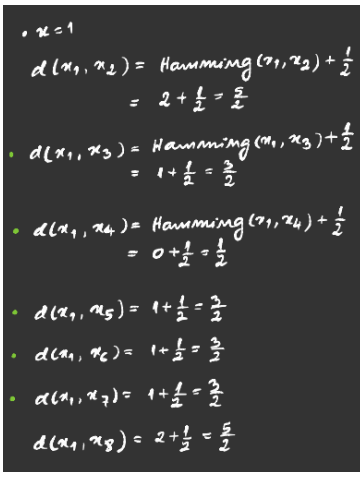
TÉCNICO LISBOA

Aprendizagem 2022/23

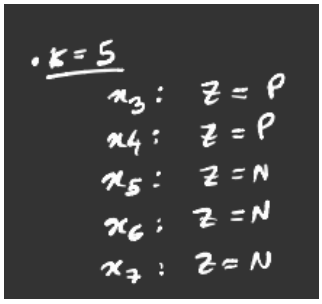
Homework II – Group 104

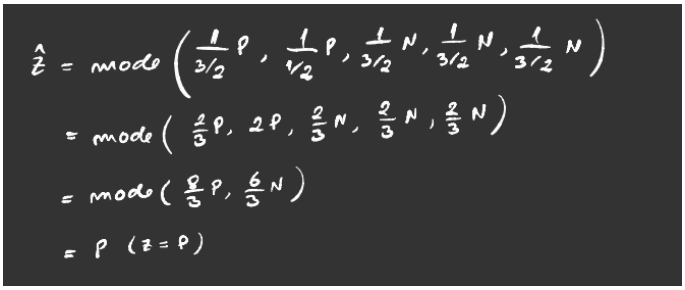
I. Pen-and-paper

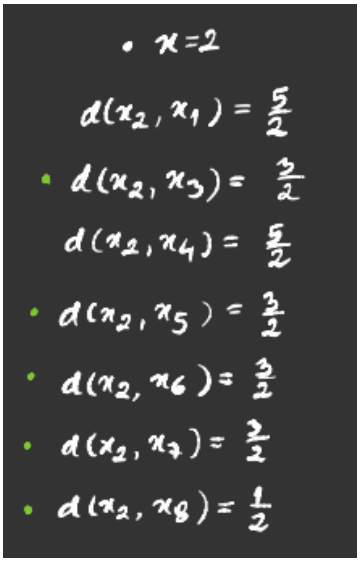
1)



