Aprendizagem 2024
Homework I – Group 016
(ist1106022, ist1106720)

Part I: Pen and paper

1.)

- Hamming Distance Values

$d(x_i, x_j)$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	–	2	1	0	1	1	1	2
x_2	2	–	1	2	1	1	1	0
x_3	1	1	–	1	2	2	0	1
x_4	0	2	1	–	1	1	1	2
x_5	1	1	2	1	–	0	2	1
x_6	1	1	2	1	0	–	2	1
x_7	1	1	0	1	2	2	–	1
x_8	2	0	1	2	1	1	1	–

- Nearest Neighbors for each Observation
  - $x_1 \rightarrow x_3$(P)$, x_4$(P)$, x_5$(N)$, x_6$(N)$, x_7$(N)
  - $x_2 \rightarrow x_3$(P)$, x_5$(N)$, x_6$(N)$, x_7$(N)$, x_8$(N)
  - $x_3 \rightarrow x_1$(P)$, x_2$(P)$, x_4$(P)$, x_7$(N)$, x_8$(N)
  - $x_4 \rightarrow x_1$(P)$, x_3$(P)$, x_5$(N)$, x_6$(N)$, x_7$(N)
  - $x_5 \rightarrow x_1$(P)$, x_2$(P)$, x_4$(P)$, x_6$(N)$, x_8$(N)
  - $x_6 \rightarrow x_1$(P)$, x_2$(P)$, x_4$(P)$, x_5$(N)$, x_8$(N)
  - $x_7 \rightarrow x_1$(P)$, x_2$(P)$, x_3$(P)$, x_4$(P)$, x_8$(N)
  - $x_8 \rightarrow x_2$(P)$, x_3$(P)$, x_5$(N)$, x_6$(N)$, x_7$(N)
- Observation Table

$obs.$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
$true$	P	P	P	P	N	N	N	N
$(+)neighbors$	2	1	3	2	3	3	4	2
$(-)neighbors$	3	4	2	3	2	2	1	3
$predicted$	N	N	P	N	P	P	P	N
$result$	FN	FN	TP	FN	FP	FP	FP	TN

- Precision, Recall and F1 Calculations

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{1}{1+3} = 0.25$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{1}{1+3} = 0.25$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.25 \times 0.25}{0.25 + 0.25} = \frac{1}{4} = 0.25$$

2.)

- Improving F1-measure by three fold

If we consider a kNN with k = 3 and using the Hamming Distance for only the first variable (A/B), we get:

$d(x_i, x_j)$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
x_1	–	1	0	0	1	1	0	1
x_2	1	–	1	1	0	0	1	0
x_3	0	1	–	0	1	1	0	1
x_4	0	1	0	–	1	1	0	1
x_5	1	0	1	1	–	0	1	0
x_6	1	0	1	1	0	–	1	0
x_7	0	1	0	0	1	1	–	1
x_8	1	0	1	1	0	0	1	–

- $x_1 \rightarrow x_3(\text{P}), x_4(\text{P}), x_7(\text{N})$
- $x_2 \rightarrow x_5(\text{N}), x_6(\text{N}), x_8(\text{N})$
- $x_3 \rightarrow x_1(\text{P}), x_4(\text{P}), x_7(\text{N})$
- $x_4 \rightarrow x_1(\text{P}), x_3(\text{P}), x_7(\text{N})$
- $x_5 \rightarrow x_2(\text{P}), x_6(\text{N}), x_8(\text{N})$
- $x_6 \rightarrow x_2(\text{P}), x_5(\text{N}), x_8(\text{N})$
- $x_7 \rightarrow x_1(\text{P}), x_3(\text{P}), x_4(\text{P})$
- $x_8 \rightarrow x_2(\text{P}), x_5(\text{N}), x_6(\text{N})$

$obs.$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
$true$	P	P	P	P	N	N	N	N
$(+)neighbors$	2	0	2	2	1	1	3	1
$(-)neighbors$	1	3	1	1	2	2	0	2
$predicted$	P	N	P	P	N	N	P	N
$result$	TP	FN	TP	TP	TN	TN	FP	TN

- Precision, Recall and F1 Calculations

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{3}{3+1} = 0.75$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{3}{3+1} = 0.75$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{0.75 \times 0.75}{0.75 + 0.75} = \frac{3}{4} = 0.75$$

3.)

