Analytics and Systems of Big Data
Assignment-1

CED16I010
Kruttika Bhat

Folder Navigation:

Files:

1. A.py : Code for data preprocessing, descriptive and predictive analytics part (as part of part A) described in the report.
2. aclose.py: A-Close algorithm for CFI (as part of part A).
3. apriori.py : Apriori program for FIM (as part of part A).
4. B.py : Code for part B.
5. C.py : Code for part C.
6. charm.py : CHARM program for CFI (as part of part A).
7. data_1.csv : Dataset for Asteraceae.
8. data_2.csv : Dataset for Poaceae.
9. diabetes.csv : Diabetes dataset used in part C.
10. fpgrowth.py : FP Growth program for FIM (as part of part A).
11. get_classes.py : Generates the datasets (data_1.csv, data_2.csv) for the 2 classes required for ARM.
12. mafia.py : MAFIA program for MFI (as part of part A).
13. pincer.py : Pincer search program for MFI (as part of part A).
14. stateDownload : Dataset used for part A and part B.

Folders:

1. ARM : Outputs for association rule mining.
    a. AClose
        i. Asteraceae
        ii. Poaceae
    b. Apriori
    c. CHARM
    d. FPGrowth
    e. MAFIA
    f. Pincer
2. B : Outputs for part B.
3. Figures: All figures for part A.
    a. Descriptive: Plots shown in descriptive analysis section.
        i. Family: Pie plots
    b. Predictive: Plots shown in the predictive analysis section.

Prerequisites:

- Pandas
- Seaborn
- Scikit-learn
- Apyori
- Pyfpgrowth

Table of Contents:

Part A

Develop a complete statistical package supporting features for

1. **Descriptive Analytics**
2. **Predictive Analytics**

As a part of your project, you are expected to test drive all preprocessing concepts and descriptive analytics mechanisms with respect to your assigned dataset. Further to completion of 1 & 2, you may test drive any 2 algorithms for FIM, CFI, MFI, LFI. Also use the patterns generated from any one of the itemsets to mine interesting rules.

Data Preprocessing:

The dataset is a plain text file which is saved as stateDownload. We use the pandas library to read it as a dataframe. First let us look at the data and see what attributes it has.

Dimensions of the dataset:
[8418 rows x 5 columns]

Column headings:
```
Index(['Symbol',
       'Synonym Symbol',
       'Scientific Name with Author',
       'National Common Name',
       'Family'],
  dtype='object')
```

Datatype of each column:

Symbol	object
Synonym Symbol	object
Scientific Name with Author	object
National Common Name	object
Family	object

All columns are string values. There are no numerical values.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family
0	JUAM	NaN	Justicia americana (L.) Vahl	American water-willow	Acanthaceae
1	JUAM	DIAM2	Dianthera americana L.	NaN	Acanthaceae
2	JUAM	DIAMS	Dianthera americana L. var. subcoriacea (Ferna...	NaN	Acanthaceae
3	JUAM	JUAMS	Justicia americana (L.) Vahl var. subcoriacea ...	NaN	Acanthaceae
4	JUAM	JUMO2	Justicia mortuifluminis Fernald	NaN	Acanthaceae
...	...	...	...	...	...
8413	ZAPA	ZAPAM	Zannichellia palustris L. var. major (Hartm.) ...	NaN	Zannichelliaceae
8414	ZAPA	ZAPAS	Zannichellia palustris L. var. stenophylla Asc...	NaN	Zannichelliaceae
8415	KALLS	NaN	Kallstroemia Scop.	caltrop	Zygophyllaceae
8416	KAPA	NaN	Kallstroemia parviflora J.B.S. Norton	warty caltrop	Zygophyllaceae
8417	KAPA	KAIN3	Kallstroemia intermedia Rydb.	NaN	Zygophyllaceae

8418 rows × 5 columns

There is a column titled Scientific Name with Author. On further observation of the raw data:

1. There are values which can easily be split.
    a. Justicia americana (L.) Vahl
    b. Ruellia hybrida Pursh
    c. Acer palmatum Thunb.
2. There are values which are very long and more difficult to split. This would require a knowledge on the rules to distinguish between the author and the scientific name.
    a. Dianthera americana L. var. subcoriacea (Fernald) Shinners
    b. Ruellia caroliniensis (J.F. Gmel.) Steud. var. cheloniformis Fernald
    c. Ruellia caroliniensis (J.F. Gmel.) Steud. ssp. caroliniensis
3. There are values which don't have a species name but instead have just the genus name (and so it's taxonomic rank would be genus). In the given examples L. seems to refer to **Carl Linnaeus** who would be the author.
    a. Ruellia L.
    b. Justicia L.
    c. Acer L.

As such, making a guess on how to split the column into 2 columns of 'Scientific Name' and 'Author' could result in contamination of the data due to poor knowledge of scientific naming. If we have an expert who could tell us the rules to split it then it would be possible. Another column could also be added called taxonomic ranking. This would require going through the dataset row by row and it would be very tedious.
As the author name is not easy to identify for some cases, we will not have a column for the author name.

But we can generate a column titled Genus, since each row has at least this, if not the species name as well. Genus is assigned as the first word in the string. We shall work with these 6 columns.

describe() output:

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	8418	5299	8418	3115	8418	8418
unique	3119	5299	8417	2553	170	1120
top	CRCR2	ELHI5	Sarracenia purpurea L. ssp. purpurea var. purp...	hybrid violet	Asteraceae	Carex
freq	40	1	2	7	1119	278

This gives count, unique, top (most frequently occuring value) and freq (frequency of top). Since there are no numerical values, measures such as mean, median and deviation are not listed.

