Analysis and Systems of Big Data

Pranav Parameshwaran

COE17B036

Part A & B

The python file for executing Part A & B is Part-A_B.py.

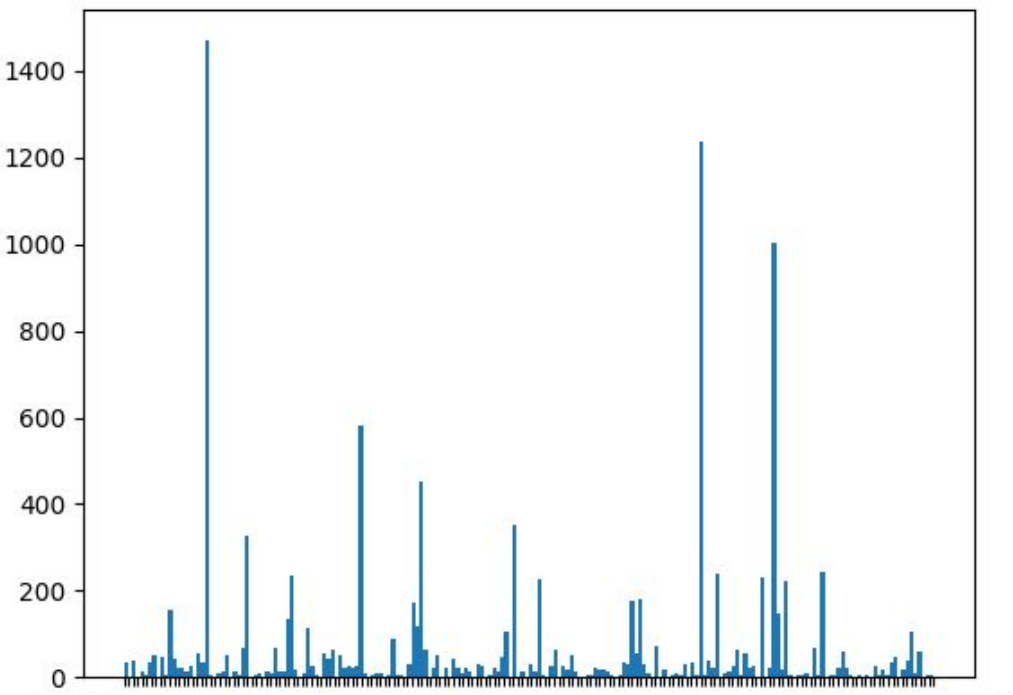
In this we will do preprocessing of the given dataset. We merge two columns into a single attribute. Hence our dataset has only 4 columns.

```
       Symbol_code   Family_code
count  11378.000000  11378.000000
mean    2054.935841     84.867639
std     1180.840607     50.499482
min        0.000000      0.000000
25%     1060.000000     37.000000
50%     2056.000000     85.000000
75%     3112.750000    130.000000
max     4096.000000    177.000000

count     11378
unique    10542
top       aster
freq          6
Name: Name or Symbol, dtype: object
Mode of Family:  Asteraceae
```

We have Label Encoded the Family and the Symbol attributes for visualization purpose.

PLOT :



Plot of the Histogram of the Distribution of the Family.

Data Cleaning

```
Symbol                          11378
Synonym Symbol                   7281
Scientific Name with Author     11378
National Common Name             4092
Family                          11378
dtype: int64
```

The count of number of values present in the dataset.

Since Synonym Symbol and National Common Name are the only columns that contain NULL Values. From observation we find that exactly one column is NULL per tuple. Since Synonym Symbol values are all distinct, we do not have the mode to replace the NULL Values.

Therefore, Combined these two columns into one Column (Name or Symbol). Cleaning is applied to remove excess spaces and Null values. Thus Shape is 11378 x 4.

I performed Transactional Encoding on the given dataset for converting the categorical data into a one-hot encoded NumPy array of ones and zeros.

