# Homework 3

Your Name: \_\_\_\_\_\_\_\_TsuCheng\_Lu\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID:\_\_\_\_\_\_\_\_A20525040\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1.** [40] **Manually** solve the questions below by using Naïve Bayes Classifier

We conducted a survey to collect people’s daily diets and try to build a model to predict whether their diets result in healthy conditions or not. The final results could be Yes, No. Note: using green rows as training, orange rows as testing.

|  |  |  |  |
| --- | --- | --- | --- |
| **Breakfast** | **Lunch** | **Dinner** | **Healthy?** |
| Ham | Carnivorous | Beef | Y |
| Milk | Carnivorous | Beef | N |
| Bread | Veggie | Pork | N |
| Bread | Veggie | Veggie | Y |
| Ham | Veggie | Veggie | Y |
| Milk | Carnivorous | Pork | N |
| Bread | Carnivorous | Beef | N |
| Ham | Veggie | Pork | Y |
| Milk | Veggie | Pork | Y |
| Milk | Carnivorous | Veggie | N |
| Noddle | Carnivorous | Pork | ? |

1). [5 points] What is laplace smoothing? And why we need it in the Naïve Bayesian classifier?

Laplace is a tool that resolve the problem of zero probabilities from the data. And why we need it in the naïve Bayesian classifier because when the naïve bayes calculate the zero probability or overfitting problems, and it is not helpful. So, we need Laplace smoothing by adding a pseudo count to every feature count for each class, and ensuring that no probability is zero.

2). [15 points] Using the Categorical Naive Bayesian Classification to make predictions on the test sets, present confusion matrix, and calculate accuracy, precision, recall, F1 measure, specificity, by considering Y as positive label

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Breakfast** | **Lunch** | **Dinner** | **Healthy?** | **predicted** | **Matrix** |
| Bread | Carnivorous | Beef | N | N | TN |
| Ham | Veggie | Pork | Y | Y | TP |
| Milk | Veggie | Pork | Y | N | FN |
| Milk | Carnivorous | Veggie | N | N | TN |

For the training set, we calculate all of the possibilities of feature, then we can get the prediction for the testing set.

|  |  |  |  |
| --- | --- | --- | --- |
| **欄1** | **欄2** | **欄3** | **欄4** |
|  |  | Predicted Labels | |
| Actual Labels | | Y | N |
| Y |  | 1 | 1 |
| N |  | 0 | 2 |

True Positive (TP) = 1

False Positive (FP) = 0

True Negative (TN) = 2

False Negative (FN) = 1

Accuracy = (1+2) / 4 = 3/4 = 0.75 = 75%

Precision = 1 / (1+0) = 1 =1 = 100%

Recall = 1 / (1+1) = 0.5 = 50%

F1 measure = 2 \* (1 \* 0.5) / (1 + 0.5) = 1/1.5 = 0.666 = 66%

Specificity = 2 / (1+2) = 2/3 = 0.666 = 66%

2). [20 points] Using the Categorical Naive Bayesian Classification to make prediction on the unseen data (note: building the model by using both the green and orange rows, and predicting the label for unseen data/last row)

|  |  |  |  |
| --- | --- | --- | --- |
| **Breakfast** | **Lunch** | **Dinner** | **Healthy?** |
| Ham | Carnivorous | Beef | Y |
| Milk | Carnivorous | Beef | N |
| Bread | Veggie | Pork | N |
| Bread | Veggie | Veggie | Y |
| Ham | Veggie | Veggie | Y |
| Milk | Carnivorous | Pork | N |
| Bread | Carnivorous | Beef | N |
| Ham | Veggie | Pork | Y |
| Milk | Veggie | Pork | Y |
| Milk | Carnivorous | Veggie | N |
| Noddle | Carnivorous | Pork | ? |

**First of all, we have to measure that taking breakfast for the breakfast is healthy or not.**

**P(N) = 5/10 = 0.5**

**P(Y) = 5/10 = 0.5**

**P (Y | C1) = 1/5 = 0.2**

**P (N | C1) = 4/5 = 0.8**

**P (Y | D2) = 2/5 = 0.4**

**P (N | D2) = 2/5 = 0.4**

We set the lunch Carnivorous as c1, and Veggie as c2. Also, setting the dinner Beef as d1, Pork as d2, and Veggie as d3. Then we can calculate the probability of the unseen value.

**P (Y | C1, D2) = 0.5(0.2\*0.4) = 0.04**

**P (N | C1, D2) = 0.5(0.8\*0.4) = 0.16**

As a result, the probability of unseen label Y is 0.04, and N is 0.16. So, the answer is N.

**2. (60 points) Python practice for Naïve Bayes classification**

**Use the Malware\_MultiClass.csv data (predicting the column “classification”), and run 5 Naïve Bayes techniques by using hold-out evaluations (85% as training)**

Note:

* You need to change different/multiple parameters to find the best NB model.
* You should evaluate the models by using accuracy, micro-precision and micro-recall, micro-F1 and micro-AUC
* Give conclusions about the best model by comparing the evaluation metrics above

Submission

* The ipynb and saved html files
* The comparison and conclusions of different models

The GaussianNB has the highest accuracy (0.7021) among the models. And BernoulliNB and MultinomialNB have similar accuracy, precision, F1 score, and recall values. Their accuracy is around 0.67 and micro-F1 around 0.62. In more detail, ComplementNB and MultinomialNB have identical performance metrics (accuracy, precision, F1, recall, and AUC). CategoricalNB stands out as having lower accuracy and precision compared to the other models, and it’s micro-F1 and recall are with nan values. Based on the metrics, GaussianNB would be the best model in terms of accuracy.