info() output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8418 entries, 0 to 8417
Data columns (total 6 columns):
 #   Column                      Non-Null Count  Dtype 
---  ------                      --------------  ----- 
 0   Symbol                      8418 non-null   object
 1   Synonym Symbol              5299 non-null   object
 2   Scientific Name with Author 8418 non-null   object
 3   National Common Name        3115 non-null   object
 4   Family                      8418 non-null   object
 5   Genus                       8418 non-null   object
dtypes: object(6)
memory usage: 394.7+ KB
```

This gives the columns, number of non-null values and data type. Nothing new to what we have already seen.

Data Quality Assessment:

1. Missing values
Number and percentage of null values:

Column title	Count	Percentage (Count/Total Number of Rows)
Symbol	0	0.0
Synonym Symbol	3119	0.370516

Scientific Name with Author	0	0.0
National Common Name	5303	0.629960
Family	0	0.0
Genus	0	0.0

The number of null values per column were found. There were 2 columns which had a very large number of null values. These 2 columns had null values which were around half of the total number of records.

I. Synonym Symbol
II. National Common Name

To deal with these missing values, the methods are:

1. Estimate the values: Since the columns store string values, there is no way of estimating it as we cannot use mean, median or mode. As a large percentage of values are missing, we cannot use interpolation technique either.
2. Eliminate rows with missing data:
    a. If there's at least 1 null value in row: Dataset becomes empty. Implying that each record has a null value in at least one of the 2 columns. Removing these records is not an option.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	0	0	0	0	0	0
unique	0	0	0	0	0	0
top	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN	NaN

    b. Eliminate rows where Synonym Symbol is missing: Number of families and genus reduces. All National Common Names would be null. Not a great idea to perform this either as it would change the dataset greatly.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	5299	5299	5299	0	5299	5299
unique	1505	5299	5299	0	152	910
top	CRCR2	ELHI5	Antennaria ampla Bush	NaN	Asteraceae	Panicum
freq	39	1	1	NaN	753	209

    c. Eliminate rows where National Common Name is missing: Number of families remains the same. Number of genus reduces. Dataset size reduces by more than

half. All Synonym Symbols become null. Not a good choice to remove these rows either.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	3115	0	3115	3115	3115	3115
unique	3115	0	3115	2553	170	746
top	EUALA2	NaN	Heteranthera Ruiz & Pav.	hybrid violet	Asteraceae	Carex
freq	1	NaN	1	7	366	139

d. Eliminate rows where there are 2 null values in the row (any row can have at max 2 null values): Number of families remains the same and the dataset doesn't change significantly.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	8414	5299	8414	3115	8414	8414
unique	3117	5299	8413	2553	170	1120
top	CRCR2	ELHI5	Sarracenia purpurea L. ssp. purpurea var. purp...	hybrid violet	Asteraceae	Carex
freq	40	1	2	7	1119	278

The dataset size doesn't change much in 4th case, let's see what these rows are where both columns are null.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
313	APFL	NaN	Apocynum ×floribundum Greene (pro sp.) [andros...	NaN	Apocynaceae	Apocynum
4268	MERO2	NaN	Mentha ×rotundifolia (L.) Huds. (pro sp.) [lon...	NaN	Lamiaceae	Mentha
7050	ARONI2	NaN	Aronia Medik.	NaN	Rosaceae	Aronia
8304	VIPR4	NaN	Viola ×primulifolia L. (pro sp.) [lanceolata ×...	NaN	Violaceae	Viola

These are not duplicates. If we delete these rows, the dataset wouldn't change significantly but it wouldn't affect the number of null values significantly either. So, we will keep these rows.

As we cannot eliminate any values and rather than filling a wrong string, we shall fill all the empty values with a string 'NA'. Later when encoding this string will be given a unique value.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
0	JUAM	NA	Justicia americana (L.) Vahl	American water-willow	Acanthaceae	Justicia
1	JUAM	DIAM2	Dianthera americana L.	NA	Acanthaceae	Dianthera
2	JUAM	DIAMS	Dianthera americana L. var. subcoriacea (Ferna...	NA	Acanthaceae	Dianthera
3	JUAM	JUAMS	Justicia americana (L.) Vahl var. subcoriacea ...	NA	Acanthaceae	Justicia
4	JUAM	JUMO2	Justicia mortuifluminis Fernald	NA	Acanthaceae	Justicia
...	...	...	...	...	...	...
8413	ZAPA	ZAPAM	Zannichellia palustris L. var. major (Hartm.) ...	NA	Zannichelliaceae	Zannichellia
8414	ZAPA	ZAPAS	Zannichellia palustris L. var. stenophylla Asc...	NA	Zannichelliaceae	Zannichellia
8415	KALLS	NA	Kallstroemia Scop.	caltrop	Zygophyllaceae	Kallstroemia
8416	KAPA	NA	Kallstroemia parviflora J.B.S. Norton	warty caltrop	Zygophyllaceae	Kallstroemia
8417	KAPA	KAIN3	Kallstroemia intermedia Rydb.	NA	Zygophyllaceae	Kallstroemia

8418 rows × 6 columns

2. Inconsistent Values

As seen earlier, the data was found for each column and all are in string format. There is no attribute which could lead to inconsistent forms such as address, phone number, date, etc. Here, if we consider ‘Scientific Name with Author’, there is a possibility of inconsistent forms. As we are unaware of how to break it down further properly, we will leave it as it is. All other columns are consistent.

3. Duplicate Values

Check the values of each column. In Scientific Name with Author, where you would expect there to be no repeated values, there is a repeated value. There is one value which occurs twice. On checking, it appears that this is one duplicate.