- Prior Probabilities

$$\text{P(P)} = \frac{5}{9} \qquad \text{P(N)} = \frac{4}{9}$$

- Likelihood of $\{y_1, y_2\}$

  Since $\{y_1\}$ and $\{y_2\}$ are dependant, we calculate the joint possibility for both positive and negative classes.

$$P(y_1 = A, y_2 = 0|P) = \frac{2}{5} \qquad P(y_1 = A, y_2 = 0|N) = 0$$

$$P(y_1 = A, y_2 = 1|P) = \frac{1}{5} \qquad P(y_1 = A, y_2 = 1|N) = \frac{1}{4}$$

$$P(y_1 = B, y_2 = 0|P) = \frac{1}{5} \qquad P(y_1 = B, y_2 = 0|N) = \frac{1}{2}$$

$$P(y_1 = B, y_2 = 1|P) = \frac{1}{5} \qquad P(y_1 = B, y_2 = 1|N) = \frac{1}{4}$$

- Likelihood of $\{y_3\}$

  Assuming $\{y_3\}$ is normally distributed.

$$\mu_P = \frac{1.1 + 0.8 + 0.5 + 0.9 + 0.8}{5} = 0.82$$

$$\sigma_P^2 = \frac{(1.1 - 0.82)^2 + (0.8 - 0.82)^2 + (0.5 - 0.82)^2 + (0.9 - 0.82)^2 + (0.8 - 0.82)^2}{5} = 0.0376$$

$$y_3|P \sim N(0.82, 0.0376)$$

$$\mu_N = \frac{1 + 0.9 + 1.2 + 0.9}{4} = 1.0$$

$$\sigma_N^2 = \frac{(1 - 1.0)^2 + (0.9 - 1.0)^2 + (1.2 - 1.0)^2 + (0.9 - 1.0)^2}{4} = 0.015$$

$$y_3|N \sim N(1.0, 0.015)$$

- Bayesian Classifier

$$P(y_3 = x|P) = \frac{1}{\sqrt{2\pi \cdot 0.0376}} e^{\frac{-1}{2} \cdot (\frac{(x-0.82)^2}{0.0376})}$$

$$P(y_3 = x|N) = \frac{1}{\sqrt{2\pi \cdot 0.015}} e^{\frac{-1}{2} \cdot (\frac{(x-1)^2}{0.015})}$$

4.)

- For (A, 1, 0.8)

$$P(P \mid y_1 = A, y_2 = 1, y_3 = 0.8) = P(P) \cdot p(y_1 = A, y_2 = 1 \mid P) \cdot p(y_3 = 0.8 \mid P)$$

$$= \frac{5}{9} \cdot \frac{1}{5} \cdot 2.0465 = 0.227$$

$$P(N \mid y_1 = A, y_2 = 1, y_3 = 0.8) = P(N) \cdot p(y_1 = A, y_2 = 1 \mid N) \cdot p(y_3 = 0.8 \mid N)$$

$$= \frac{4}{9} \cdot \frac{1}{4} \cdot 0.8586 = 0.095$$

Since $P(P \mid y_1 = A, y_2 = 1, y_3 = 0.8) > P(N \mid y_1 = A, y_2 = 1, y_3 = 0.8)$, we classify (A, 1, 0.8) as Positive (P).

