DATA MINING ASSIGNMENT-01

Venkata Krishna Mohan Sunkara
Zetao Zhao

1. Summary of the Dataset:

This dataset is obtained from Bruce Bulloch, New Zealand. Some instances have no data recorded due to experimental recording procedures required to treat the absence of a species at a site as a missing value. The data is collected using eucalypt trial methods.

i. Purpose:

The dataset consists of information regarding some contributing factors of seedlots in a species which determine the eucalyptus soil conservation in seasonally dry hill country. The extent to which these seedlots can be useful for determining the soil conservation is provided as Utility (class) attribute. The main purpose of this dataset is to represent the utility of a seedlot (as a class distribution) based on several contributing factors like latitude, altitude, Rainfall, species type, insect resistance etc...,

ii. Attributes:

A total of 19 attributes are used as factors and one attribute is used to represent the class. Out of these 19 attributes, 14 attributes are numeric and 5 are nominal.

Note: Weka considers the class label as an attribute. So, Weka displays a total of 20 attributes.

Attribute Name	**Description of Attribute**	**Type of attribute**
Abbrev	Site abbreviation	Nominal (enumerated)
Rep	Site rep	Numeric (Integer)
Locality	Site locality in North Island	Nominal (enumerated)
Map_Ref	Map location in the North Island	Nominal (enumerated)
Latitude	Latitude approximation	Nominal (enumerated)
Altitude	Altitude approximation	Numeric (Integer)
Rainfall	Rainfall (mm pa)	Numeric (Integer)
Frosts	Frosts (deg c)	Numeric (Integer)
Year	Year of planting	Numeric (Integer)
Sp	Species code	Nominal (enumerated)
PMCno	Seedlot number	Numeric (Integer)
DBH	Best diameter base	Numeric (real)

	height (cm)	
Ht	Height (m)	Numeric (real)
Surv	Survival	Numeric (Integer)
Vig	Vigour	Numeric (real)
Ins_res	Insect resistance	Numeric (real)
Stem_Fm	Stem form	Numeric (real)
Crown_Fm	Crown form	Numeric (real)
Brnch_Fm	Branch form	Numeric (real)
Utility	Utility rating	Nominal (enumerated)

Table 01: Attributes and their types.

iii. Nominal Attributes:

The values for nominal attributes in the data set is described below:

a) Abbrev:
{Cra, Cly, Nga,Wai,K81,Wak,K82,WSp,K83,Lon,Puk,Paw,K81a,Mor,Wen,WSh}
b) Locality:
{Central_Hawkes_Bay,Northern_Hawkes_Bay,Southern_Hawkes_Bay,Central_Hawkes_Bay_(coastal),Central_Wairarapa,South_Wairarapa,Southern_Hawkes_Bay_(coastal),Central_Poverty_Bay}
c) Map_Ref:
{N135_382/137,N116_848/985,N145_874/586,N142_377/957,N158_344/626,N162_081/300,N158_343/625,N151_912/221,N162_097/424,N166_063/197,N146_273/737,N141_295/063,N98_539/567,N151_922/226}
d) Latitude:
{39__38,39__00,40__11,39__50,40__57,41__12,40__36,41__08,41__16,40__00,39__43,82__32}
e) Sp:
{co,fr,ma,nd,ni,ob,ov,pu,rd,si,mn,ag,bxs,br,el,fa,jo,ka,re,sm,ro,nc,am,cr,pa,ra,te}
f) Utility:
{none, low, average, good, best}

iv. Instances:

There is a total of 736 instances in the dataset.

2. Attributes Exploration:

This section explores the attributes provided in the dataset.

i. Nominal Attributes:

There are 6 Nominal attributes in total including the utility class. The details about a particular attribute can be seen by clicking on an attribute and its distribution and details are displayed in the Selected attribute area of Weka tool.

a.) Abbrev:

There are 16 different labels for the abbrev attribute.

Name: Abbrev Type: Nominal
Missing: 0 (0%) Distinct: 16 Unique: 0 (0%)

No.	Label	Count	Weight
1	Cra	30	30.0
2	Cly	24	24.0
3	Nga	22	22.0
4	Wai	70	70.0
5	K81	65	65.0
6	Wak	73	73.0
7	K82	45	45.0
8	WSp	59	59.0
9	K83	49	49.0
10	Lon	53	53.0
11	Puk	84	84.0
12	Paw	55	55.0
13	K81a	33	33.0
14	Mor	63	63.0
15	Wen	6	6.0
16	WSh	5	5.0

Fig 01: abbrev attributes distribution.

b.) Locality:

There are 8 different labels for the Locality attribute.

Name: Locality Type: Nominal
Missing: 0 (0%) Distinct: 8 Unique: 0 (0%)

No.	Label	Count	Weight
1	Central_Hawkes_...	93	93.0
2	Northern_Hawke...	24	24.0
3	Southern_Hawke...	86	86.0
4	Central_Hawkes_...	70	70.0
5	Central_Wairarapa	192	192.0
6	South_Wairarapa	210	210.0
7	Southern_Hawke...	55	55.0
8	Central_Poverty_...	6	6.0

Fig 02: Locality attribute distribution

c.) Map_Ref:

There are 14 different labels for the Map_ref attribute:

Name: Map_Ref Type: Nominal
Missing: 0 (0%) Distinct: 14 Unique: 0 (0%)

No.	Label	Count	Weight
1	N135_382/137	30	30.0
2	N116_848/985	24	24.0
3	N145_874/586	22	22.0
4	N142_377/957	70	70.0
5	N158_344/626	147	147.0
6	N162_081/300	73	73.0
7	N158_343/625	45	45.0
8	N151_912/221	59	59.0
9	N162_097/424	53	53.0
10	N166_063/197	84	84.0
11	N146_273/737	55	55.0
12	N141_295/063	63	63.0
13	N98_539/567	6	6.0
14	N151_922/226	5	5.0

Fig 03: Map_Ref attribute distribution

d.) Latitude:

There are 12 different labels for Latitude attribute:

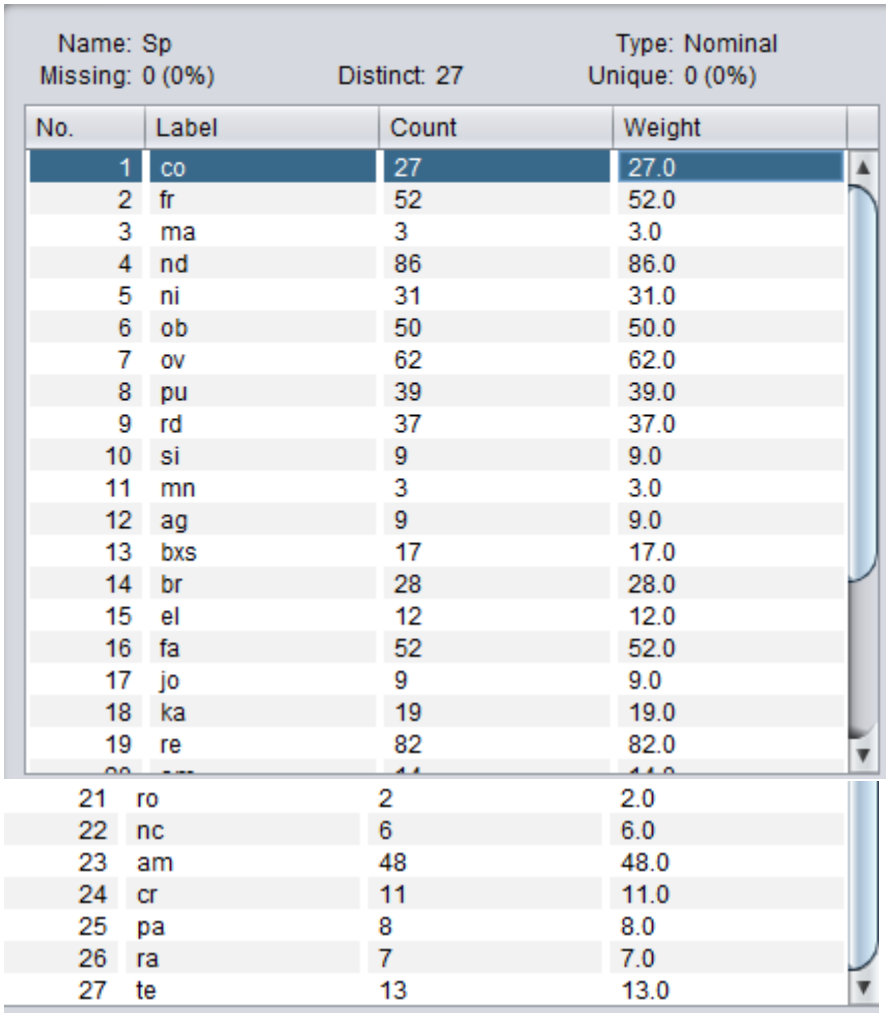
Name: Latitude Type: Nominal
Missing: 0 (0%) Distinct: 12 Unique: 0 (0%)

No.	Label	Count	Weight
1	39__38	30	30.0
2	39__00	24	24.0
3	40__11	22	22.0
4	39__50	70	70.0
5	40__57	192	192.0
6	41__12	73	73.0
7	40__36	64	64.0
8	41__08	53	53.0
9	41__16	84	84.0
10	40__00	55	55.0
11	39__43	63	63.0
12	82__32	6	6.0

Fig 04: Latitude attribute distribution.

e.) Species Code:

There are 27 different labels for Sp attribute.



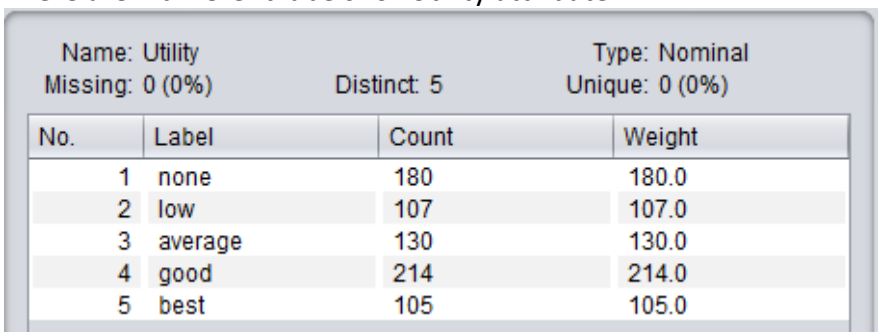
Name: Sp
Missing: 0 (0%)
Distinct: 27
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	co	27	27.0
2	fr	52	52.0
3	ma	3	3.0
4	nd	86	86.0
5	ni	31	31.0
6	ob	50	50.0
7	ov	62	62.0
8	pu	39	39.0
9	rd	37	37.0
10	si	9	9.0
11	mn	3	3.0
12	ag	9	9.0
13	bxs	17	17.0
14	br	28	28.0
15	el	12	12.0
16	fa	52	52.0
17	jo	9	9.0
18	ka	19	19.0
19	re	82	82.0
21	ro	2	2.0
22	nc	6	6.0
23	am	48	48.0
24	cr	11	11.0
25	pa	8	8.0
26	ra	7	7.0
27	te	13	13.0

Fig 05,06: Sp attribute distribution.

f.) Utility:

There are 4 different labels for Utility attribute:



Name: Utility
Missing: 0 (0%)
Distinct: 5
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	none	180	180.0
2	low	107	107.0
3	average	130	130.0
4	good	214	214.0
5	best	105	105.0

Fig 07: Utility attribute distribution.

We can visualize all the attributes by using the visualize all tab. The visualization of all the attributes is shown below:

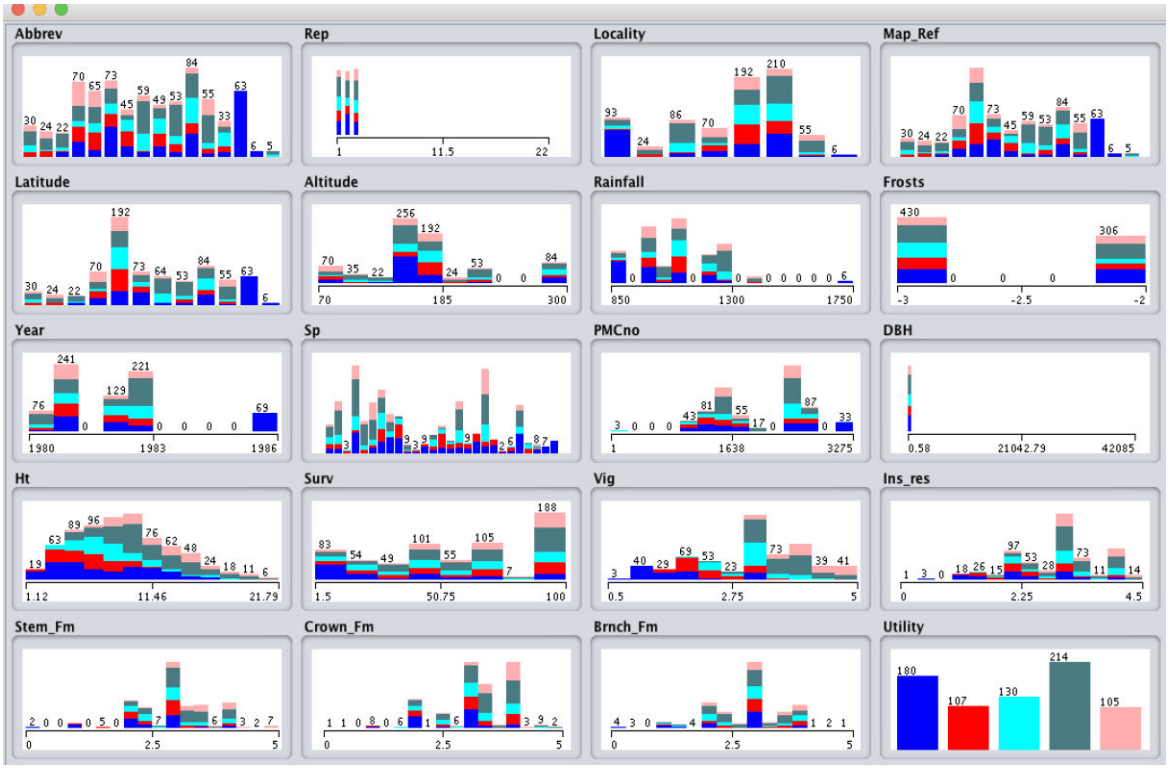


Fig 07.1: Visualization of all attributes

ii. Summary of Attributes:

a.) Numeric Attributes:

The attribute type is also displayed in the Selected attribute area of Weka tool. So, we selected all the attributes one at a time and summarized the numeric attributes in to a table. The Selected Area showed the min, max, mean and stdDev of each attribute. We calculated variance by taking square of stdDev.

Name	Min	Max	Mean	Variance	Std Dev	Missing
Rep	1	22	2.026	1.128816	1.104	0
Altitude	70	300	172.024	3506.179	59.213	0
Rainfall	850	1750	1095.938	20975.7289	144.83	0
Frosts	-3	-2	-2.584	0.243049	0.493	0
Year	1980	1986	1982.141	2.524921	1.589	0
PMCno	1	3275	2054.739	382468.0336	618.44	7(1%)
DBH	0.58	42085	72.947	2408021.1684	1551.78	1(0%)
Ht	1.12	21.79	9.295	16.120225	4.105	1(0%)
Surv	1.5	100	59.675	955.675396	30.914	94(13%)
Vig	0.5	5	3.076	1.026169	1.013	69(9%)
Ins_res	0	4.5	2.897	0.667489	0.817	69(9%)
Stem_Fm	0	5	2.996	0.509796	0.714	69(9%)

Crown_Fm	0	5	3.204	0.564001	0.751	69(9%)
Brnch_Fm	0	5	2.841	0.620944	0.788	69(9%)

Table 02: Numeric Attributes Summary

b.) Nominal Attributes:

If we summarize only the Nominal Attributes in the dataset, we get a total of 6 attributes.

Name	Number of Labels	Less frequent Label Name	More frequent Label Name	Missing
Abbrev	16	Wsh	Puk	0
Locality	8	Central_Poverty_bay	South_Wairarapa	0
Map_Ref	14	N151_922/226	N158_344/626	0
Latitude	12	82_32	40_57	0
Sp	27	ro	nd	0
Utility	5	Best	good	0

Table 03: Nominal Attributes Summary

iii. To add a new attribute which is an index of instances, we need to select the choose button in the Weka tool. Now we select the filters → unsupervised →attribute →AddID. Now a get the following screen:

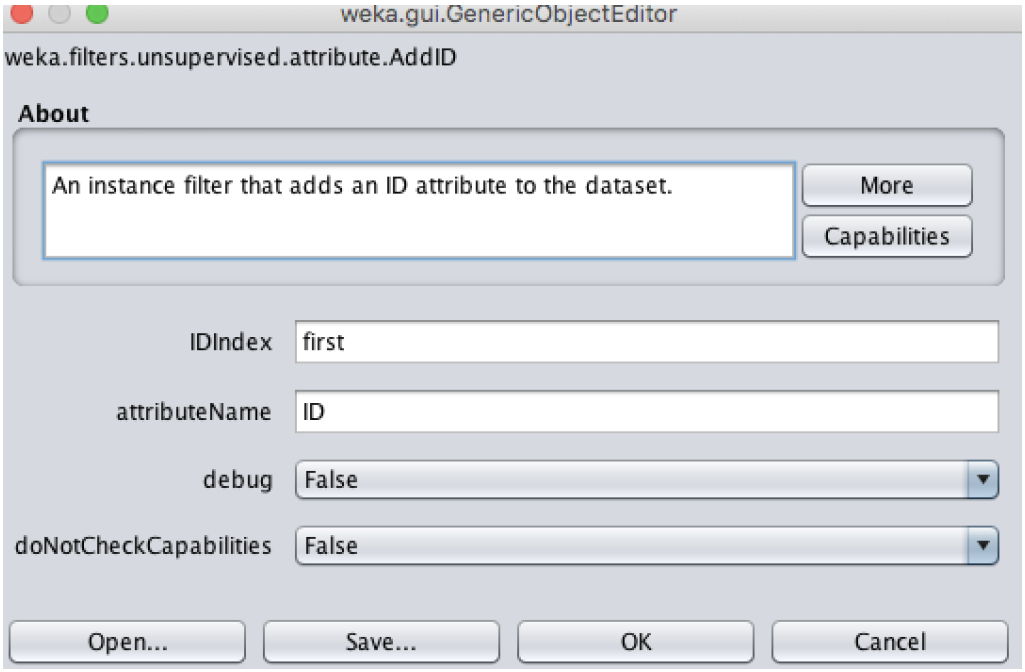


Fig 08: Add ID filter.