```
Sarracenia purpurea L. ssp. purpurea var. purpurea    2
Fimbristylis capillaris (L.) A. Gray                  1
Scutellaria nervosa Pursh var. calvifolia Fernald     1
Panicum walteri Pursh                                 1
Rubus pratensis L.H. Bailey                           1
                                                     ..
Bromus anatolicus Boiss. & Heldr.                     1
Ramischia secunda (L.) Garcke                         1
Vitis L.                                              1
Geum L.                                               1
Apocynum pumilum (A. Gray) Greene                     1
Name: Scientific Name with Author, Length: 8417, dtype: int64
```

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
7760	SAPUP6	NA	Sarracenia purpurea L. ssp. purpurea var. purp...	northern purple pitcherplant	Sarraceniaceae	Sarracenia
7765	SAPUP6	SAPUP7	Sarracenia purpurea L. ssp. purpurea var. purp...	NA	Sarraceniaceae	Sarracenia

So, these 2 rows can be combined to remove the duplicate row.

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	8417	8417	8417	8417	8417	8417
unique	3119	5300	8417	2554	170	1120
top	CRCR2	NA	Fimbristylis capillaris (L.) A. Gray	NA	Asteraceae	Carex
freq	40	3118	1	5302	1119	278

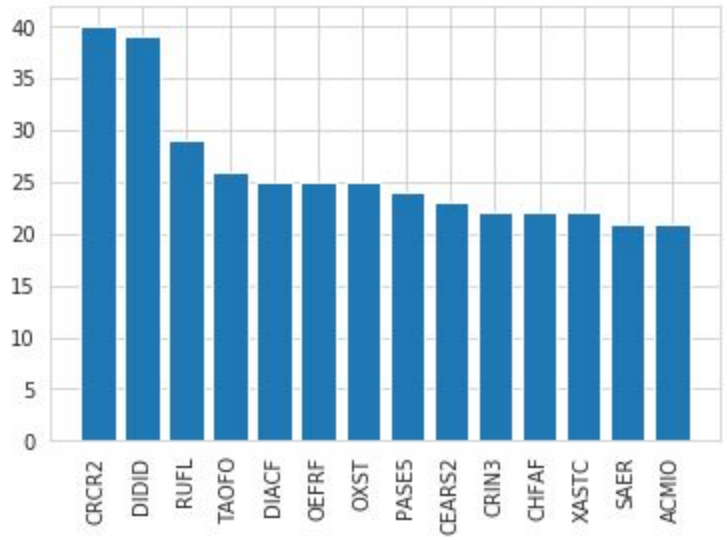
	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
7760	SAPUP6	SAPUP7	Sarracenia purpurea L. ssp. purpurea var. purp...	northern purple pitcherplant	Sarraceniaceae	Sarracenia

Now, all the values in ‘Scientific Name with Author’ are unique.

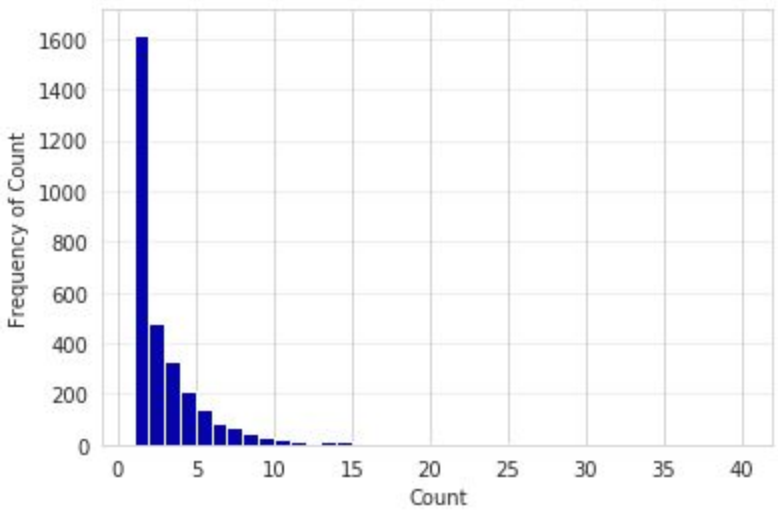
Descriptive Analytics:

1. Symbol:

This column contains 3119 unique values. It is not possible to construct a plot for all these values. We'll just look at the values whose count is greater than 20. There are 14 such values as seen in the bar graph below.

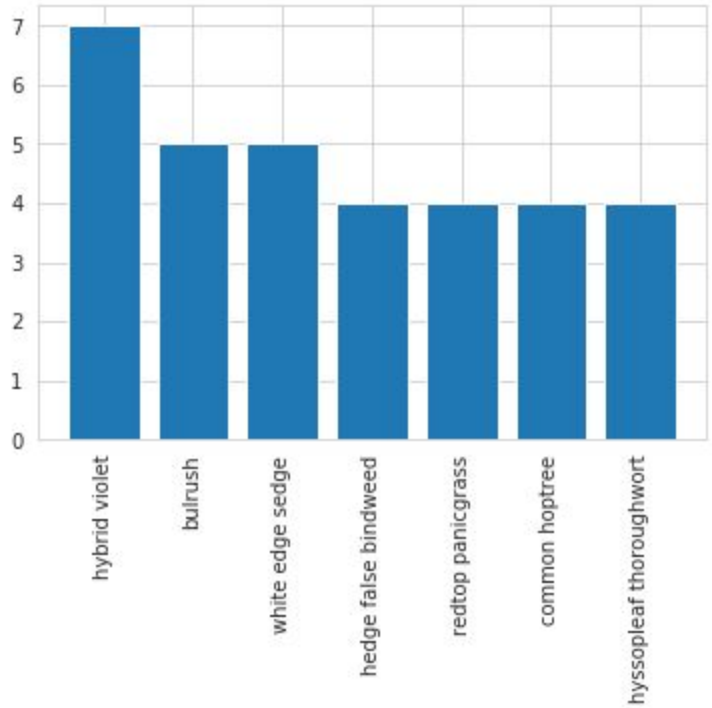


To see how many values were there with count of 1, a histogram with bin size 1, was plotted which took the count of each unique value. Out of 3119, 1614 values have a count of 1. This is about half the total number of unique values.