- For (B, 1, 1)

$$P(P \mid y_1 = B, y_2 = 1, y_3 = 1) = P(P) \cdot p(y_1 = B, y_2 = 1 \mid P) \cdot p(y_3 = 1 \mid P)$$

$$= \frac{5}{9} \cdot \frac{1}{5} \cdot 1.337 = 0.1486$$

$$P(N \mid y_1 = B, y_2 = 1, y_3 = 1) = P(N) \cdot p(y_1 = B, y_2 = 1 \mid N) \cdot p(y_3 = 1 \mid N)$$

$$= \frac{4}{9} \cdot \frac{1}{4} \cdot 3.257 = 0.3619$$

Since $P(P \mid y_1 = B, y_2 = 1, y_3 = 1) < P(N \mid y_1 = B, y_2 = 1, y_3 = 1)$, we classify (B, 1, 1) as Negative (N).

- For (B, 0, 0.9)

$$P(P \mid y_1 = B, y_2 = 0, y_3 = 0.9) = P(P) \cdot p(y_1 = B, y_2 = 0 \mid P) \cdot p(y_3 = 0.9 \mid P)$$

$$= \frac{5}{9} \cdot \frac{1}{5} \cdot 1.8895 = 0.2099$$

$$P(N \mid y_1 = B, y_2 = 0, y_3 = 0.9) = P(N) \cdot p(y_1 = B, y_2 = 0 \mid N) \cdot p(y_3 = 0.9 \mid N)$$

$$= \frac{4}{9} \cdot \frac{1}{2} \cdot 2.333 = 0.518$$

Since $P(P \mid y_1 = B, y_2 = 0, y_3 = 0.9) < P(N \mid y_1 = B, y_2 = 0, y_3 = 0.9)$, we classify (B, 0, 0.9) as Negative (N).

5.)

- Total Terms in Classes
  - $Amazing \to 1$
  - $run \to 1$
  - $I \to 1$
  - $like \to 1$
  - $it \to 1$

$$N_P = 1 + 1 + 1 + 1 + 1 = 5$$

  - $Too \to 1$
  - $tired \to 1$
  - $Bad \to 1$
  - $run \to 1$

$$N_N = 1 + 1 + 1 + 1 = 4$$

  - $Amazing \to 1$
  - $run \to 2$

  - $I \rightarrow 1$
  - $like \rightarrow 1$
  - $it \rightarrow 1$
  - $Too \rightarrow 1$
  - $tired \rightarrow 1$
  - $Bad \rightarrow 1$

$$N_N = 8$$

- Likelihoods Calculations
  Using the formula:

$$p(t_i \mid c) = \frac{freq(t_i) + 1}{N_c + V}$$

Positive Class (P):
Term: "I":

$$p(\text{"}I\text{"} \mid P) = \frac{1+1}{5+8} = \frac{2}{13}$$

Term: "like":

$$p(\text{"}like\text{"} \mid P) = \frac{1+1}{5+8} = \frac{2}{13}$$

Term: "to":

$$p(\text{"}to\text{"} \mid P) = \frac{0+1}{5+8} = \frac{1}{13}$$

Term: "run":

$$p(\text{"}run\text{"} \mid P) = \frac{1+1}{5+8} = \frac{2}{13}$$

Negative Class (N):
Term: "I":

$$p(\text{"}I\text{"} \mid N) = \frac{0+1}{4+8} = \frac{1}{12}$$

Term: "like":

$$p(\text{"}like\text{"} \mid N) = \frac{0+1}{4+8} = \frac{1}{12}$$

Term: "to":

$$p(\text{"}to\text{"} \mid N) = \frac{0+1}{4+8} = \frac{1}{12}$$

Term: "run":

$$p(\text{"}run\text{"} \mid N) = \frac{1+1}{4+8} = \frac{2}{12} = \frac{1}{6}$$

- Classifying "I like to run"
  Prior Probabilities:

$$P(P) = P(N) = \frac{1}{2}$$

For Positive Class (P):

$$P(P \mid \text{"I like to run"}) = P(P) \cdot p(\text{"}I\text{"} \mid P) \cdot p(\text{"}like\text{"} \mid P) \cdot p(\text{"}to\text{"} \mid P) \cdot p(\text{"}run\text{"} \mid P)$$

$$= \frac{1}{2} \cdot \frac{2}{13} \cdot \frac{2}{13} \cdot \frac{1}{13} \cdot \frac{2}{13}$$
$$= \frac{1}{2} \cdot \frac{8}{28561} = \frac{4}{28561} \approx 0.00014$$

