0, y_3 = 0.9)} =$$

$$= \frac{P(\text{class} = \text{P}) \times P(y_1 = \text{B}, y_2 = 0 \mid \text{class} = \text{P}) \times P(y_3 = 0.9 \mid \text{class} = \text{P})}{P(y_1 = \text{B}, y_2 = 0) \times P(y_3 = 0.9)} =$$

$$= \frac{\frac{5}{9} \times \frac{1}{5} \times 1.544655}{\frac{3}{9} \times 1.830473} = 0.2812852$$

4. For $\theta = 0.3$:

   $(y_1 = \text{A}, y_2 = 1, y_3 = 0.8)$ is classified as positive.

   $(y_1 = \text{B}, y_2 = 1, y_3 = 1)$ is classified as positive.

   $(y_1 = \text{B}, y_2 = 0, y_3 = 0.9)$ is classified as negative.

   $\text{accuracy} = \frac{\text{TP+TN}}{\text{All}} = \frac{3}{3} = 1$

   For $\theta = 0.5$:

   $(y_1 = \text{A}, y_2 = 1, y_3 = 0.8)$ is classified as negative

   $(y_1 = \text{B}, y_2 = 1, y_3 = 1)$ is classified as negative

   $(y_1 = \text{B}, y_2 = 0, y_3 = 0.9)$ is classified as negative

   $\text{accuracy} = \frac{\text{TP+TN}}{\text{All}} = \frac{1}{3} = 0.\overline{3}$

   For $\theta = 0.7$:

   $(y_1 = \text{A}, y_2 = 1, y_3 = 0.8)$ is classified as negative

   $(y_1 = \text{B}, y_2 = 1, y_3 = 1)$ is classified as negative

   $(y_1 = \text{B}, y_2 = 0, y_3 = 0.9)$ is classified as negative

   $\text{accuracy} = \frac{\text{TP+TN}}{\text{All}} = \frac{1}{3} = 0.\overline{3}$

   Between the given values, the $\theta = 0.3$ decision threshold optimizes testing accuracy.

II Programming

5.

