IS-733 CLASSWORK – Understanding Naive Bayes and K-nearest neighbors – UB01976

1a. Manually calculate prediction using **the Naive Bayes Model and K nearest neighbor** for the test example for the following example:

ID	Contains Link	Contains Money Words	Length	Class
1	Yes	Yes	Long	Spam
2	No	No	Short	Ham
3	Yes	No	Long	Spam
4	No	Yes	Short	Spam
5	Yes	Yes	Short	Spam
6	No	No	Long	Ham
7	Yes	No	Short	Ham
8	No	Yes	Long	Spam
9	Yes	Yes	Long	Spam
10	No	No	Short	Ham

1a Ans: - Prediction using the Naive Bayes Model

To calculate the predictability using Naive Bayes, we convert the categorial values into numerical values for calculations as follows;

Contains Link: - if it is YES =1, NO = 0

Contains Money Words: - if it is YES =1, NO = 0

Length: - if it is Long = 1, Short = 0

Class: - if it is Spam = 1, Ham = 0

Based on these numerical values we rearrange the values in the example, and the numerical values table would be as follows;

ID	Contains Link	Contains Money Words	Length	Class
1	1	1	1	1
2	0	0	0	0
3	1	0	1	1
4	0	1	0	1
5	1	1	0	1
6	0	0	1	0
7	1	0	0	0
8	0	1	1	1

9	1	1	1	1
10	0	0	0	0

Now using these values, we calculate the prior and conditional probabilities required for the prediction of class.

Prior probability for class spam **P(Spam)** = Number of Spams / Total = 6 / 10 = **0.6**

Prior probability for class ham **P(Ham)** = Number of Ham / Total = 4 / 10 = **0.4**

P(E1) = P (Contains Link = yes) = 5/10 = 0.5

P(E2) = P (Contains money words = no) = 5/10 = 0.5

P(E3) = P (Length = yes) = 5/10 = 0.5

Next, we need to calculate the Conditional probabilities, let us take points of data, which are (1,0,1) for this we predict using calculating for the class;

For Spam (Class = 1)

P (Contains Link = 1 | Spam): Spam with contains Link = 1: IDs 1,3,5,9 = 4 out of 6 are Spam which contains link =1

P (Contains Link = 1 | Spam): 4 / 6 **≈ 0.6667**

P (Contains Money words = 0 | Spam): Spam with contains Money words = 0: IDs 3 = 1 out of 6 are Spam which contains Money words = 0

P (Contains Money words = 0 | Spam): 1 / 6 **≈ 0.1667**

P (Length = 1 | Spam): Spam with length = 1: IDs 1,3,8,9 = 4 out of 6 are Spam with length =1

P (Length = 1 | Spam): 4 / 6 **≈ 0.6667**

Similarly for Ham (Class = 0)

P (Contains Link = 1 | Ham): Ham with contains Link = 1: IDs 7 = 1 out of 4 are Ham which contains link =1

P (Contains Link = 1 | Ham): 1 / 4 **≈ 0.25**

P (Contains Money words = 0 | Ham): Ham with contains Money words = 0: IDs 2,6,7,10 = 4 out of 4 are Ham which contains Money words = 0

P (Contains Money words = 0 | Ham): 4 / 4 **≈ 1**

P (Length = 1 | Ham): Ham with length = 1: IDs 6 = 1 out of 4 are Spam with length =1

P (Length = 1 | Ham): 1 / 4 **≈ 0.25**

Now for the posterior probability for the Spam class is as follows;

P(Features/Spam) = P (Contains Link = 1/Spam) X P (Contains Money words = 0/Spam) X P (Length = 1/Spam)

P(Features/Spam) = 0.6667 X 0.1667 X 0.6667

P(Features/Spam) ≈ 0.074

For posterior probability for the Ham class is as follows;

P(Features/Ham) = P (Contains Link = 1/Ham) X P (Contains Money words = 0/Ham) X P (Length = 1/Ham)

P(Features/Ham) = 0.25 X 1 X 0.25

P(Features/Ham) ≈ 0.0625

Now to calculate the P (Spam/Features) = P (Features/Spam) X P (Spam) / P (E1) X P (E2) X P (E3)

