# Big Data Assignment Report

Kausik N

COE17B010

## Steps to RUN:

Pip Install Modules –

pickle, numpy, pandas, matplotlib seaborn, plotly, tqdm, functools, mlxtend, sklearn

Run - ***main.py***

# Pre-processing

## File: Preprocessing.py

## Methods:

1. Reading Dataset – ReadCSVFile(path)
2. Frequency Distribution – FreqDist(Data)
3. Missing Count – MissingCount(Data)
4. Cleaning – MissingClean(Data)

   ### Remove Technique

   If missing data in field (Symbol or Scientific Name) – Remove Row as this is unique data

   ### Ignore Technique

   If missing data in Synonym field – no need to do anything as it is a optional field for data

   ### Subset Technique

   If missing data in Common Name or Family – replace smartly

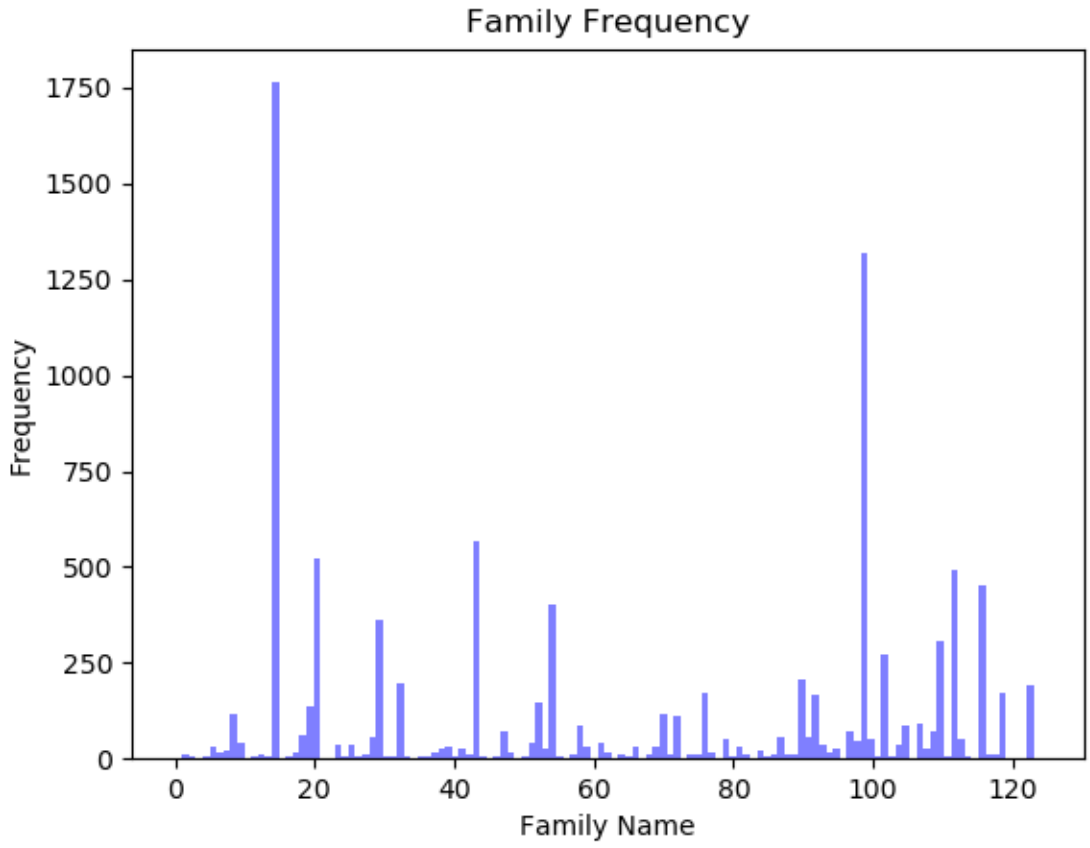
   Family of subspecies will be same as subspecies

So, if data missing, using Scientific Name, identify its species and use its Family to fill in missing family name (Subset Technique)

Redundant Technique – RedundantClean(Data)

If data rows are repeated, remove those redundant rows and keep only unique rows

5. Visualising – Histogram(Data) for Family Frequency



# Part A

File: Algorithms.py

Algorithms:

Encoding – One Hot Encoding – OneHotEncoding(Data)

FIM

1. Apriori
2. FPGrowth

CFI

1. Charm
2. Apriori-Close

MFI

1. Pincer Search
2. Mafia

LFI

1. Apriori Based LFI
2. FPGrowth Based LFI

# Part B

## File: Algorithms.py

Function: RuleMining(FrequentItemsets)