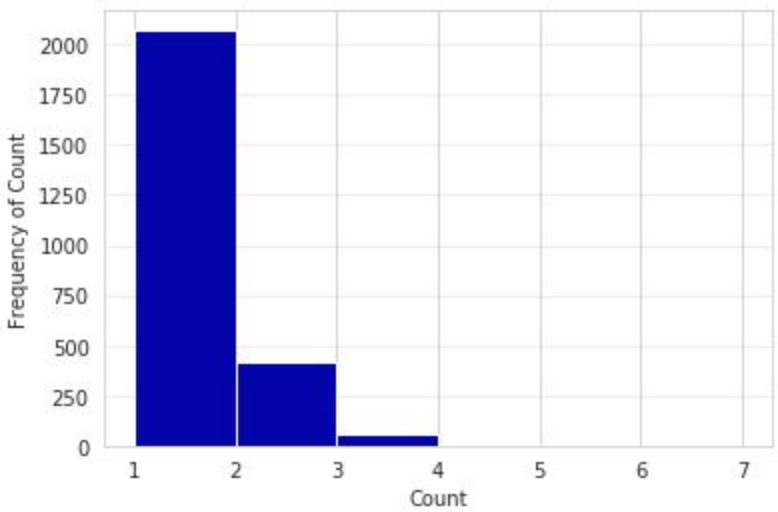


2. National Common Name:

This column contains 2553 unique values (not including ‘NA’). We’ll look at the values whose count is greater than 3. There are 7 such values.



Out of 2553, 2069 values have a count of 1. A histogram is plotted with bin size 1, to visualise this count.



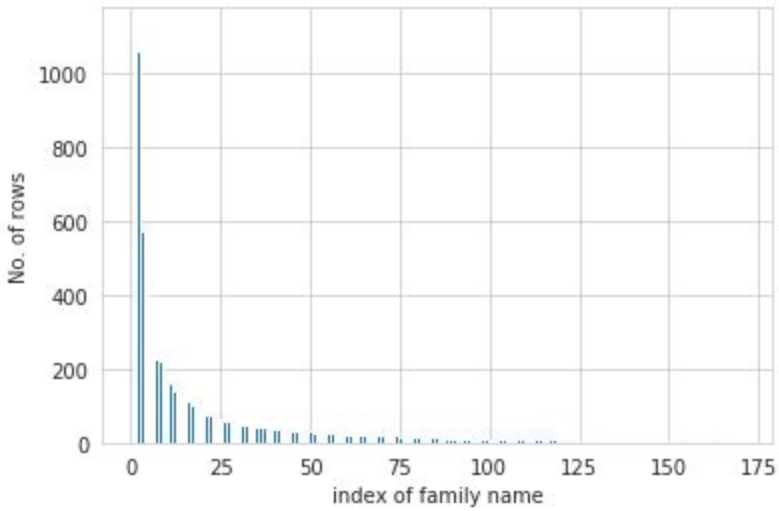
3. Family

This column contains 170 unique values. The number of rows belonging to each family was displayed in a bar chart in descending order. To see all the names of the families, please run the code.

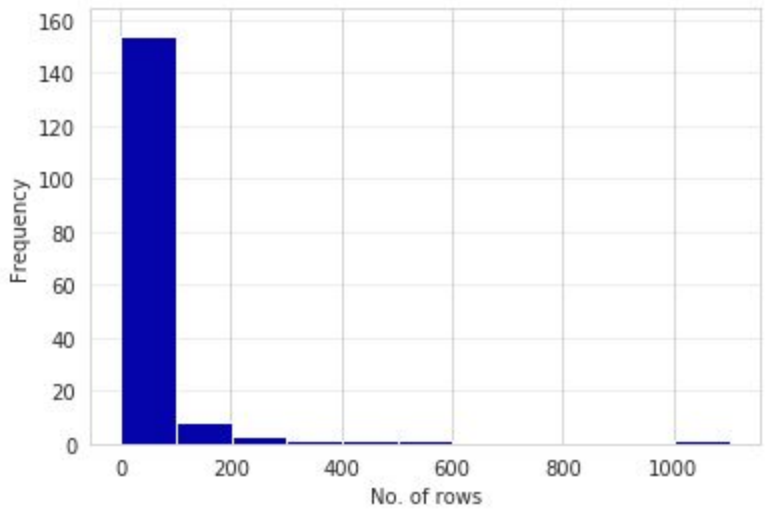
0	Asteraceae	1119
1	Poaceae	1058
2	Cyperaceae	569
3	Rosaceae	462
4	Fabaceae	400
...	...	...
165	Teloschistaceae	1
166	Fissidentaceae	1
167	Pottiaceae	1
168	Lecanoraceae	1
169	Cladoniaceae	1

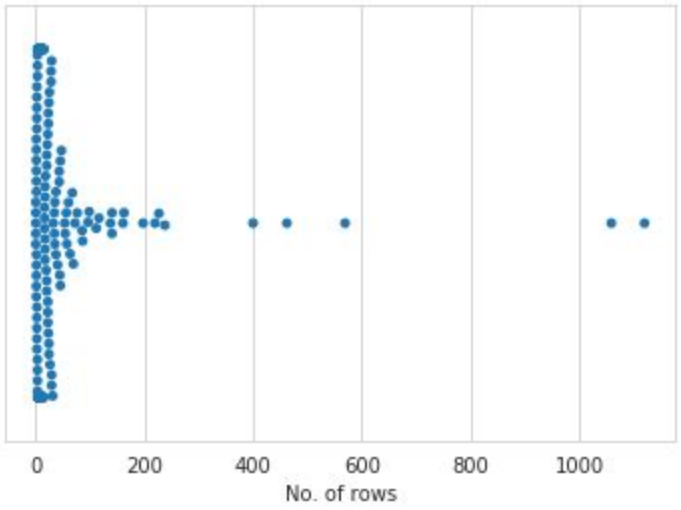
170 rows × 2 columns

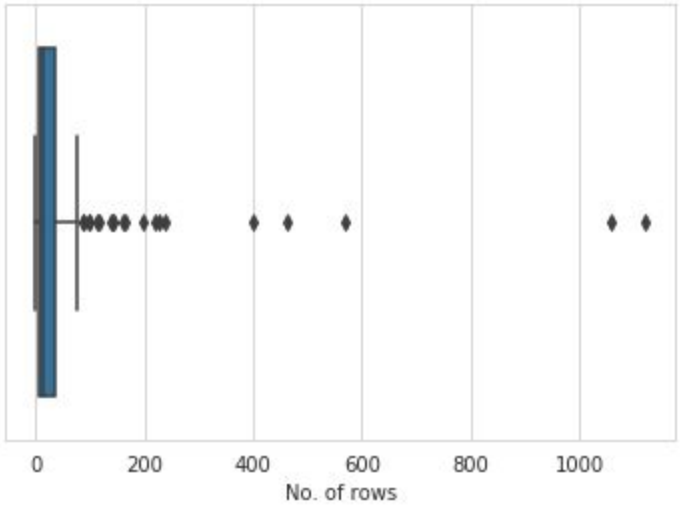
There are 2 families which dominate the dataset. Each by about 1/8th i.e 12.5%.



The index is taken on x-axis rather than the name for readability. A histogram (with bin size 100), swarm plot and box plot of these counts were also plotted. There are many families which have count less than 100.


160
140
120
100
80
60
40
20
0
Frequency
0
200
400
600
800
1000
No. of rows

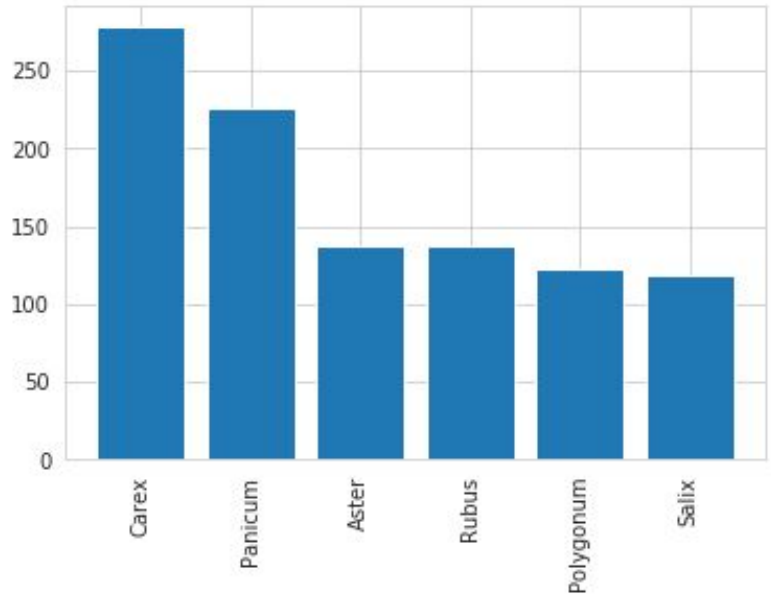

0
200
400
600
800
1000
No. of rows


0
200
400
600
800
1000
No. of rows

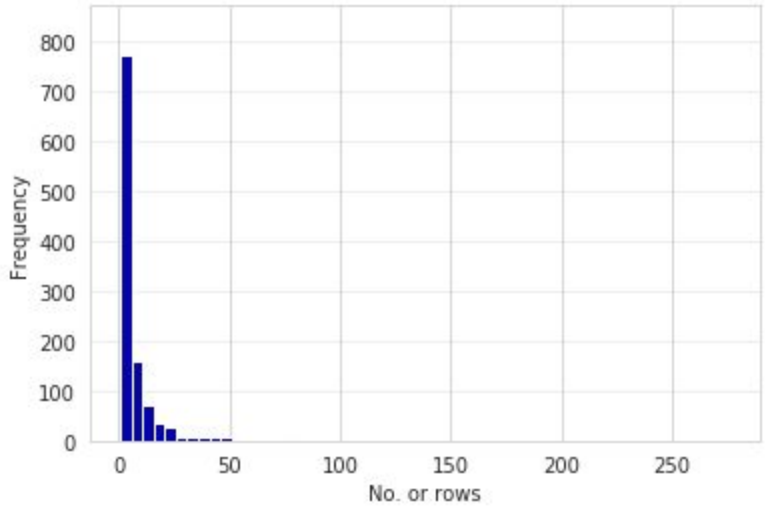
Pie charts were created for each family which showed the number of unique symbols which belonged to that family. These can be found in the folder titled ‘Family’.

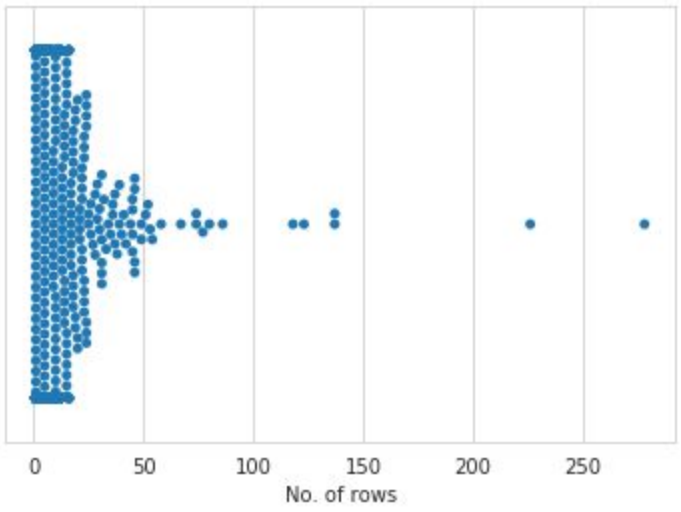
4. Genus

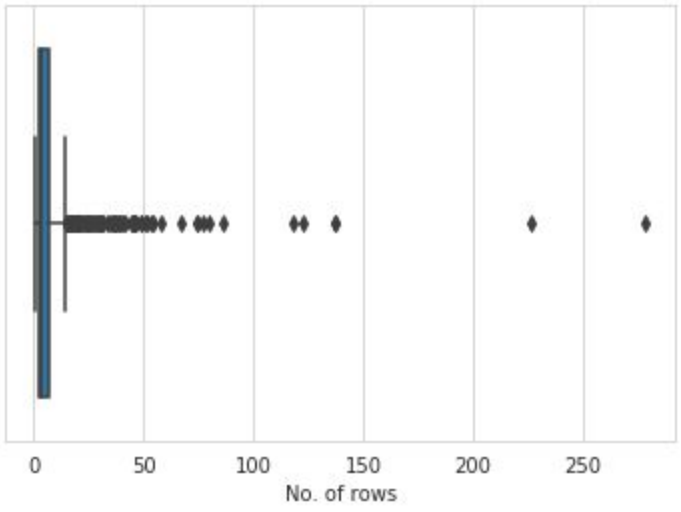
This column contains 1120 unique values. We’ll look at the values whose count is greater than 100. There are 6 such values.



Out of 1120, 771 genus occur 5 or less times. A histogram (with bin size 5), swarm plot and box plot are plotted to visualise this count.