Now, after adding the ID we get the new attribute added to the list of attributes and we notice the number of attributes to be 21:

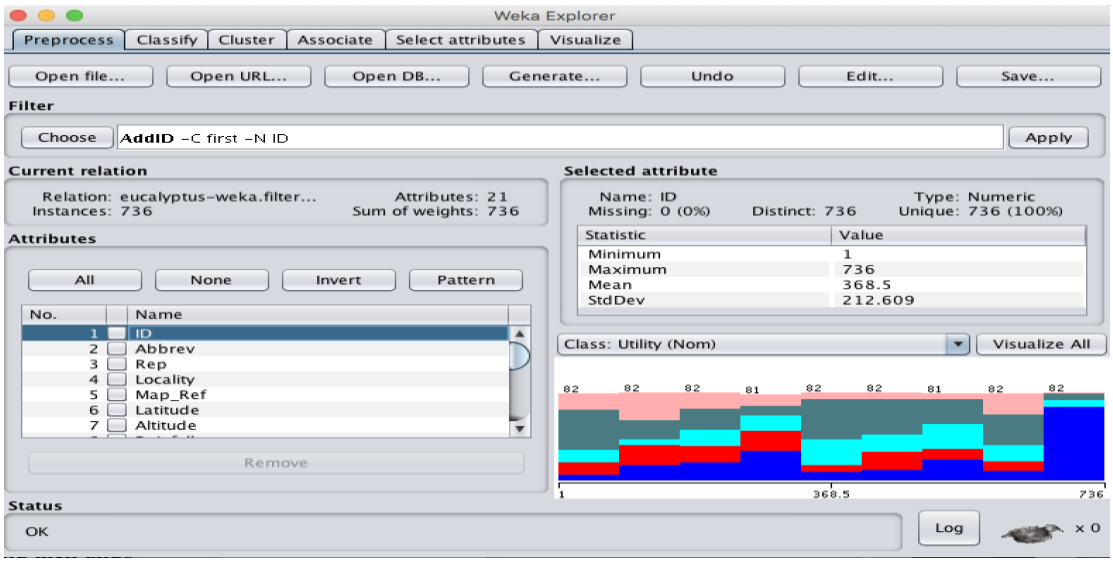


Fig 09: ID attribute distribution.

Now, if we need to examine the distribution of each feature we can click on the visualize tab in the menu and select the graph which we want to visualize. The rows in the pair plot represent y-axis and the columns represent the x-axis. For, Example if we want to visualize 'ID' as x-axis and 'Ht' as Y-axis we can simply scroll on the pair plot to that corresponding row (8^{th}) and column (1^{st}). Now if we press that particular plot a new window pops up which shows the graph in much more detail. Now the window has drop downs from which we can select the required attribute on x-axis and required attribute on y-axis.

The Pair plot looks like:

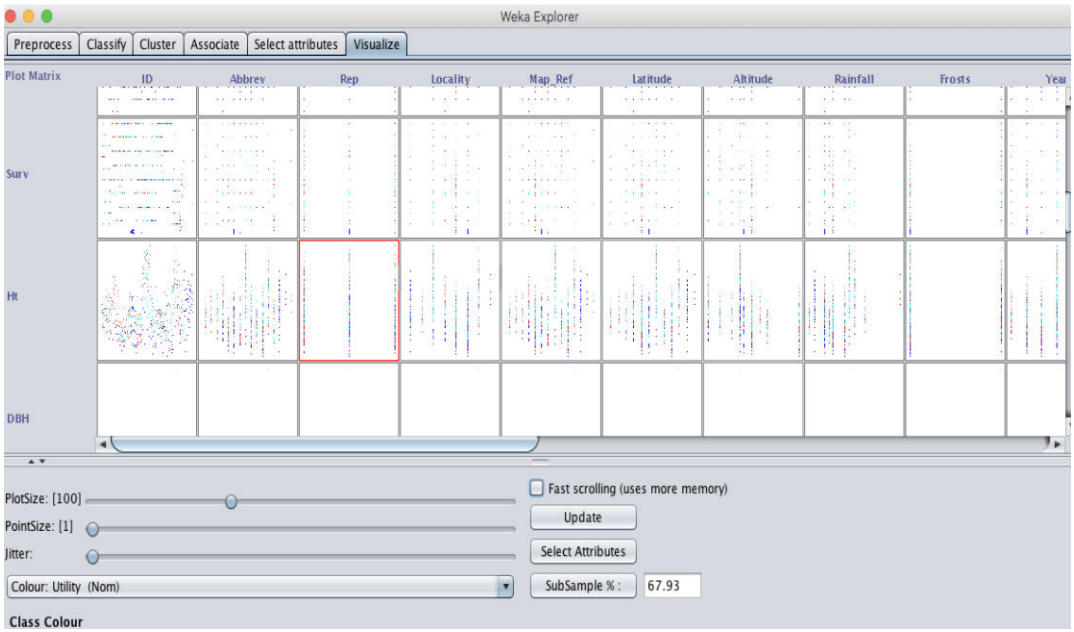


Fig 10: Pair plot for all attributes.

The plot of ‘ID’ on x-axis and ‘Ht’ on Y-axis looks like this: The colors in this plot represents the utility class for that point (datum).

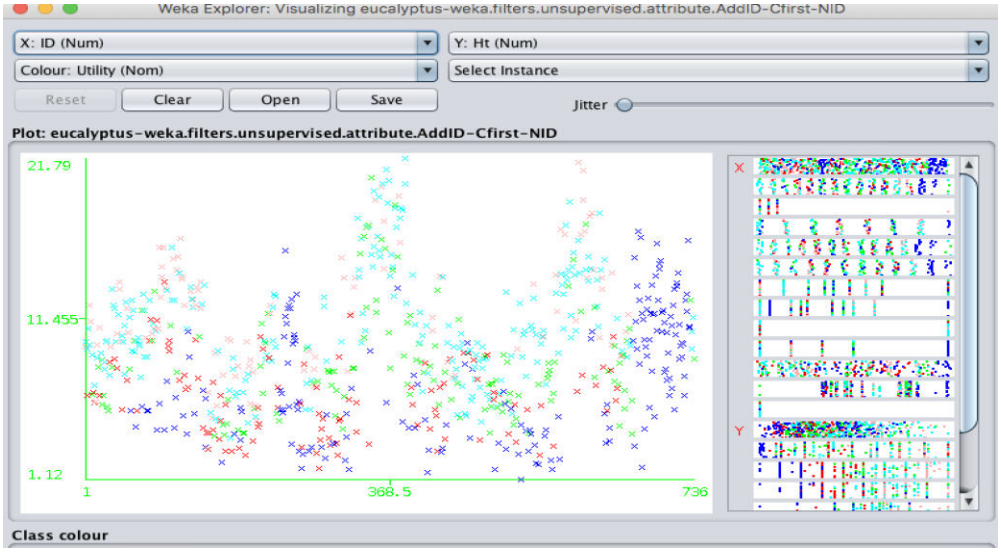


Fig 11: Visualization of ID vs Ht

Now let us select ‘ID’ as x-axis and a nominal attribute ‘Locality’ on Y-axis:

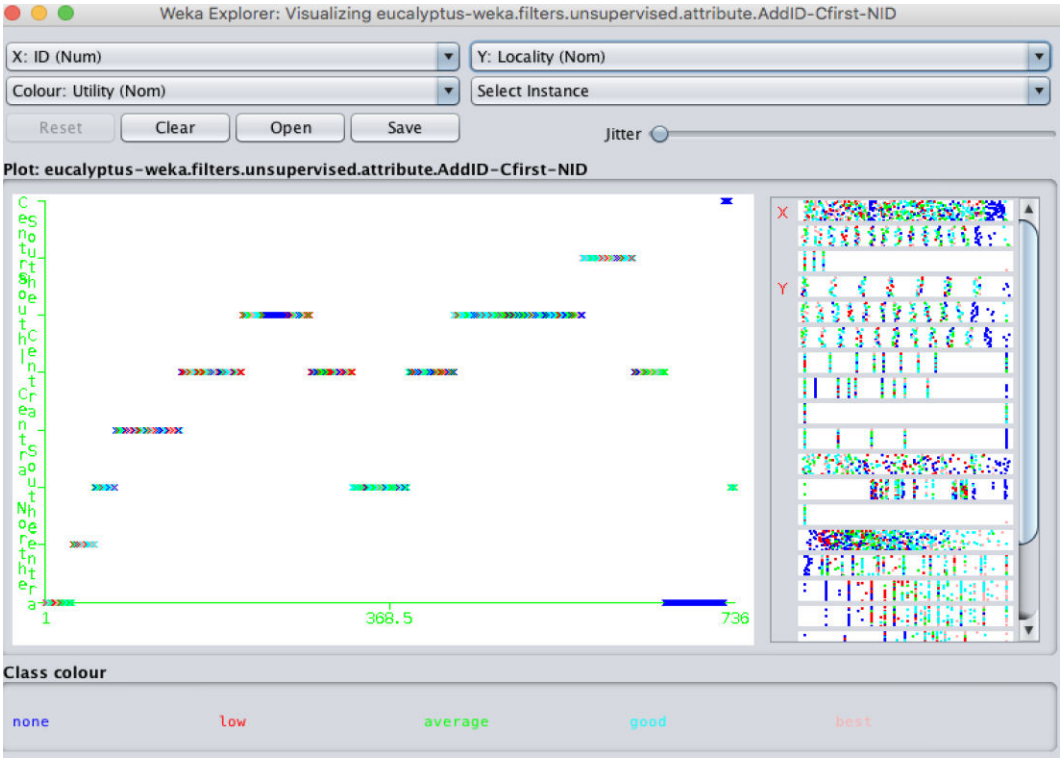


Fig 12: Visualization of ID vs Locality

The above Plot represents ID vs Locality and we can see that the distribution of points is concentrated along 8 distinct values of Y because there are 8 different labels for locality attribute.

iv. a.)

The cross tabulations for pairwise attributes is displayed in fig 10. Now let us visualize Rainfall (Numeric) vs Locality (Locality). We can select the x-axis as Rainfall and Locality as y-axis. The obtained plot is displayed in Fig 13:

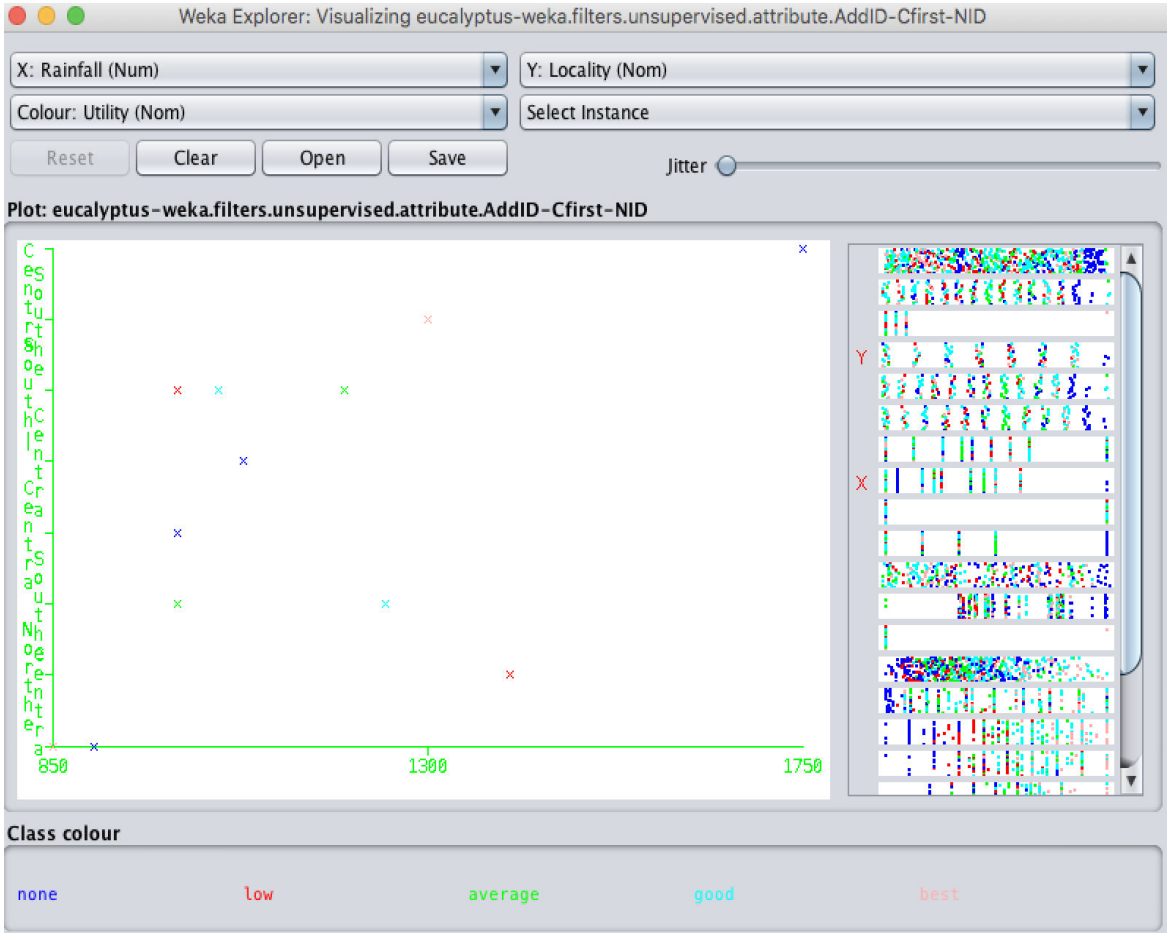


Fig 13: Visualization of Rainfall x Locality

Summary of Visualization: Now we notice that the plot is sparse and the x-axis is represented by Rainfall with values ranging from 850-1750 and the Y-axis is represented by locality and has 8 distinct labels. Each point the graph has several instances of data. It means several instances in the data has rainfall and Locality attribute values equal and their class value is also equal. From this, we may assume that there might be an association between rainfall, locality and utility.