Done for FPGrowth generated frequent itemsets

Minimum Confidence = 1.0

```
- Part B - Rules Mining ------------------------------------------------------------------------------------------
RuleSet:
 <bound method NDFrame.head of       antecedents     consequents  antecedent support  consequent support  support  confidence
ction
2          (ALGR)  (Alismataceae)                0.14                0.58     0.14    1.000000  1.724138    0.0588         inf
4         (ALTR7)  (Alismataceae)                0.10                0.58     0.10    1.000000  1.724138    0.0420         inf
6         (SALA2)  (Alismataceae)                0.26                0.58     0.26    1.000000  1.724138    0.1092         inf
1     (Aceraceae)     (NULLVALUE)                0.20                0.38     0.10    0.500000  1.315789    0.0240    1.240000
7  (Alismataceae)         (SALA2)                0.58                0.26     0.26    0.448276  1.724138    0.1092    1.341250
0     (NULLVALUE)     (Aceraceae)                0.38                0.20     0.10    0.263158  1.315789    0.0240    1.085714
3  (Alismataceae)          (ALGR)                0.58                0.14     0.14    0.241379  1.724138    0.0588    1.133636
5  (Alismataceae)         (ALTR7)                0.58                0.10     0.10    0.172414  1.724138    0.0420    1.087500>
```

# Part C

## File: DecisionTree.py, BayesClassifier.py

Decision Tree done for balance-scale.csv dataset

```
Decision Tree on balance-scale dataset:



Dataset Length:  625
Dataset Shape:  (625, 5)
Dataset:    class name  left-weight  left-distance  right-weight  right-distance
0           B           1            1              1             1
1           R           1            1              1             2
2           R           1            1              1             3
3           R           1            1              1             4
4           R           1            1              1             5
```

```
Results Using Gini Index:
Predicted values:
['R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'L'
 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'L' 'R'
 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R'
 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:
 [[ 0  6  7]
 [ 0 67 18]
 [ 0 19 71]]
Accuracy:
 73.40425531914893
C:\Users\Kausik N\AppData\Local\Programs\Python\Python38-32\lib\site-pack

Precision and F-score are ill-defined and being set to 0.0 in labels with

Report:
              precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.73      0.79      0.76        85
           R       0.74      0.79      0.76        90

    accuracy                           0.73       188
   macro avg       0.49      0.53      0.51       188
weighted avg       0.68      0.73      0.71       188
```

```
Results Using Entropy:
Predicted values:
['R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L'
 'L' 'R' 'L' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L'
 'L' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L'
 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'L' 'R' 'L' 'L' 'L' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'R' 'R' 'L' 'R' 'L'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'L' 'L' 'L' 'R' 'L' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L'
 'L' 'L' 'L' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'L' 'R'
 'L' 'R' 'R' 'L' 'L' 'R' 'L' 'R' 'R' 'R' 'R' 'R' 'L' 'R' 'R' 'R' 'R' 'R'
 'R' 'L' 'R' 'L' 'R' 'R' 'L' 'R' 'L' 'R' 'L' 'R' 'L' 'L' 'L' 'L' 'L' 'R'
 'R' 'R' 'L' 'L' 'L' 'R' 'R' 'R']
Confusion Matrix:
 [[ 0  6  7]
 [ 0 63 22]
 [ 0 20 70]]
Accuracy:
 70.74468085106383
C:\Users\Kausik N\AppData\Local\Programs\Python\Python38-32\lib\site-pack

Precision and F-score are ill-defined and being set to 0.0 in labels with

Report:
              precision    recall  f1-score   support

           B       0.00      0.00      0.00        13
           L       0.71      0.74      0.72        85
           R       0.71      0.78      0.74        90

    accuracy                           0.71       188
   macro avg       0.47      0.51      0.49       188
weighted avg       0.66      0.71      0.68       188
```

Bayes Classifier done for Iris Dataset

```
Bayes Classifier on Iris dataset:



Results Using Bayes Classifier:
Predicted values:
[0 1 1 0 2 2 2 0 0 2 1 0 2 1 1 0 1 1 0 0 1 1 2 0 2 1 0 0 1 2 1 2 1 2 2 0 1
 0 1 2 2 0 1 2 1 2 0 0 0 1 0 0 2 2 2 2 2 1 2 1]
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
```