Synonym Symbol and Scientific Name with Author are all unique values. So, no plots were done for these columns.

Predictive Analytics:

Feature Encoding:
Integer encoding is used since all the columns are categorical. We get the mean, median, deviation, etc from each column. This encoding would be required for training predictive models.

After Encoding:

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
0	1477	2909	3924	40	0	542
1	1477	1558	2477	254	0	339
2	1477	1559	2478	254	0	339
3	1477	2387	3925	254	0	542
4	1477	2407	3926	254	0	542
...	...	...	...	...	...	...
8413	3111	5288	8394	254	168	1111
8414	3111	5289	8395	254	168	1111
8415	1506	2909	3928	631	169	543
8416	1508	2909	3930	2363	169	543
8417	1508	2425	3929	254	169	543

8417 rows × 6 columns

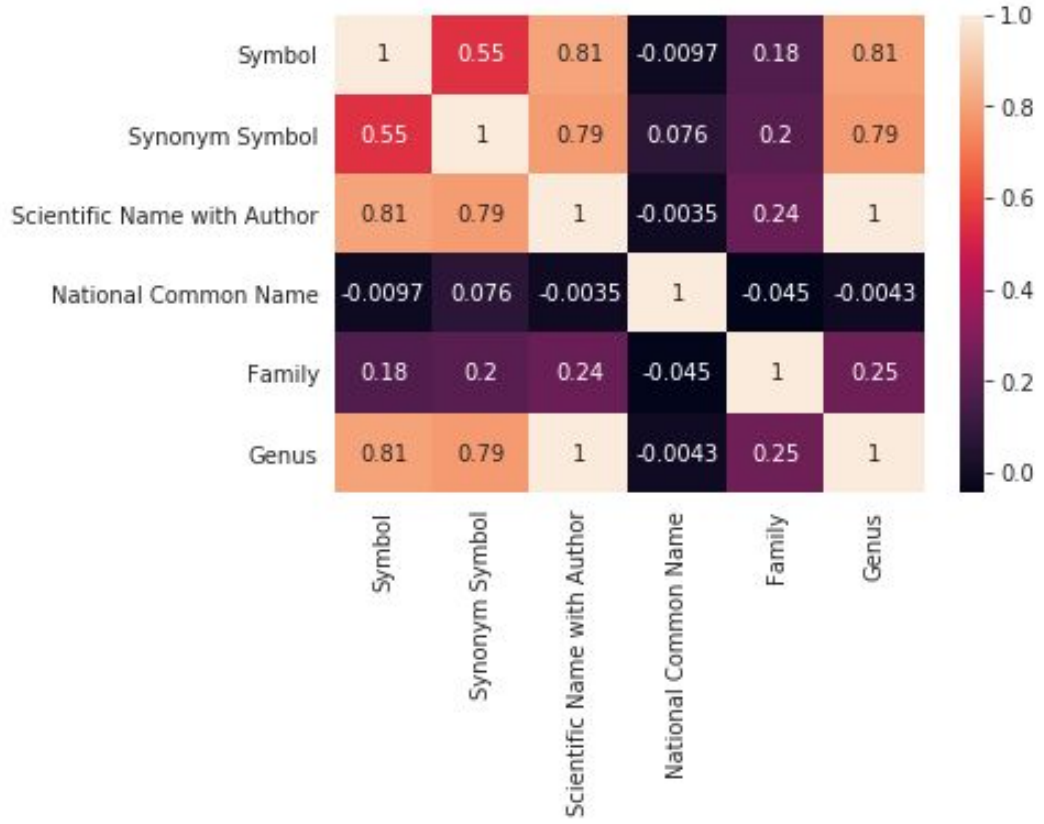
describe() output

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family	Genus
count	8417.000000	8417.000000	8417.000000	8417.000000	8417.000000	8417.000000
mean	1583.854105	2745.598551	4208.000000	634.737199	77.944992	575.547226
std	902.355110	1220.594496	2429.922941	670.139119	47.477936	325.137396
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	820.000000	2104.000000	2104.000000	254.000000	35.000000	284.000000
50%	1612.000000	2909.000000	4208.000000	254.000000	79.000000	594.000000
75%	2397.000000	3195.000000	6312.000000	835.000000	117.000000	868.000000
max	3118.000000	5299.000000	8416.000000	2553.000000	169.000000	1119.000000

info output():

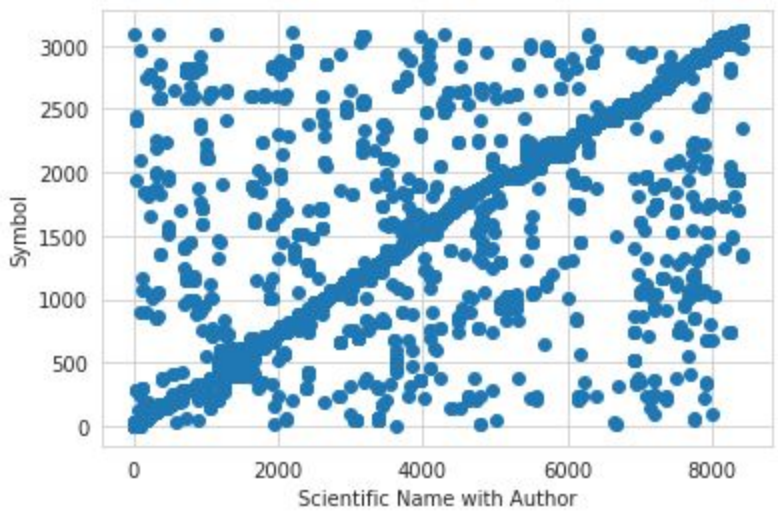
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8417 entries, 0 to 8417
Data columns (total 5 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   Symbol                      8417 non-null   int16
 1   Synonym Symbol              8417 non-null   int16
 2   Scientific Name with Author 8417 non-null   int16
 3   National Common Name        8417 non-null   int16
 4   Family                      8417 non-null   int16
dtypes: int16(5)
memory usage: 148.0 KB
```