We can obtain the actual instances at a point by simply clicking on the point. If we click on a point then a new window appears which represents the actual instances of data that are at that point.

b.)
Now let us visualize the plot between Crown_Fm and Surv attributes. If we select the Crown_Fm as X-axis and Surv as Y-axis we get the following graph as displayed in fig 14.

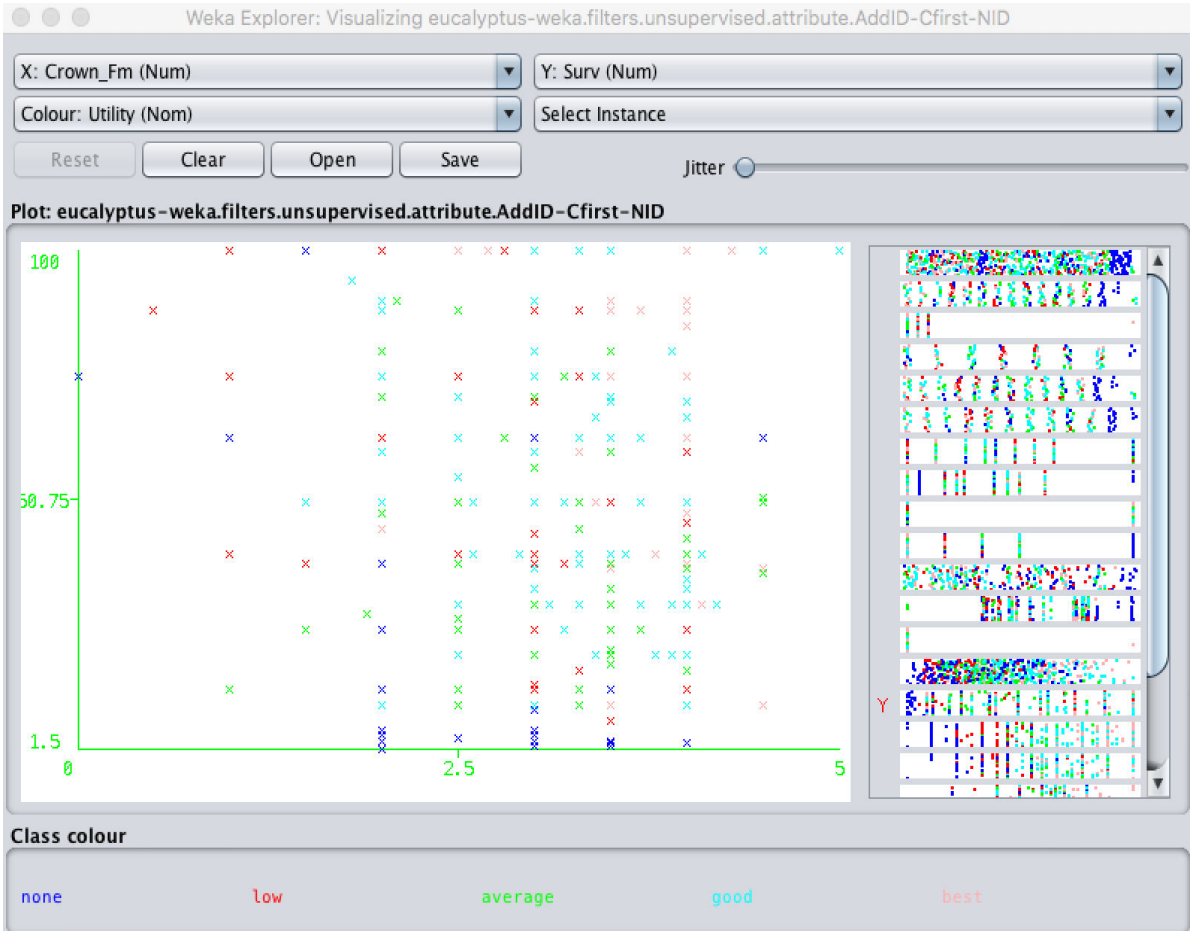


Fig 14: Visualization of Crown_Fm x Surv

Summary of Visualization: Now we notice that the plot is somewhat sparse because both attributes are numerical. We notice the range of X-axis is from 0-5 and the range of Y-axis is from 1.5-100 which are the minimums and maximums of Crown_Fm and Surv attributes respectively. The Class attribute (Utility) is displayed in terms of colors for each point. We also notice that for each value of Crown_Fm there are different values of Surv associated with it. From the plot, it seems that there isn't any correlation among the attributes.

3. Data Cleanup and Processing:

This Section performs some data cleaning and pre-processing.

i. Removing the Latitude attribute:

This can be achieved by selecting the attribute name from the list and simply clicking the remove button at the end of attribute list. Now in the Fig 15, we can notice that the number of attributes to be reduced to 20 (previously it was 21 because of addition of ID attribute). This can also be achieved by choosing the Remove filter from weka →filters→ Unsupervised → Attribute → Remove → providing the attribute indices as 6.