For Negative Class (N):

$$P(N \mid \text{"I like to run"}) = P(N) \cdot p(\text{"}I\text{"} \mid N) \cdot p(\text{"}like\text{"} \mid N) \cdot p(\text{"}to\text{"} \mid N) \cdot p(\text{"}run\text{"} \mid N)$$
$$= \frac{1}{2} \cdot \frac{1}{12} \cdot \frac{1}{12} \cdot \frac{1}{12} \cdot \frac{1}{6}$$
$$= \frac{1}{2} \cdot \frac{1}{20736} = \frac{1}{20736} \approx 0.000048$$

Conclusion: Since $P(P \mid \text{"I like to run"}) > P(N \mid \text{"I like to run"})$, we classify the sentence "I like to run" as Positive (P).

Part II: Programming

1.) To compare the performance of a *kNN* classifier with $k = 5$ and a naive Bayes classifier, a 5-fold stratified cross-validation was performed on the on a heart disease dataset:

a.) For comparing accuracies of the two classifiers, a box plot was generated for each:

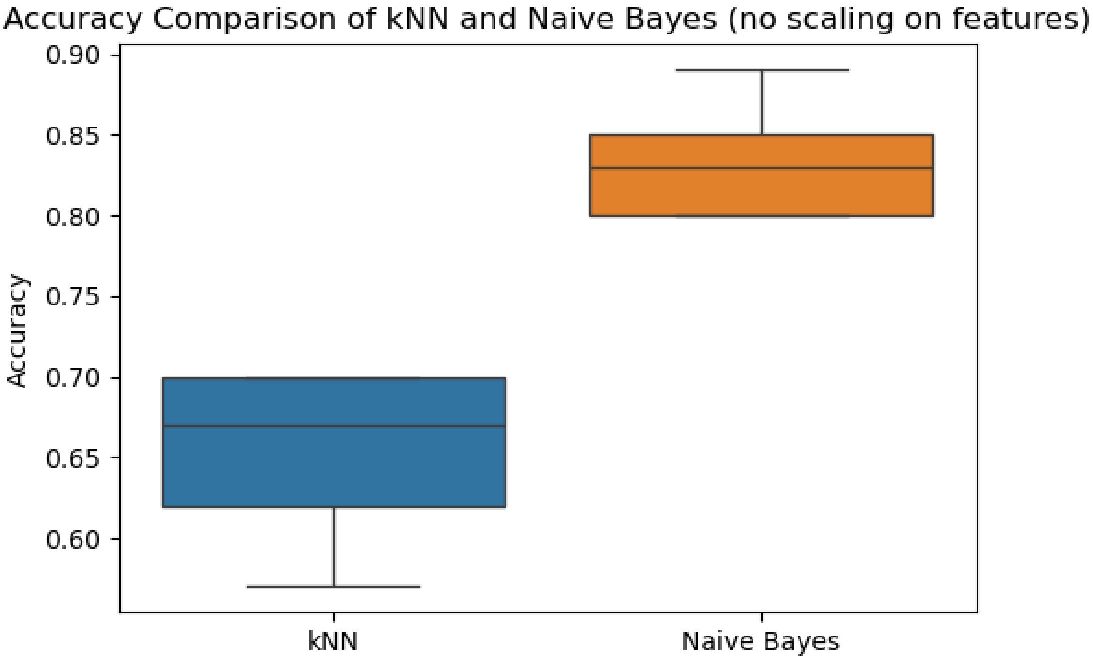

Figure 1: Boxplot graphs for the accuracies of the *kNN* and Naive Bayes classifiers, no scaling applied on feature data

The performance of the *kNN* is less consistent as the box plot is wider, while the Naive Bayes classifier has a more consistent performance, as the box plot is narrower.