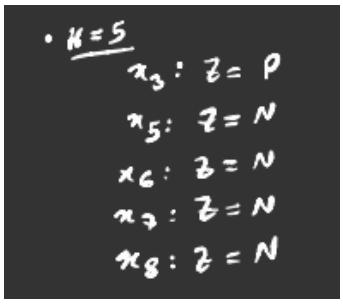
x = 1

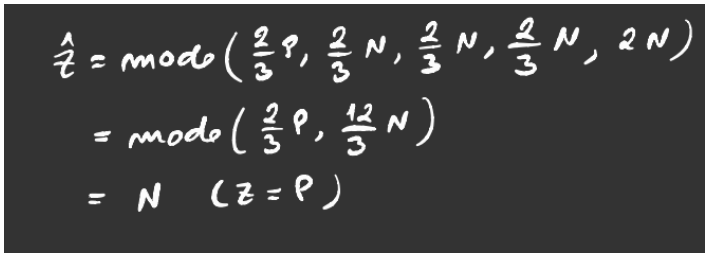


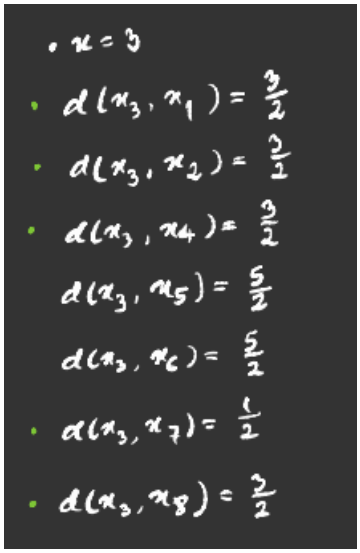




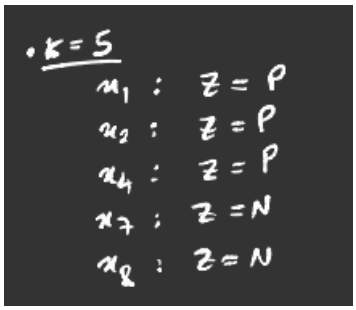
x = 2

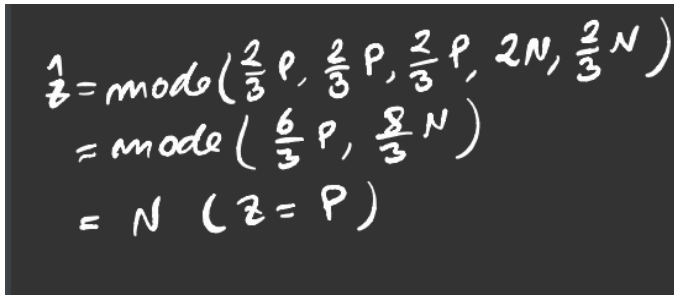






x = 3





TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 104

x = 4

- $x = 4$
- $d(x_4, x_1) = \frac{1}{2}$
- $d(x_4, x_2) = \frac{5}{2}$
- $d(x_4, x_3) = \frac{3}{2}$
- $d(x_4, x_5) = \frac{3}{2}$
- $d(x_4, x_6) = \frac{3}{2}$
- $d(x_4, x_7) = \frac{3}{2}$
- $d(x_4, x_8) = \frac{5}{2}$

- $k = 5$
  - $x_1 : z = P$
  - $x_3 : z = P$
  - $x_5 : z = N$
  - $x_6 : z = N$
  - $x_7 : z = N$

$\hat{z} = mode(2P, \frac{2}{3}P, \frac{2}{3}N, \frac{2}{3}N, \frac{2}{3}N)$
$= mode(\frac{8}{3}P, \frac{5}{3}N)$
$= P \ (z = P)$

x = 5

- $x = 5$
- $d(x_5, x_1) = \frac{3}{2}$
- $d(x_5, x_2) = \frac{3}{2}$
- $d(x_5, x_3) = \frac{5}{2}$
- $d(x_5, x_4) = \frac{3}{2}$
- $d(x_5, x_6) = \frac{1}{2}$
- $d(x_5, x_7) = \frac{5}{2}$
- $d(x_5, x_8) = \frac{3}{2}$

- $k = 5$
  - $x_1 : z = P$
  - $x_2 : z = P$
  - $x_4 : z = P$
  - $x_6 : z = N$
  - $x_8 : z = N$

$\hat{z} = mode(\frac{2}{3}P, \frac{2}{3}P, \frac{2}{3}P, 2N, \frac{2}{3}N)$
$= mode(\frac{6}{3}P, \frac{8}{3}N)$
$= N \ (z = N)$

x = 6

- $x = 6$
- $d(x_6, x_1) = \frac{3}{2}$
- $d(x_6, x_2) = \frac{3}{2}$
- $d(x_6, x_3) = \frac{5}{2}$
- $d(x_6, x_4) = \frac{3}{2}$
- $d(x_6, x_5) = \frac{1}{2}$
- $d(x_6, x_7) = \frac{5}{2}$
- $d(x_6, x_8) = \frac{3}{2}$

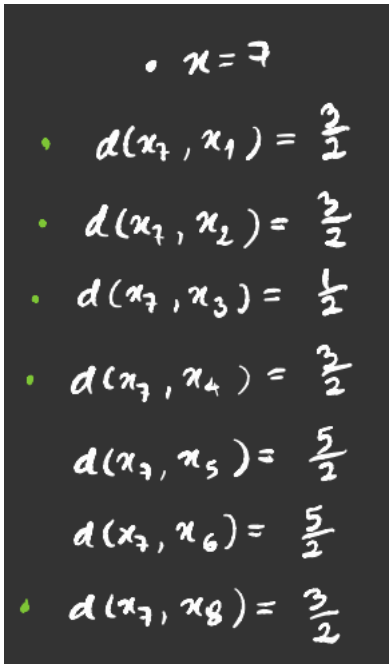
- $k = 5$
  - $x_1 : z = P$
  - $x_2 : z = P$
  - $x_4 : z = P$
  - $x_5 : z = N$
  - $x_8 : z = N$