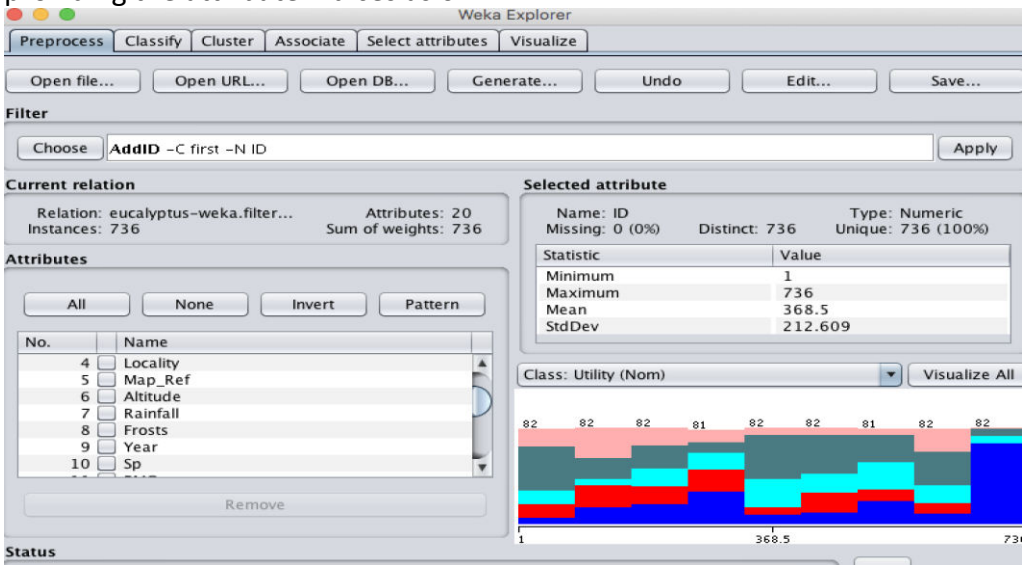


Fig 15: After Removal of Latitude attribute

Now we can undo it by clicking on the undo button and observe that the Latitude attribute is restored.

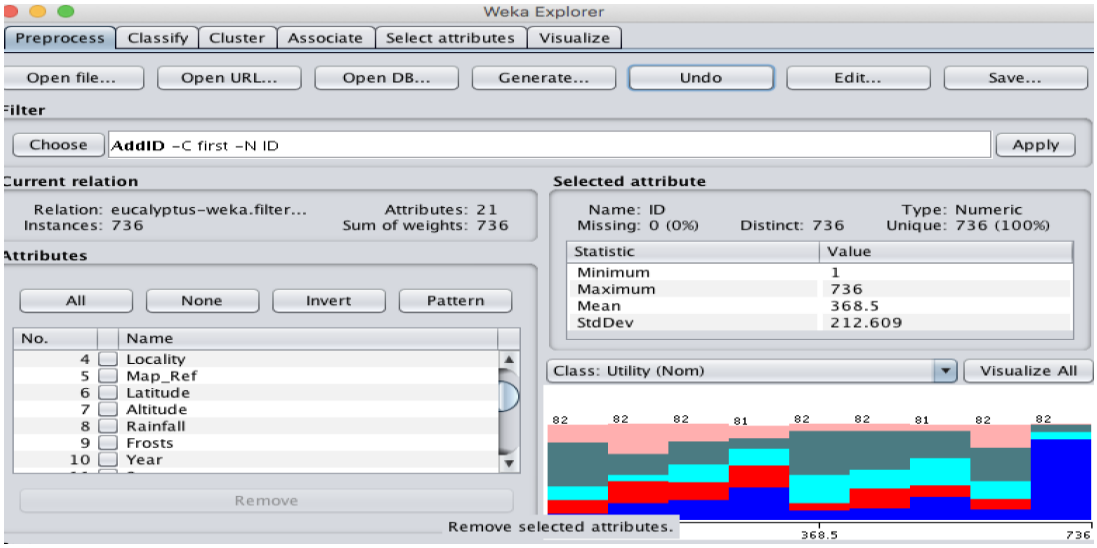


Fig 16: After Undoing the previous action

ii. Adding the LogRainfall Attribute:

This can be achieved by selecting the AddExpression filter from Weka → filters → Unsupervised → Attribute → AddExpression. Now if we left click on the filter name, a window pops up which is displayed in fig 17. Now we enter the expression as log(a8) where 8 is the index number of Rainfall attribute and apply it.

Now we can observe the distribution of the newly added attribute LogRainfall in fig 18.