This is due to the fact that the *kNN* is non-parametric and therefore is more sensitive across folds, and with non-scaled data, the distance between points is not taken into account, leading to a less accurate and more variable classification, with some classes being more dominant than others, for example.

b.) However, when choosing to scale the feature data with a Min-Max Scaler, the box plots look as follows:

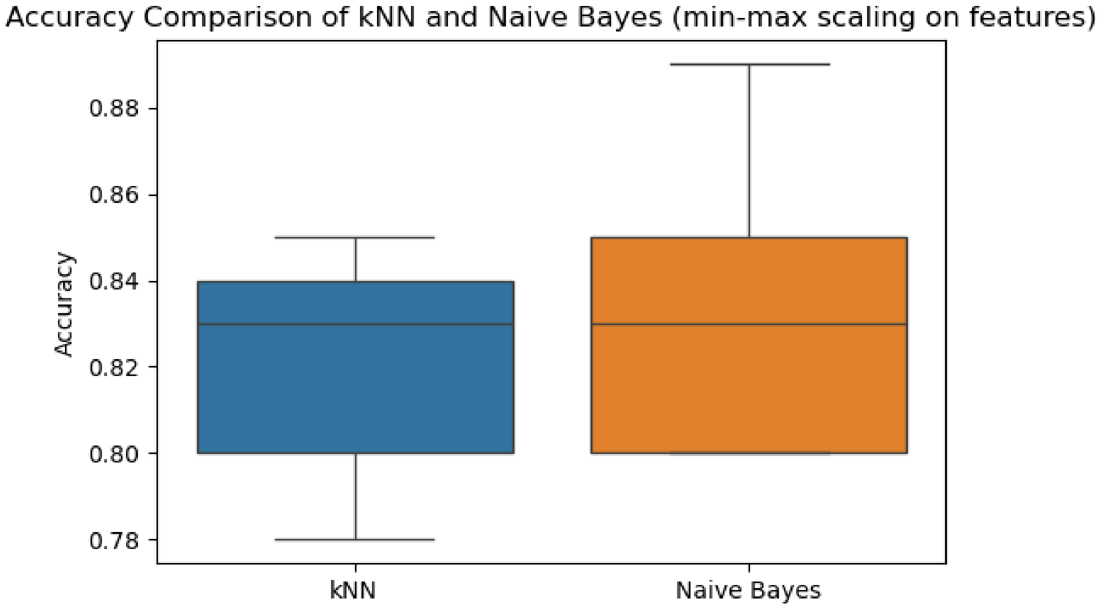

Figure 2: Boxplot graphs for the accuracies of the *kNN* and Naive Bayes classifiers, scaling data with Min-Max Scaler

This happens because the *kNN* classifier, when choosing to scale feature data, the distance starts becoming relevant and variability is heavily reduced, since no class imposes dominant on others, for example. This leads to a more consistent and accurate classification.

For *Naive Bayes*, the scaling of the data does not generally affect the performance, since it is a parametric model.

c.) Perfoming a paired t-test on the accuracies of the two classifiers, using `scipy`'s method `ttest_rel` that considers H_0 to be the hypothesis that the two classifiers have the same stastical significance regarding accuracy, and H_1 to be the hypothesis that *kNN* is more accurate than Naive Bayes, the results are as follows:

- When not scaling feature data: P-value = 0.998415501126768
  - Considering a 1% threshold, *kNN* is not statistically superior than Bayes
  - Considering a 5% threshold, *kNN* is not statistically superior than Bayes
  - Considering a 10% threshold, *kNN* is not statistically superior than Bayes
- When min-max scaling feature data: P-value = 0.7532332545792753
  - Considering a 1% threshold, *kNN* is not statistically superior than Bayes
  - Considering a 5% threshold, *kNN* is not statistically superior than Bayes
  - Considering a 10% threshold, *kNN* is not statistically superior than Bayes

We can therefore easily conclude that the hypothesis ”the *kNN* model is statistically superior to naïve Bayes regarding accuracy” is **false / rejected**.