Now, that the columns are encoded, we can check the correlation between the columns. This is shown in a heat map for better visualisation:

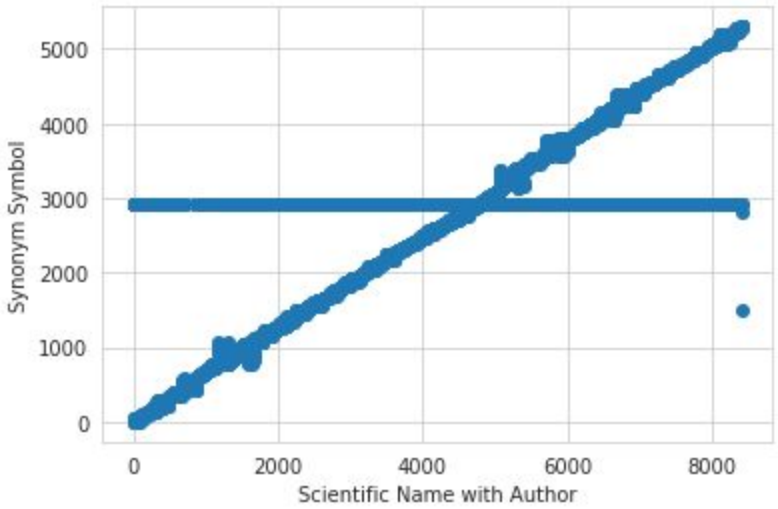


When doing predictive analytics, we should have a goal in mind. From this dataset, one possible objective would be to predict the family given the other variables. In order to be able to do predictive analytics, there should be a good correlation between the independent and dependent variables. But, there is a very low correlation between the other columns and the

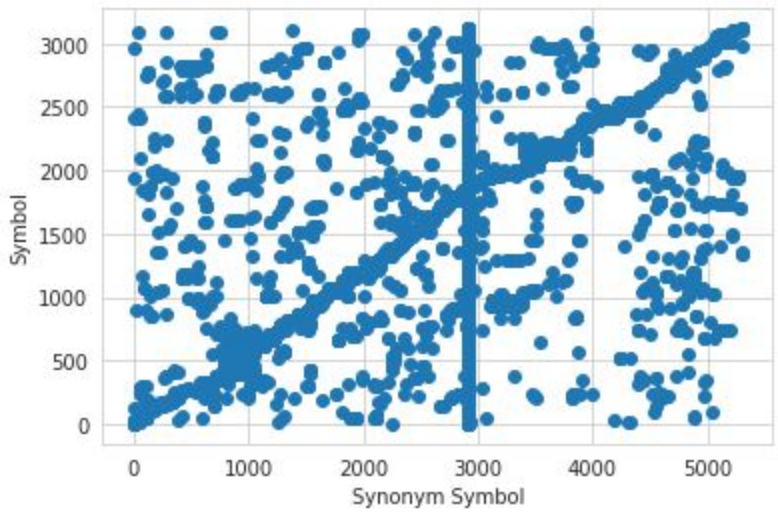
Family column. Scientific Name with Author shows a high correlation with Symbol and Synonym Symbol.



A scatter plot is plotted to observe the data. Scientific Name with Author is a column having unique values. While Symbol has values which repeat multiple times and which has only 3119 unique values. So, there would be an overlap in the values which results in the linear relationship.



Synonym Symbol is a column with unique values except for ‘NA’ which was encoded as 2909 and can be seen as the horizontal line cutting through the data. But if we ignore that, then the linear relationship is quite interesting as both the columns have unique values. This would be due to the method in which encoding was done. Since we used cat codes then the ordering and numbering was done in an alphabetical manner.



This looks very similar to Symbol vs Scientific Name with Author except for the addition of the NA line (vertical line). This makes sense as Synonym Symbol and Scientific Name with Author show a strong linear relationship.

Due to the poor correlation with Family, it does not seem that classification would be done very well.

Case 1:

- Feature columns= Symbol, Synonym Symbol, Genus and National Common Name
- Target= Family
- train-test split=0.4 and random state=1
- Naive Bayes classifier accuracy=0.194
- Decision tree classifier accuracy=0.214

As expected, the accuracy is very poor.

Case 2:

- Feature columns= Symbol, Synonym Symbol, Scientific Name with Author
- Target= Genus
- train-test split=0.3 and random state=1
- Naive Bayes classifier accuracy=0.787
- Decision tree classifier accuracy=0.139

Here Naive Bayes gives a high accuracy but this could be due to overfitting due to the perfect correlation between Scientific Name with Author and Genus. If we take that attribute out, then

Case 3:

- Feature columns= Symbol, Synonym Symbol
- Target= Genus
- train-test split=0.3 and random state=1
- Naive Bayes classifier accuracy=0.346

- Decision tree classifier accuracy=0.121

Again as expected the accuracy is poor.

Conclusion:

The column Scientific Name with Author should be broken down into Genus, Species, Taxonomic Rank, Author, etc. This would make the dataset more suitable for predictive modelling. As of now, there is not enough dependency between variables for prediction. This could be due to the method in which the null values were replaced and due to the encoding as well.

Association Rule Mining:

For association rule mining, the data does not need to be encoded, nor do we need to consider any missing values. So, we will use the dataset which had been obtained after removing the duplicate row. This dataset is:

	Symbol	Synonym Symbol	Scientific Name with Author	National Common Name	Family
0	JUAM	NaN	Justicia americana (L.) Vahl	American water-willow	Acanthaceae
1	JUAM	DIAM2	Dianthera americana L.	NaN	Acanthaceae
2	JUAM	DIAMS	Dianthera americana L. var. subcoriacea (Ferna...	NaN	Acanthaceae
3	JUAM	JUAMS	Justicia americana (L.) Vahl var. subcoriacea ...	NaN	Acanthaceae
4	JUAM	JUMO2	Justicia mortuifluminis Fernald	NaN	Acanthaceae
...	...	...	...	...	...
8412	ZAPA	ZAPAM	Zannichellia palustris L. var. major (Hartm.) ...	NaN	Zannichelliaceae
8413	ZAPA	ZAPAS	Zannichellia palustris L. var. stenophylla Asc...	NaN	Zannichelliaceae
8414	KALLS	NaN	Kallstroemia Scop.	caltrop	Zygophyllaceae
8415	KAPA	NaN	Kallstroemia parviflora J.B.S. Norton	warty caltrop	Zygophyllaceae
8416	KAPA	KAIN3	Kallstroemia intermedia Rydb.	NaN	Zygophyllaceae

8417 rows × 5 columns

But, since Scientific Name with Author and Synonym Symbol have unique values and there are no repeated items, we will not consider those columns. The columns that we will be using are ‘Symbol’, ‘National Common Name’, ‘Family’, ‘Genus’.