$\hat{z} = mode(\frac{2}{3}P, \frac{2}{3}P, \frac{2}{3}P, 2N, \frac{2}{3}N)$
$= mode(\frac{6}{3}P, \frac{8}{3}N)$
$= N \ (z = N)$

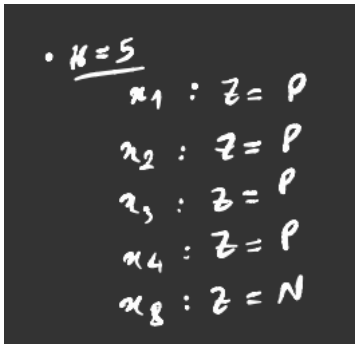
TÉCNICO LISBOA

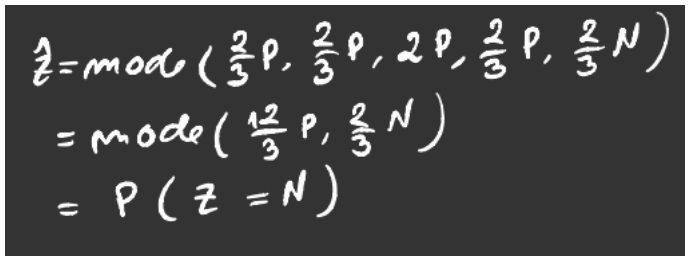
Aprendizagem 2022/23

Homework II – Group 104



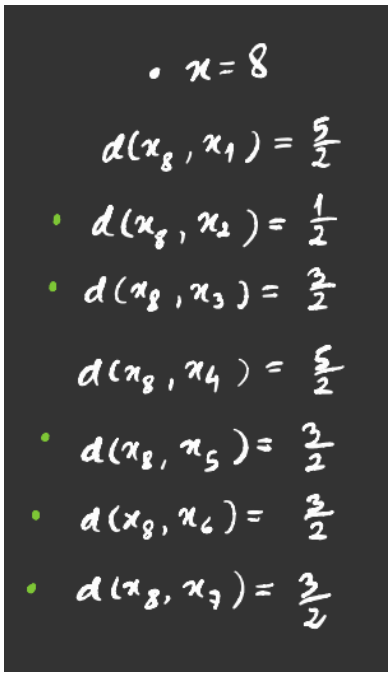
x = 7



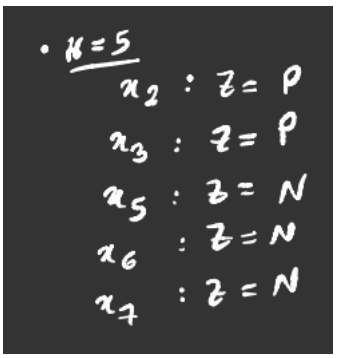


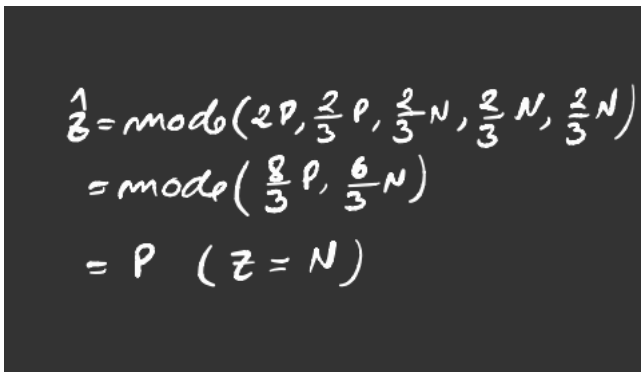
Auxiliar for all the calculations:

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

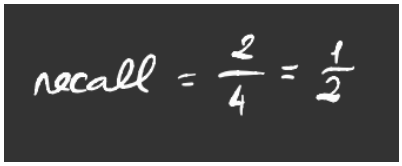


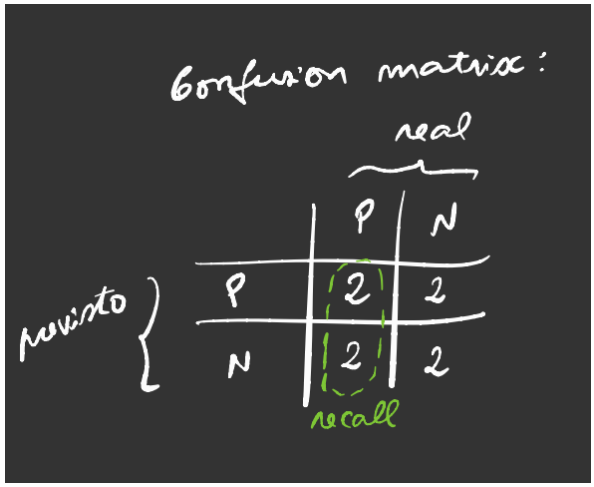
x = 8





Answer:



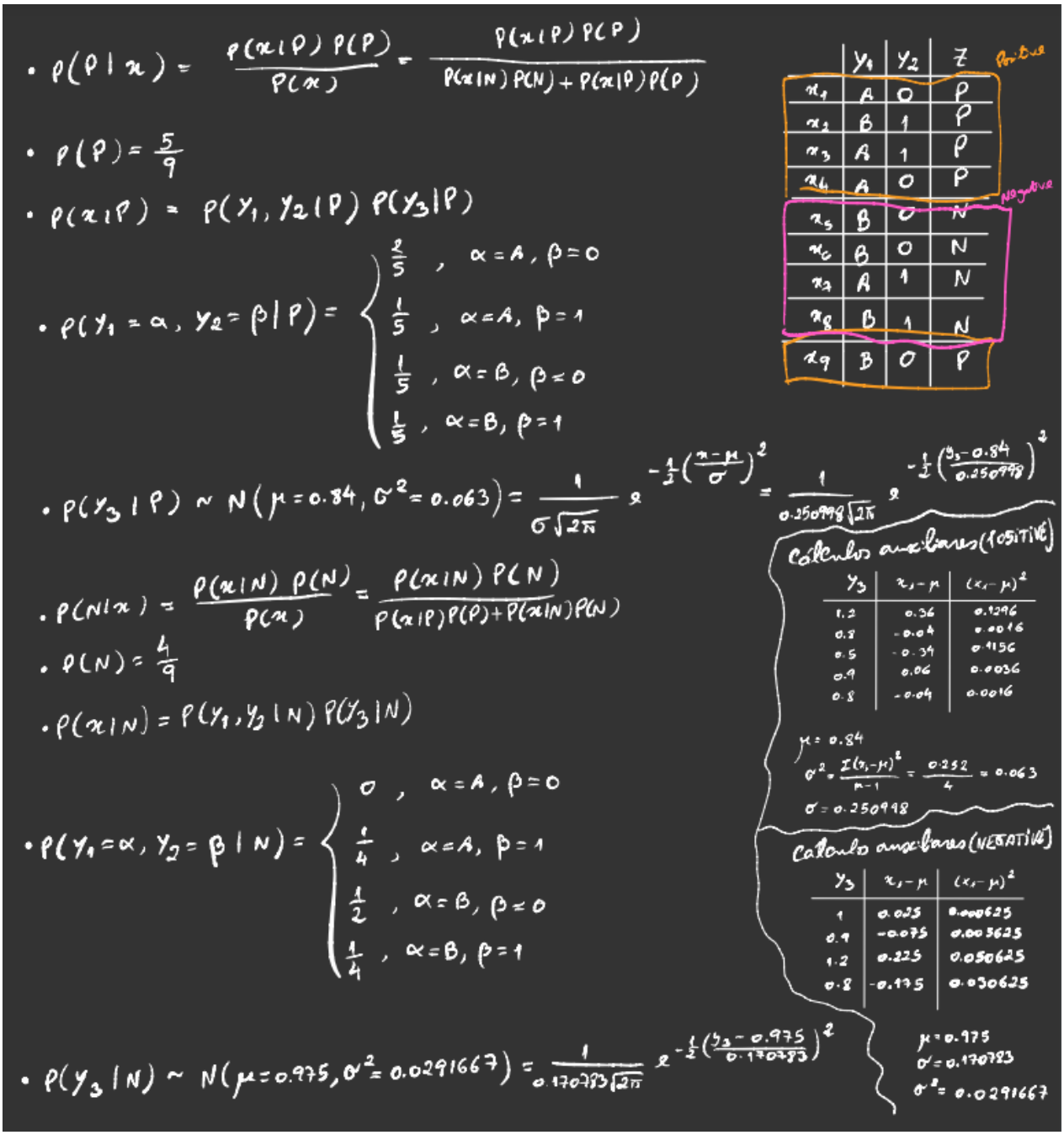


TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 104

2)