P(Spam/Features) = 0.074 X 0.6 / 0.5 X 0.5 X 0.5 = 0.044 / 0.125

P(Spam/Features) ≈ 0.352

Similarly, for P (Ham/Features) = P (Features/Ham) X P (Ham) / P (E1) X P (E2) X P (E3)

P(Ham/Features) = 0.0625 X 0.4 / 0.5 X 0.5 X 0.5 = 0.018 / 0.125

P(Ham/Features) ≈ 0.2

Hence, **P(Spam/Features) ≈ 0.352 is greater than P(Ham/Features) ≈ 0.2** for the data point (1,0,1). So, it is classified as **Spam** (Contains link = Yes, Contains Money words = 0, Length = Long).

Similarly, we can calculate for data points (1,0,0) as follows;

Prior probability for class spam **P(Spam)** = Number of Spams / Total = 6 / 10 = **0.6**

Prior probability for class ham **P(Ham)** = Number of Ham / Total = 4 / 10 = **0.4**

P(E4) = P (Contains Link = yes) = 5/10 = 0.5

P(E5) = P (Contains money words = no) = 5/10 = 0.5

P(E6) = P (Length = Short) = 5/10 = 0.5

For Spam (Class = 1)

P (Contains Link = 1 | Spam): 4 / 6 ≈ **0.6667**

P (Contains Money words = 0 | Spam): 1 / 6 ≈ **0.1667**

P (Length = 0 | Spam): 2 / 6 ≈ **0.3333**

For Ham (Class = 0)

P (Contains Link = 1 | Ham): 1 / 4 ≈ **0.25**

P (Contains Money words = 0 | Ham): 4 / 4 ≈ **1**

P (Length = 0 | Ham): 3 / 4 ≈ **0.75**

P(Features/Spam) = P (Contains Link = 1/Spam) X P (Contains Money words = 0/Spam) X P (Length = 0/Spam)

P(Features/Spam) = 0.6667 X 0.1667 X 0.3333

P(Features/Spam) ≈ 0.0370

P(Features/Ham) = P (Contains Link = 1/Ham) X P (Contains Money words = 0/Ham) X P (Length = 0/Ham)

P(Features/Ham) = 0.25 X 1 X 0.75

P(Features/Ham) ≈ 0.1875

Now, for P(Spam/Features) = 0.0370 X 0.6 / 0.5 X 0.5 X 0.5 = 0.0222 / 0.125

P(Spam/Features) ≈ 0.1776

P(Ham/Features) = 0.1875 X 0.4 / 0.5 X 0.5 X 0.5 = 0.075 / 0.125

P(Ham/Features) ≈ 0.6

Clearly in this case, **P(Ham/Features) > P (Spam/Features)**. Hence, the data points **(1,1,0)** which are Contains link=yes, contains money words=0, and Length =short belongs to class **Ham**.

Now, we use KNN classifier method as well to predict the class labels of emails whether they are Spam or Ham, which is as follows;

We convert the categorial data into numerical data points, after which we need to calculate the Euclidean distance between the points to predict the class.

We predict the class for data points (1,0,1) as follows;

ID	Contains Link, Contains Money words, and Length	Class	Euclidean Distance
1	(1,1,1)	Spam	Sqrt [(1-1) ^2 + (1-0) ^2 +(1-1) ^2)] = 1

2	(0,0,0)	Ham	Sqrt [(0-1) ^2 + (0-0) ^2 +(0-1) ^2)] = 1.41
3	(1,0,1)	Spam	Sqrt [(1-1) ^2 + (0-0) ^2 +(1-1) ^2)] = 0
4	(0,1,0)	Spam	Sqrt [(0-1) ^2 + (1-0) ^2 +(0-1) ^2)] = 1.73
5	(1,1,0)	Spam	Sqrt [(1-1) ^2 + (1-0) ^2 +(0-1) ^2)] = 1.41
6	(0,0,1)	Ham	Sqrt [(0-1) ^2 + (0-0) ^2 +(1-1) ^2)] = 1
7	(1,0,0)	Ham	Sqrt [(1-1) ^2 + (0-0) ^2 +(0-1) ^2)] = 1
8	(0,1,1)	Spam	Sqrt [(0-1) ^2 + (1-0) ^2 +(1-1) ^2)] = 1.41
9	(1,1,1)	Spam	Sqrt [(1-1) ^2 + (1-0) ^2 +(1-1) ^2)] = 1
10	(0,0,0)	Ham	Sqrt [(0-1) ^2 + (0-0) ^2 +(0-1) ^2)] = 1.41