	Symbol	National Common Name	Family	Genus
count	8417	3115	8417	8417
unique	3119	2553	170	1120
top	CRCR2	hybrid violet	Asteraceae	Carex
freq	40	7	1119	278

NaN is not considered as an item when generating the transactions.

We cannot take the entire dataset as it would be computationally heavy. Instead we will check the frequent items for 2 classes. Among the families, there were 2 which dominated all the other families with more than 1000 rows:

1. Asteraceae - having 1119 rows

	Symbol	National Common Name	Genus
0	Symbol_ACHIL	NCN_yarrow	Genus_Achillea
1	Symbol_ACMI2	NCN_common_yarrow	Genus_Achillea
2	Symbol_ACMIO	NCN_western_yarrow	Genus_Achillea
3	Symbol_ACMIO	NaN	Genus_Achillea
4	Symbol_ACMIO	NaN	Genus_Achillea
...	...	...	...
1114	Symbol_YOTH	NCN_tall_false_hawksbeard	Genus_Youngia
1115	Symbol_YOTH	NaN	Genus_Crepis
1116	Symbol_YOTH	NaN	Genus_Youngia
1117	Symbol_YOTH	NaN	Genus_Youngia
1118	Symbol_YOUNG	NCN_youngia	Genus_Youngia

1119 rows × 3 columns

2. Poaceae - having 1058 rows

	Symbol	National Common Name	Genus
0	Symbol_AGCA5	NCN_colonial_bentgrass	Genus_Agrostis
1	Symbol_AGCA5	NaN	Genus_Agrostis
2	Symbol_AGCA5	NaN	Genus_Agrostis
3	Symbol_AGCA5	NaN	Genus_Agrostis
4	Symbol_AGCA5	NaN	Genus_Agrostis
...	...	...	...
1053	Symbol_VUOCO	NaN	Genus_Festuca
1054	Symbol_ZIAQ	NCN_annual_wildrice	Genus_Zizania
1055	Symbol_ZIAQA2	NCN_annual_wildrice	Genus_Zizania
1056	Symbol_ZIAQA2	NaN	Genus_Zizania
1057	Symbol_ZIZAN	NCN_wildrice	Genus_Zizania

1058 rows × 3 columns

Since, we are considering these 2 subsets, there will only be 3 columns ‘Symbol’, ‘National Common Name’, ‘Genus’. To make it easier to understand which item comes from which column, a string is appended to the beginning of each value.

FIM:

1. Apriori algorithm:

Class	S. no.	Min. Support Count	Min. confidence (to get interesting rules)	Frequent items					
				Length 1	Length 2 (NCN=National Common Name)			Length 3	Total
					Symbol & NCN	NCN & Genus	Symbol & Genus		
Asteraceae	1	2	-	359	0	66	178	0	603
	2	2	0.5	0	0	66	164	0	230
	3	3	-	221	0	10	107	0	338
	4	3	0.5	0	0	10	105	0	115
	5	5	-	118	0	0	53	0	171
	6	5	0.5	0	0	0	52	0	52
	7	20	0.8	0	0	0	3	0	3
Poaceae	1	2	-	315	0	45	190	0	550
	2	2	0.5	0	0	45	168	0	213
	3	3	-	215	0	8	120	0	343
	4	3	0.5	0	0	8	110	0	118
	5	3	0.7	0	0	8	94	0	102
	6	3	0.9	0	0	8	51	0	59
	7	20	0.8	0	0	0	2	0	2

2. FP Growth Algorithm

Class	S. no.	Minimum Support Count	Minimum confidence (to get interesting rules)	Frequent items					
				Length 1	Length 2 (NCN=National Common Name)			Length 3	Total
					Symbol & NCN	NCN & Genus	Symbol & Genus		
Asteraceae	1	2	-	350	0	66	178	0	594
	2	2	0.5	0	0	69	182	0	251
	3	3	-	219	0	10	107	0	117
	4	3	0.5	0	0	10	120	0	130
	5	5	-	117	0	0	53	0	170
	6	5	0.5	0	0	0	59	0	59
	7	20	0.8	0	0	0	4	0	4
Poaceae	1	2	-	297	0	45	190	0	532
	2	2	0.5	0	0	49	175	0	224
	3	3	-	205	0	8	120	0	333
	4	3	0.5	0	0	8	118	0	126
	5	3	0.7	0	0	8	96	0	104
	6	3	0.9	0	0	8	52	0	60
	7	20	0.8	0	0	0	2	0	2

The items and rules can be found in the ARM folder labelled as (S no.).txt.
Confidence measure was used to find interesting rules.
Note: The FP growth algorithm shows a slightly higher number because it distinguishes items like a,b from b,a.

Observations:

- There were no items of size 3 i.e from all the columns.
- When no confidence is applied, the majority of the items are of size 1.

- When confidence is applied, the items of size 1 are discarded, since confidence will give us the rules.
    - Among the rules of size 2, majority are between Symbol and Genus, while there are some frequent items between National Common Name and Genus, there are none between Symbol and National Common Name.
- As the minimum support count and minimum confidence increases, the number of items/rules decreases.
- In case of Asteraceae, if the minimum support is 5 then there are no rules between National Common Name and Genus.

CFI:

1. A-Close
2. CHARM

The itemsets were generated for Asteraceae and Poaceae for support counts 2, 3, 5 and 20 and the outputs are in files titled 1, 2, 3 and 4 respectively.

MFI:

1. MAFIA
2. Pincer Search

The itemsets were generated for Asteraceae and Poaceae for support counts 2, 3, 5 and 20 and the outputs are in text files titled 1, 2, 3 and 4 respectively.

Part B

Explore various online resources to identify other measures of rules and pattern evaluation. Test drive these various measures for any one of the above itemset types.

Let:

X,Y	be itemsets
X ⇒ Y	an association rule
T	a set of transactions of a given database.

In order to select interesting rules from the set of all possible rules, the best-known constraints are minimum thresholds on support and confidence.