	Y_1	Y_2	Z
x_1	A	0	P
x_2	B	1	P
x_3	A	1	P
x_4	A	0	P
x_5	B	0	N
x_6	B	0	N
x_7	A	1	N
x_8	B	1	N
x_9	B	0	P

Y_3	$x_i-\mu$	$(x_i-\mu)^2$
1.2	0.36	0.1296
0.8	-0.04	0.0016
0.5	-0.34	0.1156
0.9	0.06	0.0036
0.8	-0.04	0.0016

Y_3	$x_i-\mu$	$(x_i-\mu)^2$
1	0.025	0.000625
0.9	-0.075	0.005625
1.2	0.225	0.050625
0.8	-0.175	0.030625

3)

MAP : $\underset{\text{positive}}{\text{argmax}}\ P(\text{Positive} \mid x) = \underset{\text{positive}}{\text{argmax}}\ \frac{P(x \mid \text{Positive})\, P(\text{Positive})}{P(x)}$

De acordo com os parâmetros obtidos na alínea anterior, calcularam-se as seguintes probabilidades:

	y_1	y_2	y_3	z
x_1	A	1	0.8	Positive
x_2	B	1	1	Positive
x_3	B	0	0.9	Negative

$$P(P \mid x_1) = \frac{P(x_1 \mid P)\, P(P)}{P(x_1 \mid P)\, P(P) + P(x_1 \mid N)\, P(N)} = \frac{P(y_1 = A, y_2 = 1 \mid P)\, P(y_3 = 0.8 \mid P)\, P(P)}{P(y_1 = A, y_2 = 1 \mid P)\, P(y_3 = 0.8 \mid P)\, P(P) + P(y_1 = A, y_2 = 1 \mid N)\, P(y_3 = 0.8 \mid N)\, P(N)}$$

$$\approx 0.531769$$

$$P(P \mid x_2) = \frac{P(x_2 \mid P)\, P(P)}{P(x_2 \mid P)\, P(P) + P(x_2 \mid N)\, P(N)} = \frac{P(y_1 = B, y_2 = 1 \mid P)\, P(y_3 = 1 \mid P)\, P(P)}{P(y_1 = B, y_2 = 1 \mid P)\, P(y_3 = 1 \mid P)\, P(P) + P(y_1 = B, y_2 = 1 \mid N)\, P(y_3 = 1 \mid N)\, P(N)}$$

$$\approx 0.359505$$

$$P(P \mid x_3) = \frac{P(x_3 \mid P)\, P(P)}{P(x_3 \mid P)\, P(P) + P(x_3 \mid N)\, P(N)} = \frac{P(y_1 = B, y_2 = 0 \mid P)\, P(y_3 = 0.9 \mid P)\, P(P)}{P(y_1 = B, y_2 = 0 \mid P)\, P(y_3 = 0.9 \mid P)\, P(P) + P(y_1 = B, y_2 = 0 \mid N)\, P(y_3 = 0.9 \mid N)\, P(N)}$$

$$\approx 0.266913$$

In conclusion:

$$P(P \mid x_1) = 0.531769$$

$$P(P \mid x_2) = 0.359505$$

$$P(P \mid x_3) = 0.266913$$

Aprendizagem 2022/23

Homework II – Group 104

4)

De acordo com os valores obtidos na alínea anterior,

- $\theta = 0.5$

  $f(x_1|\theta) =$ Positive
  $f(x_2|\theta) =$ Negative
  $f(x_3|\theta) =$ Negative
  } accuracy $= \frac{2}{3}$

- $\theta = 0.7$

  $f(x_1|\theta) =$ Negative
  $f(x_2|\theta) =$ Negative
  $f(x_3|\theta) =$ Negative

