Pascua, Primrose F.
Saliba, Martin Thomas D.
CS 180 - MP 2

INTRODUCTION

We chose a dataset containing prices of laptops and their respective specifications. This dataset really piqued our interests as we both are BS Computer Science students but we don't know any technical stuff when it comes to laptop specifications and prizes. A regular person buying a laptop without any background or knowledge on tech stuff will find it hard to know if he/she is spending right on the laptop he/she chose. Our objective on this MP is to classify the price range of a laptop when given a specification. In this MP, we will use Naive Bayes Classifier to classify the given specifications. We will also have a brief comparison on Multinomial, Gaussian, and Bernoulli which are different Naive Bayes method present in scikit learn.

Dataset:
https://www.kaggle.com/ionaskel/laptop-prices?fbclid=IwAR0FDkakFjxcbfyuCF4QXClCH6_G9blRpqpEAVxGLRxXVbSZ4Vm90QySzWo

METHODOLOGY

The dataset contains 12 columns namely: Unnamed(*index*), Company, Product, TypeName, Inches, ScreenResolution, Cpu, Ram, Memory, Gpu, OpSys, Weight, and Price_euros.

As stated previously, we will use a Naive Bayes Classifier in this MP. Here is a quick run-through of the methodology (You can refer to the jupyter notebook file for a more detailed methodology):

- Pre-processing
    - Dropping of unnecessary columns
    - Converting entries into numerical data (TypeName, OpSys, Ram, Cpu, Gpu, ScreenResolution, and Memory)
    - Changing the Price entry into a classification (Low, Average, High)
    - Save into a new .csv file.

- Classification
    - Use Naive Bayes (Multinomial, Gaussian, Bernoulli)
    - Create X and Y vectors from the cleaned dataset.
    - Split into testing and training sets.
    - Fit and predict using the 3 methods.
- Input
    - Sample Input: [1,1,1,4,3,1,1]
    - Sample output:

```
Low -> 0-799 euros
Average -> 800-1499 euros
High 1500 euros and above

('MonomialNB Price Range Prediction: ', 'Low')
('GaussianNB Price Range Prediction: ', 'Low')
('BernoulliNB Price Range Prediction: ', 'Low')
```

DATA & ANALYSIS

- Gaussian

```
             precision    recall  f1-score   support

    Average       0.65      0.65      0.65       128
       High       0.59      0.69      0.64        71
        Low       0.85      0.78      0.81       127

avg / total       0.72      0.71      0.71       326

[[83 29 16]
 [21 49  1]
 [23  5 99]]
```

Accuracy = 0.7085889570552147

- Multinomial

```
              precision    recall  f1-score   support

     Average       0.54      0.70      0.61       128
        High       0.58      0.46      0.52        71
         Low       0.83      0.66      0.74       127

 avg / total       0.66      0.63      0.64       326

[[90 23 15]
 [36 33  2]
 [42  1 84]]
```

Accuracy = 0.6349693251533742

- Bernoulli

```
              precision    recall  f1-score   support

     Average       0.00      0.00      0.00       128
        High       0.00      0.00      0.00        71
         Low       0.39      1.00      0.56       127

 avg / total       0.15      0.39      0.22       326

[[  0   0 128]
 [  0   0  71]
 [  0   0 127]]
```

Accuracy = 0.3895705521472393

Looking at the following tables, Gaussian is the most effective of the three methods in this dataset with an accuracy of at least 80%. Multinomial has a somehow acceptable accuracy with at least 60% while Bernoulli performed really bad with its accuracy not even reaching 50%.

CONCLUSION

In this dataset and with the pre-processing done by the students, Gaussian performed best, next is Multinomial, and Bernoulli performed worst. However, it is

important to note that this may not be the case for other datasets. Thus, choosing a correct method is vital when doing these kinds of projects.

INDIVIDUAL CONTRIBUTION

- Pascua, Primrose
    - MP Idea and dataset
    - Cleaning of code
    - User input classification
- Saliba, Martin Thomas
    - Pre-processing of data
    - Classification
    - Documentation