- Support:
  Support is a so-called frequency constraint. Its main feature is that it possesses the property of down-ward closure which means that all subsets of a frequent set (support > min. support threshold) are also frequent.

  $$sup(X \rightarrow Y) = sup(Y \rightarrow X) = P(X \text{ and } Y)$$

- Confidence:
  The confidence value of a rule, X ⇒ Y, with respect to a set of transactions T, is the proportion of the transactions that contain X which also contains Y.

  $$conf(X \rightarrow Y) = P(Y \mid X) = P(X \text{ and } Y)/P(X) = sup(X \rightarrow Y)/sup(X)$$

Some other measures are:

- Leverage:
  Leverage measures the difference of X and Y appearing together in the data set and what would be expected if X and Y were statistically dependent.

  $$leverage(X \rightarrow Y) = P(X \text{ and } Y) - (P(X)P(Y))$$

- Conviction:
  Conviction compares the probability that X appears without Y if they were dependent with the actual frequency of the appearance of X without Y.

  $$conviction(X \rightarrow Y) = P(X)P(\text{not } Y)/P(X \text{ and not } Y)=(1-sup(Y))/(1-conf(X \rightarrow Y))$$

- Lift:
  Lift is the ratio of the observed support to that expected if X and Y were independent.

$$\begin{aligned} lift(X \rightarrow Y) &= lift(Y \rightarrow X) \\ &= P(X \text{ and } Y)/(P(X)P(Y)) \\ &= conf(X \rightarrow Y)/sup(Y) \\ &= conf(Y \rightarrow X)/sup(X) \end{aligned}$$

- Coverage: A simple measure of how often an item set appears in the data set.

$$coverage(X) = P(X) = sup(X)$$

For the apriori algorithm, minimum lift can be applied to prune the rules. A lift of 1 indicates that there is no association between X and Y. So we'll try different minimum lifts and see how the rules get pruned. For each rule the conviction, leverage and coverage are also computed.

We'll start with the itemset produced by applying Apriori algorithm on Asteraceae class with minimum support count=2 and minimum confidence=0.5.

The initial values before adding lift:

- Length 1=0
- Length 2=0
  - Symbol & NCN=0
  - NCN & Genus=66
  - Symbol & Genus=164
- Length 3=0
- Total=230

Class	S. no.	Minimum Lift	Frequent items			
			Length 2 (NCN=National Common Name)			Total
			Symbol & NCN	NCN & Genus	Symbol & Genus	
Asteraceae	1	2	0	66	164	230
	2	5	0	66	158	224

	3	10	0	66	141	207
	4	50	0	35	72	107
	5	100	0	22	37	59
	6	200	0	14	16	30
	7	400	0	0	2	2

The rules are found in the B folder with (S no).txt as file name.

Observations:

- We get a different set of rules if lift is changed compared to when support is changed.
- A rule may have very high lift but very low support

Part C

Test drive Decision tree, Bayes Classification using appropriate packages from libraries in the platform of your choice. Also, construct a confusion matrix and evaluate the performance of your classifier. You may choose any open source dataset for classification.

Decision Tree and Bayes Classification were implemented on diabetes dataset and iris dataset.

1. Check the dataset. The dataset is already clean and prepared for classification, so no processing is required.
2. Assign the feature columns and the target column. Here the target would be the outcome (1=Person has diabetes, 2=Person doesn't have diabetes).
3. Split data into training and testing sets. The random state of 1 ensures that when we run the program again with the same random state value then we'll get the same output.
4. Decision Tree:
    a. Generate decision tree model using scikit-learn. Scikit-learn comes with a prebuilt function called DecisionTreeClassifier().
    b. Input the training data using fit()
    c. Input the test data using predict()
5. Naive Bayes:
    a. Generate the model using scikit-learn. Scikit-learn has a prebuilt function GaussianNB(). Parameters like prior probability or variance for smoothing can be passed to it. We did not calculate it, so nothing was passed to it.
    b. Input the training data using fit()
    c. Input the test data using predict()
6. Evaluate the model. For evaluation of the classifier we are using:
    a. accuracy_score() from metrics package of scikit-learn. This will tell the accuracy percentage.
    b. Along with that we'll also give the total number of mislabelled points.
    c. Display the confusion matrix.
    d. Use classification_report() which will give
        i. Precision: Accuracy of positive predictions. What percent of predictions are correct?

$$\text{Precision} = TP/(TP + FP)$$

        ii. Recall: Fraction of positives that were correctly identified. What percent of positive cases did you identify?

$$\text{Recall} = TP/(TP+FN)$$

        iii. F1-score: What percent of positive predictions were correct?

$$\text{F1 Score} = 2*(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

        iv. Support: The number of actual occurrences of the class in the specified dataset

Example output:

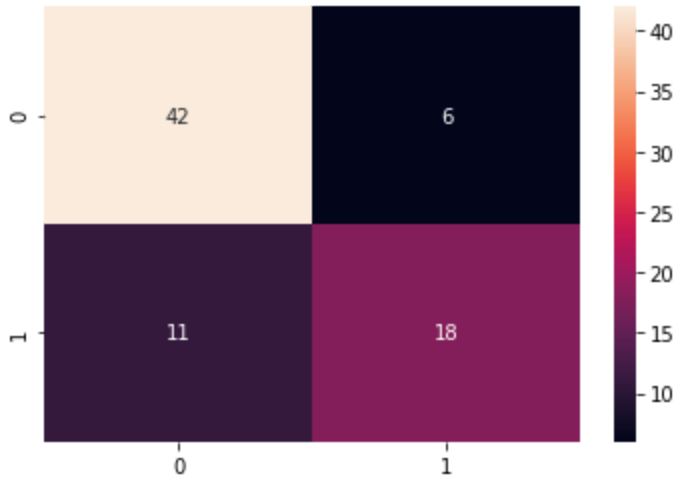
(For diabetes.csv, decision tree classifier when train test split is 0.1 and depth is 3)

```
Accuracy: 0.7792207792207793
Number of mislabeled points out of a total 77 points : 17
              precision    recall  f1-score   support

           0       0.79      0.88      0.83        48
           1       0.75      0.62      0.68        29

    accuracy                           0.78        77
   macro avg       0.77      0.75      0.76        77
weighted avg       0.78      0.78      0.77        77

Confusion Matrix:
[[42  6]
 [11 18]]
```



Run the program C.py for below measures to view the confusion matrix and performance of each.

Diabetes.csv

1. Decision Tree:

S. no.	Depth	Train Test split	Accuracy
1	3	0.1	0.779
2	3	0.2	0.798

3	3	0.3	0.770
4	3	0.4	0.733
5	4	0.1	0.766
6	4	0.2	0.792
7	4	0.3	0.783
8	2	0.1	0.792
9	2	0.2	0.733

- Changing the depth doesn't change the accuracy by that much.
- Best accuracy is found when the train test split is 0.2 and depth is 3.

2. Naive Bayes

S. no.	Train Test split	Accuracy
1	0.1	0.792
2	0.2	0.772
3	0.3	0.783
4	0.4	0.727

- Best performance is seen when train test split is 0.1.
- Should not increase train test split further than 0.4, so we stop at 0.4.
- Both Decision Tree and Naive Bayes show similar performance for this dataset.

Iris Dataset

1. Decision Tree:

S. no.	Depth	Train Test split	Accuracy
1	3	0.1	1
2	3	0.2	0.966

3	3	0.3	0.955
4	3	0.4	0.966

- Changing the depth did not change the accuracy.
- Changing the train test split did not have a huge impact on accuracy.
- Showed higher accuracy than diabetes dataset. Could be due to the quality of the data.

2. Naive Bayes

S. no.	Train Test split	Accuracy
1	0.1	1
2	0.2	0.966
3	0.3	0.933
4	0.4	0.95

- Accuracy is not much different from decision tree.
- In both cases for train test split of 0.1 shows 100% accuracy. This could be due to overfitting.