2.) To compare the performance of uniform and distance-based weights *kNN* classifiers with varying amounts of neighbors (k) used in classifications, a 80-20 train-test split was performed for each combination:

a.) The following plots showcase the obtained results:

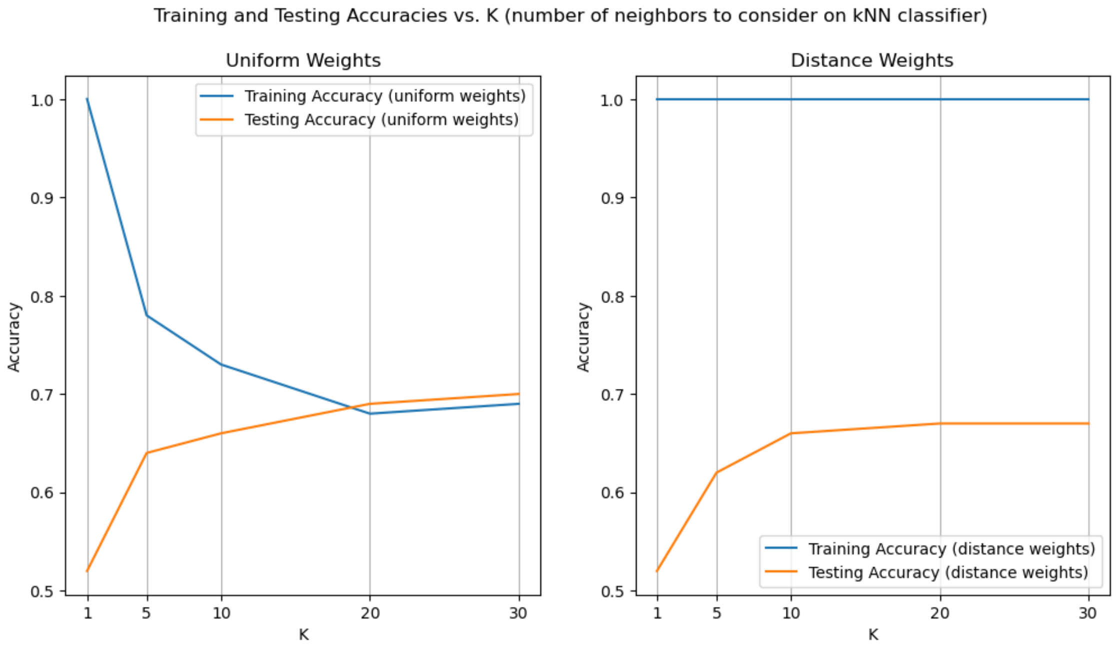

Figure 3: Accuracy plots for the *kNN* classifier with uniform and distance-based weights

b.) Generally, the bigger the value of k on a *kNN* classifier, the more accurate the predictions are, until a certain point. This point, for either uniform or distance-based weights, is when the accuracies on both training and testing data is the highest, which seems to be around $k = 20$ for this dataset.

In terms of generalization capabilities, the distance-based weighting system seems to be the more capable of the both, as it is capable of maintaing its accuracy on training data while also improving its accuracy on the testing data.

Before the suggested point, for the uniform weights, the model is overfitting on the training data. Past $k = 30$ both models may be at the risk of underfitting, a phenomenon not observable for the dataset at hand.

3.) Some properties from the dataset that may justify the shortcomings of Naive Bayes classifier on this dataset are:

- Since Naive Bayes assumes that all features are independent, it ignores the relantionships and correlation that these features may have in relation to heart disease diagnosis.
- The dataset has features which are not normally distributed or are not numerical, which is a violation of the Gaussian Naive Bayes conditions. Some of these features include the categorical features, such as cp, fbs, $resteg$, $exang$ and $slope$, for example.