These are the following algorithms that I have used for generating patterns:

- Frequent Itemset Mining
    - Apriori Algorithm
    - FP Growth Algorithm

- Closed Frequent Itemset Mining
    - A-Close
    - CHARM

- Maximal Frequent Itemset Mining
    - MAFIA
    - Pincer Search

- Longest Frequent Itemset Mining
    - LF_Apriori
    - LF_FPGrowth

```
RULES: ----------------------



Minimum Support = 25%
     support                                               itemsets
0        0.5  (Viscum flavescens sensu Pursh p.p., non Viscu...
1        0.5  (Phoradendron serotinum (Raf.) M.C. Johnst. va...
2        0.5       (Phoradendron serotinum (Raf.) M.C. Johnst.)
3        0.5                    (Phoradendron macrotomum Trel.)
4        0.5  (Phoradendron leucarpum (Raf.) Reveal & M.C. J...
..       ...                                                ...
251      0.5  (Phoradendron serotinum (Raf.) M.C. Johnst., P...
252      0.5  (Phoradendron serotinum (Raf.) M.C. Johnst., P...
253      0.5  (Phoradendron serotinum (Raf.) M.C. Johnst., P...
254      0.5  (Phoradendron serotinum (Raf.) M.C. Johnst., P...
255      0.5  (Phoradendron serotinum (Raf.) M.C. Johnst., P...

[256 rows x 2 columns]
```

```
--------------------------------------------------------------------------------------------------------
FOR FAMILY:  Xyridaceae
Minimum Support =  1.75
[['XYTO', 'Xyridaceae']]
Minimum Support =  1.75
[[{'XYTO'}, {1, 2, 3, 4, 5, 6}], [{'XYTO', 'Xyridaceae'}, {1, 2, 3, 4, 5, 6}]]
Minimum Support =  1.75
[{'XYTO', 'Xyridaceae'}]
Minimum Support =  1.75
['XYTO', 'Xyridaceae']
Minimum Support = 25%
[['XYTO', 'Xyridaceae']]
Minimum Support = 25%
[['XYTO', 'Xyridaceae']]
```

Rule Generation

Part C

The python file for executing Part C is PartC.py.

Decision Tree

Using Gini Index and Entropy Method for training on the Iris dataset :

```
Results Using Gini Index:
Predicted values:
['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-virginica'
s-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor'
is-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor', 'I
s-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', '
ca', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-ve
Confusion Matrix:
 [[19  0  0]
 [ 0 20  1]
 [ 0  1 19]]
Accuracy :  96.66666666666667
Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       0.95      0.95      0.95        21
           2       0.95      0.95      0.95        20

    accuracy                           0.97        60
   macro avg       0.97      0.97      0.97        60
weighted avg       0.97      0.97      0.97        60
```

```
Results Using Entropy:
Predicted values:
['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-virgin
s-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor', 'Iris-versico
is-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor'
s-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica
ca', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iri
Confusion Matrix:
 [[19  0  0]
 [ 0 20  1]
 [ 0  1 19]]
Accuracy :  96.66666666666667
Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       0.95      0.95      0.95        21
           2       0.95      0.95      0.95        20

    accuracy                           0.97        60
   macro avg       0.97      0.97      0.97        60
weighted avg       0.97      0.97      0.97        60
```

Bayes Classifier

Used Gaussian Bayes Classifier for training on the Iris Dataset.

```
Results BayesClassifier:
Predicted values:
['Iris-setosa', 'Iris-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-virginica
-versicolor', 'Iris-versicolor', 'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor'
s-setosa', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', 'Iris-versicolor', 'I
-virginica', 'Iris-virginica', 'Iris-setosa', 'Iris-versicolor', 'Iris-virginica', '
a', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-virginica', 'Iris-ve
Confusion Matrix:
 [[19  0  0]
 [ 0 19  2]
 [ 0  1 19]]
Accuracy :  95.0
Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        19
           1       0.95      0.90      0.93        21
           2       0.90      0.95      0.93        20

    accuracy                           0.95        60
   macro avg       0.95      0.95      0.95        60
weighted avg       0.95      0.95      0.95        60
```