Sorting and ordering these values according to the value of (1,0,1);

ID	Distance to Data point (1,0,1)	Class
3	**0**	**Spam**
1	**1**	**Spam**
9	**1**	**Spam**
6	**1**	**Ham**
7	**1**	**Ham**
2	**1.41**	**Ham**
5	**1.41**	**Spam**
8	**1.41**	**Spam**
10	**1.41**	**Ham**
4	**1.73**	**Spam**

Now with the **K=2**, we predict the class label for Data point (1,0,1) i.e. ID–3;

The nearest Neighbours of (1,0,1) are ID – 1,9 with datapoints (1,1,1) and distances of 1 which is nearest to 0. They have the class label as Spam. Hence, the class label for the Data point (1,0,1) based on the KNN with value k=2 is **Spam**.

We can do this **K=3** as well, which gives us the nearest Neighbours of datapoint (1,0,1) as ID-1,9,6 with Datapoints (1,1,1) and (0,0,1) and distances of 1 respectively. For these we have class label as Spam for ID-1,9 and Ham for ID-6. We choose the majority class for the datapoints in case of different ones for nearest Neighbours. Hence, based on this the **majority is Spam** and the predicted class label for datapoint (1,0,1) would be **Spam** as well based on **KNN with k=3**.

1b. write code (with AI assistant) to build a naive Bayes and KNN classifier. You can use hamspam.csv to test it out.

1b Ans: - Please find the Python code to build the naive bayes and KNN classifier in the GitHub link attached below;

https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/03032025_CW/Classwork_03032025.ipynb

2a. Create a ROC.
Step1: Given the threshold (0.95,0.90,0.85,0.80,0.75,0.70), derive True Positive and False Positive
Step2: Calculate the True Positive Rate (TPR) and False Positive Rate (FPR).
Step3: plot the set points (FRP, TPR) on the ROC diagram

2a Ans: - The True positives and False positives for the dataset give (Roc.csv) is as follows;

True Positives = True Positive (TP): Prediction >= threshold and True_Label = 1

False Positive (FP): Prediction >= threshold and True_Label = 0

True Negative (TN): Prediction < threshold and True_Label = 0

False Negative (FN): Prediction < threshold and True_Label = 1

True Positive Rate (TPR) = TP / (TP + FN) (Sensitivity or Recall)

False Positive Rate (FPR) = FP / (FP + TN) (1 - Specificity)

Based on the formulas, the below table is computed, and values are calculated;

	Threshold	TP	FP	TN	FN	TPR	FPR
0	0.95	39	4	74	33	0.541667	0.051282
1	0.90	46	5	73	26	0.638889	0.064103
2	0.85	51	5	73	21	0.708333	0.064103
3	0.80	54	5	73	18	0.750000	0.064103
4	0.75	55	6	72	17	0.763889	0.076923
5	0.70	58	6	72	14	0.805556	0.076923

The code to calculate the TP, and FP, and ROC plot is also in the provided GitHub link below;

https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/03032025_CW/Classwork_03032025.ipynb

2b. Write code (with AI assistant) to fit the model using your favorite classifier (NB, KNN, or Decision tree); using the hamspam.csv**,** ask to output an ROC curve and AUC score.

2b Ans: - The code to fit the model into the classifier like NB and get the output of ROC curve and AUC score is in the below GitHub link;

https://github.com/UB01976/is7332025/blob/main/data-mining-project-repo/03032025_CW/Classwork_03032025.ipynb