  accuracy $= \frac{1}{3}$

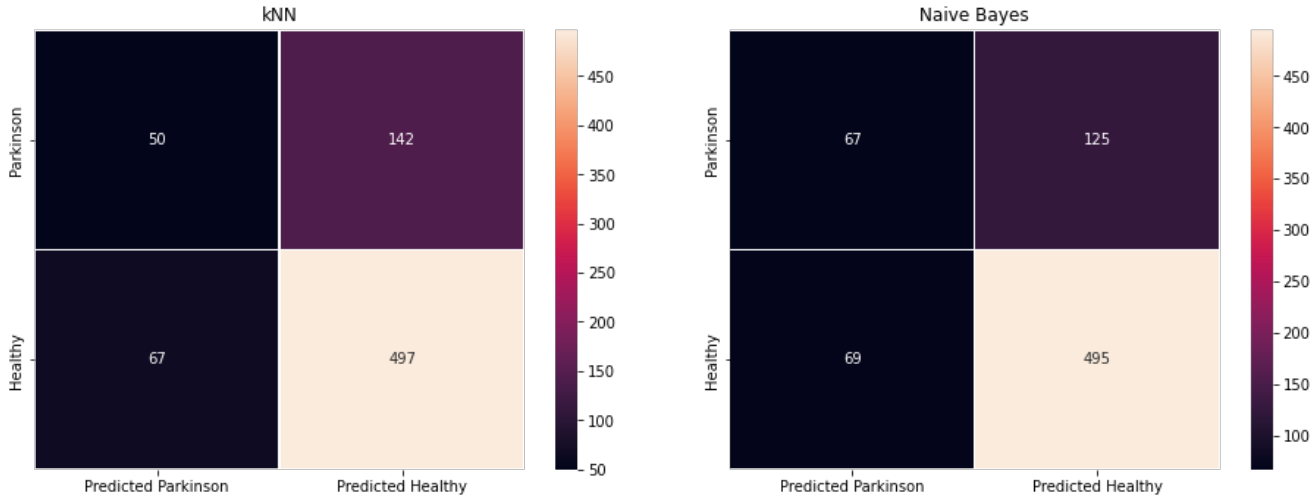
- $\theta = 0.3$

  $f(x_1|\theta) =$ Positive
  $f(x_2|\theta) =$ Positive
  $f(x_3|\theta) =$ Negative
  } accuracy $= 1$

Logo, $\theta = 0.3$ é o valor do decision threshold para o qual a accuracy é otimizada.

II. Programming and critical analysis

5)



TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 104

6)

Hypothesis:
H0 (null hypothesis): there is no statistical difference between kNN and Naïve Bayes regarding accuracy.
H1: kNN is statistically superior to Naïve Bayes regarding accuracy. (p1>p2?)

Results:
The result p-value, after using scipy to test H1, is the following:
p1>p2? pval= 0.9104578404340254

Conclusion:
We do not reject the null hypothesis, for statistical significance of 5%, because pval > 0.05. Therefore, there is no statistical significance between kNN and Naïve Bayes, regarding accuracy. Having that in mind, we can infer that the statement is not true.

Bellow, we present the code used to produce the results:

```
pred_1, pred_2 = acc_folds["kNN"], acc_folds["Naive Bayes"]
# predictor 1 is better than 2?
res = stats.ttest_rel(pred_1, pred_2, alternative='greater')
print("p1>p2? pval=",res.pvalue)

# predictor 2 is better than 1?
res = stats.ttest_rel(pred_1, pred_2, alternative='less')
print("p1<p2? pval=",res.pvalue)

# performance of predictor 1 differs from predictor 2?
res = stats.ttest_rel(pred_1, pred_2, alternative='two-sided')
print("p1!=p2? pval=",res.pvalue)
```

Output:
p1>p2? pval= 0.9104578404340254
p1<p2? pval= 0.08954215956597458
p1!=p2? pval= 0.17908431913194917

TÉCNICO LISBOA

Aprendizagem 2022/23

Homework II – Group 104

7)

From 6), we couldn't reject H0. Moreover, when testing the opposite, that is, the H1 hypothesis that Naïve Bayes is statistically superior to kNN regarding accuracy, we obtained a pvalue of 0.0895 which, in this case, allows us to reject the null hypothesis for a significance of 10%. When we conjugate this, with the fact that the mean accuracy of Naïve Bayes (0.74) is greater than the mean accuracy of kNN (0.72), evidence points towards Naïve Bayes performing slightly better than kNN.

That could be explained by the fact that Naïve Bayes is not affected by high dimensional datasets (large number of attributes), while kNN can be - 752 features in our dataset.

Naïve Bayes is a linear classifier, while kNN is a distance-based classifier, which classifies data based on proximity to K-neighbors. However, the scale of the different dataset features could vary, especially for this dataset, which has many numeric features, causing kNN to perform sub optimally. To sum up this point, kNN could be more affected by not normalizing the data than Naïve Bayes, giving the latter a slightly edge regarding accuracy.