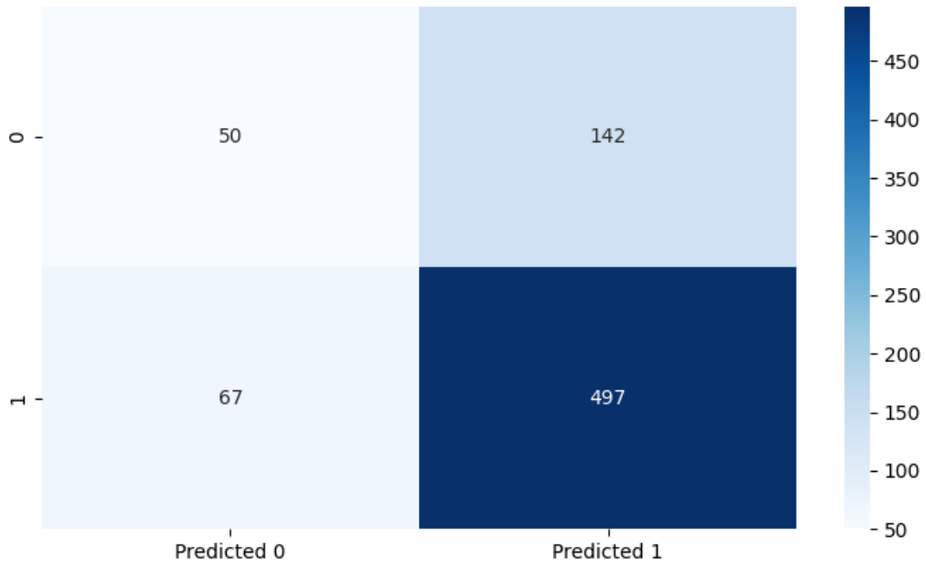

Figure 1: kNN Confusion Matrix

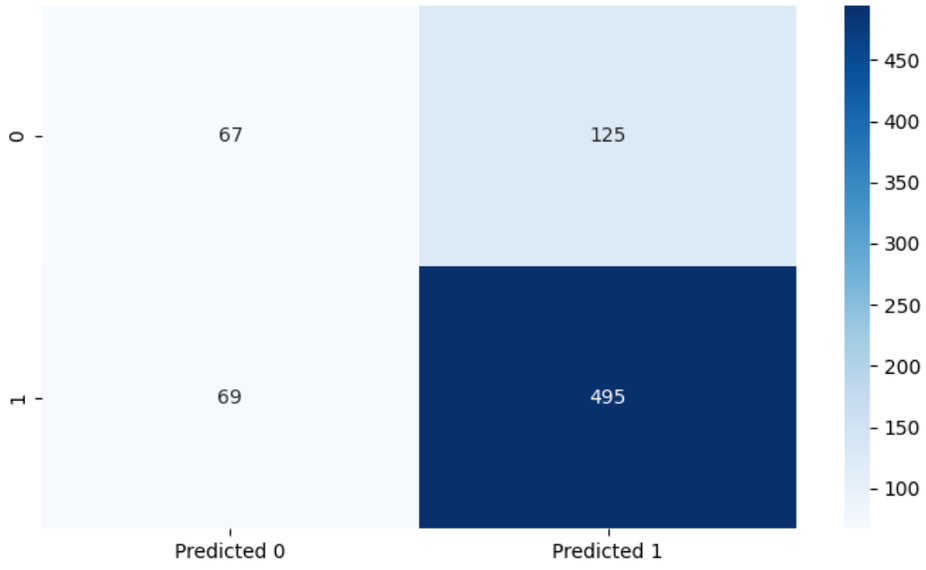

Figure 2: Naïve Bayes Gaussian Confusion Matrix

6. Defining H_0 as "*kNN is statistically superior to Naïve Bayes regarding accuracy*" and H_1 as "*kNN is not statistically superior to Naïve Bayes regarding accuracy*, the hypothesis test lead to a p-value of 0.9104476998751558. Therefore, we reject H_0, concluding that kNN is not statistically superior to Naïve Bayes regarding accuracy.

7. The Naïve Bayes classifier may have had better accuracy results due to the small number of neighbours, k, of the kNN classifier, which is known to have its accuracy increased as the value of k rises. The reduced dependence between the variables of the dataset favors the Naïve Bayes classifier, which assumes variable independence, increasing this classifier's accuracy. Furthermore, the Naïve Bayes classifier is known to have better accuracy, compared to the kNN classifier, when applied to big data.

III Appendix

```python
from scipy.io.arff import loadarff
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import StratifiedKFold
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from scipy import stats

if __name__ == "__main__":
  data = loadarff("pd_speech.arff")
  df = pd.DataFrame(data[0])
  df["class"] = df["class"].astype(int)
  X, y = df.drop("class", axis=1), df["class"]

  classifiers = [KNeighborsClassifier(n_neighbors=5), GaussianNB()]
  matrices = [np.zeros((2, 2)), np.zeros((2, 2))]
  skf = StratifiedKFold(n_splits=10, random_state=0, shuffle=True)
  acc = [[], []]

  for train_index, test_index in skf.split(X, y):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    for i in range(len(classifiers)):
      classifiers[i].fit(X_train, y_train)
      y_pred = classifiers[i].predict(X_test)
      matrices[i] += confusion_matrix(y_test, y_pred)
      acc[i].append(classifiers[i].score(X_test, y_test))

  for matrix in matrices:
    matrix = pd.DataFrame(matrix,
                          index=["0", "1"],
                          columns=["Predicted 0", "Predicted 1"])
    sns.heatmap(matrix, annot=True, fmt="g", cmap="Blues")
    plt.show()

  print(
      "kNN is statistically superior to Naive Bayes regarding accuracy is
   suported by a p-value of {}."
      .format(stats.ttest_rel(acc[0], acc[1], alternative="greater").pvalue))
```