Another point could be the fact that kNN's number of neighbors (k) is too small - as we know, the error rate of kNN decreases by increasing the value of k. Following this idea and having in mind that hyperparameters can influence the accuracy of a model, Naïve Bayes has two hyperparameters to tune for smoothing – alpha and beta. Contrarily, kNN has only one – k, which gives a finer tuning and control to the former model.

On the other hand, Naïve Bayes has its own downsides too: kNN does not assume independence between variables, contrarily to Naïve Bayes. Therefore, for datasets with high dimensionality, like the one from the homework, the probability of existing dependence between variables on the dataset increases which, consequently, could decrease Naïve Bayes accuracy, as this model would be wrongly assuming the independence of those variables.

III. APPENDIX

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

raw_data = loadarff('pd_speech.arff.txt')
df = pd.DataFrame(raw_data[0])
df['class'] = df['class'].str.decode('utf-8')
X, y = df.drop("class", axis=1), df['class']

classifiers = [
    ("kNN", KNeighborsClassifier()),
    ("Naive Bayes", GaussianNB())
]

# b) manual fold iteration
knn_y_test, knn_y_pred = [], []
naive_bayes_y_test, naive_bayes_y_pred = [], []
y_values = {"kNN": {"y_test": [], "y_pred": []},
            "Naive Bayes": {"y_test": [], "y_pred": []}}
acc_folds = {"kNN" : [], "Naive Bayes": []}
folds = StratifiedKFold(n_splits=10, shuffle=True, random_state=0)

# iterate per fold
for train_k, test_k in folds.split(X, y):
    X_train, X_test = X.iloc[train_k], X.iloc[test_k]
    y_train, y_test = y.iloc[train_k], y.iloc[test_k]

    for name, classifier in classifiers:

      predictor = classifier
      # train and assess
      predictor.fit(X_train, y_train)
      y_pred = predictor.predict(X_test)
```

```
      y_values[name]["y_test"] += list(y_test.astype(int))
      y_values[name]["y_pred"] += list(y_pred.astype(int))

      acc_folds[name].append(round(metrics.accuracy_score(y_test, y_pred),5))

cm_knn = np.array(confusion_matrix(y_values["kNN"]["y_test"],
                                   y_values["kNN"]["y_pred"], labels=[0,1]))
cm_naive_bayes = np.array(confusion_matrix(y_values["Naive Bayes"]["y_test"],
                                   y_values["Naive Bayes"]["y_pred"],
                                   labels=[0,1]))
confusion_knn = pd.DataFrame(cm_knn,
                             index=['Parkinson', 'Healthy'],
                             columns=['Predicted Parkinson', 'Predicted Healthy'])
confusion_naive_bayes = pd.DataFrame(cm_naive_bayes,
                                     index=['Parkinson', 'Healthy'],
                                     columns=['Predicted Parkinson', 'Predicted Healthy'])

fig, (ax1, ax2) = plt.subplots(ncols=2, nrows=1, figsize=(14, ))
g1 = sns.heatmap(confusion_knn, annot=True, fmt='g', linewidths=.5, ax=ax1)
g2 = sns.heatmap(confusion_naive_bayes, annot=True, fmt='g', linewidths=.5, ax=ax2)
ax1.set_title("kNN")
ax2.set_title("Naive Bayes")
g1.set_xticklabels(g1.get_xticklabels(), rotation = 0)
g2.set_xticklabels(g2.get_xticklabels(), rotation = 0)
fig.tight_layout(pad=5.0)
plt.show()

print("Fold accuracies:", acc_folds)

pred_1, pred_2 = acc_folds["kNN"], acc_folds["Naive Bayes"]
# predictor 1 is better than 2?
res = stats.ttest_rel(pred_1, pred_2, alternative='greater')
print("p1>p2? pval=",res.pvalue)

# predictor 2 is better than 1?
res = stats.ttest_rel(pred_1, pred_2, alternative='less')
print("p1<p2? pval=",res.pvalue)

# performance of predictor 1 differs from predictor 2?
res = stats.ttest_rel(pred_1, pred_2, alternative='two-sided')
print("p1!=p2? pval=",res.pvalue)
```

END