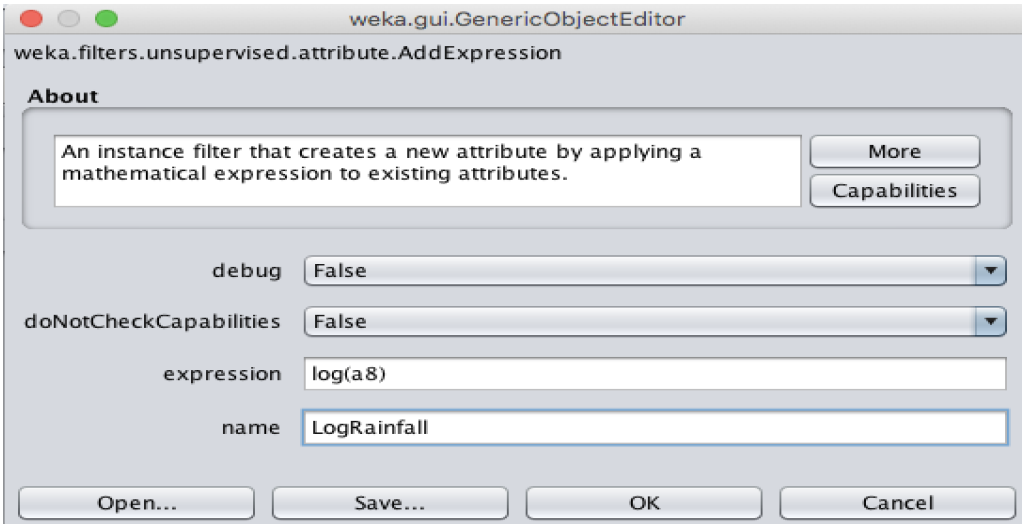


Fig 17: Adding the Log Rainfall attribute

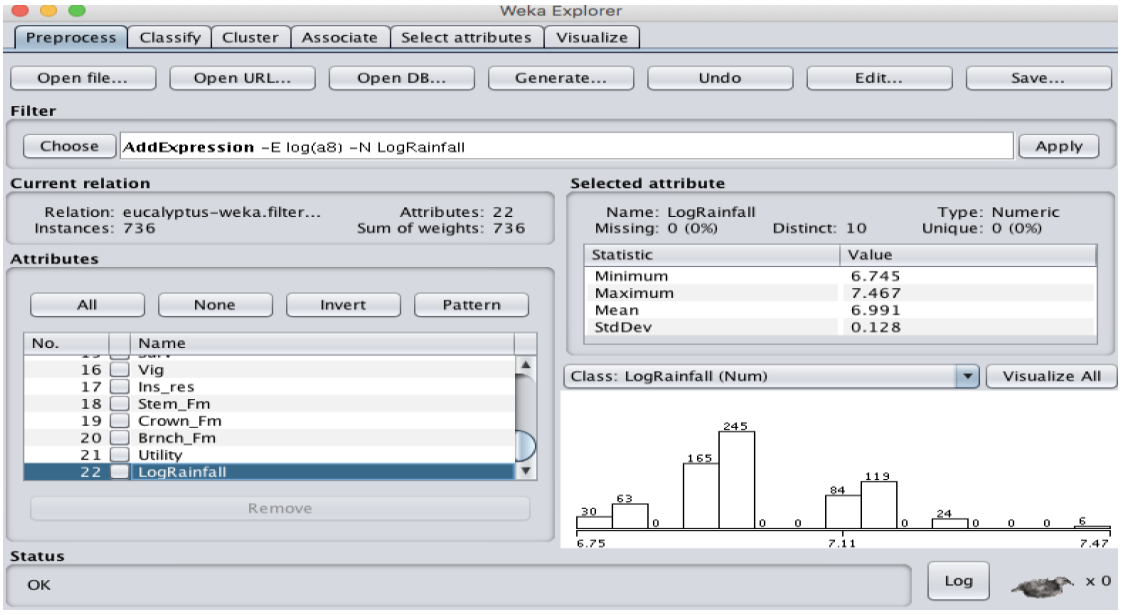


Fig 18: Distribution of LogRainfall attribute.

Now we can observe the distribution of LogRainfall and we can determine its mean, max, min and standard deviation. We see that more instances are concentrated in the range [6.75, 7.2]

iii. Replacing the Rainfall Values (Annual precipitation):

The fig 19 displays the distribution of Rainfall before any replacement. We observe that the min is 850 and the max is 1750 and we observe that there are just 24 values around 1400. So Now we perform the operation such that any values above 1500mm are replaced with 1400mm.

This can be achieved by using NumericCleaner Filter from Weka → filters →Unsupervised →attribute →NumericCleaner. Now we obtain a window as shown in fig 20 where we enter the attribute indices as 8 (No.of Rainfall attribute) and set the maxThreshold value to 1500 and the maxDefault to 1400. When we apply the filter it automatically updates the values above maxThreshold with the values of maxDefault.

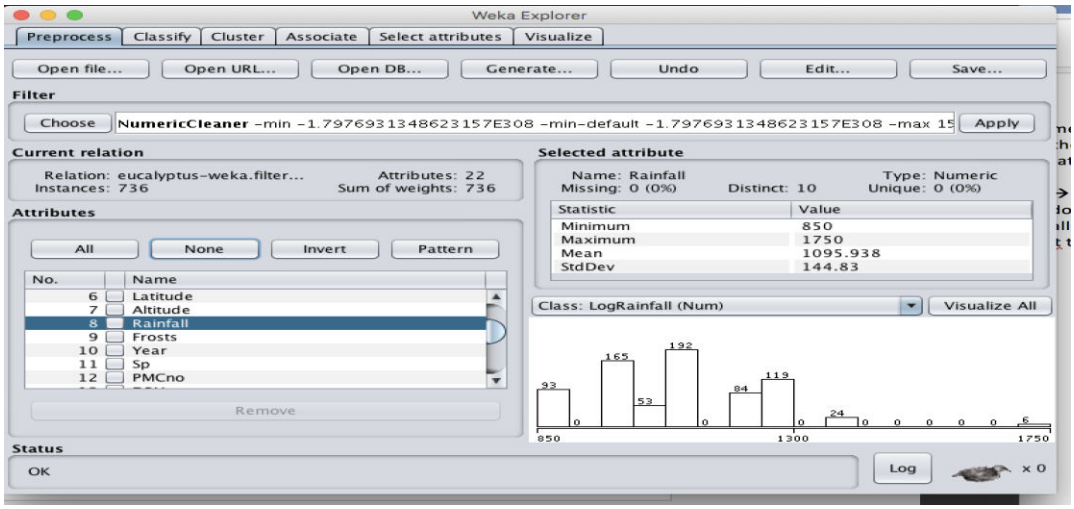


Fig 19: Rainfall attribute distribution before cleaning

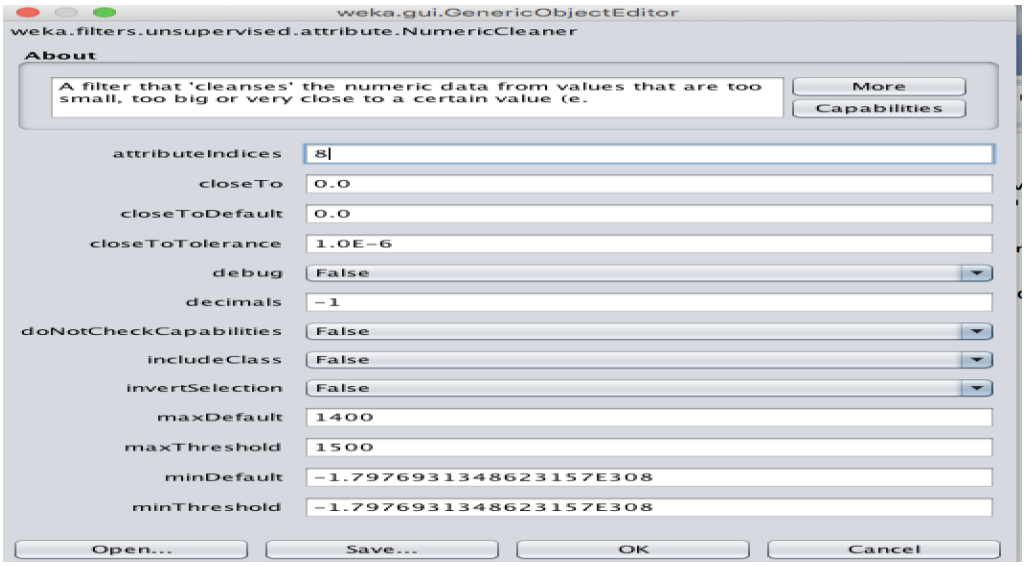


Fig 20: The NumericCleaner filter Window.

The fig 21 represents the distribution of Rainfall attribute after replacing values. We can notice that the maximum of the attribute is now 1400 where as previously it was 1750. We can also notice a corresponding decrease in mean and standard deviation. Now we also notice that the number of instances around 1400 has increased from 24 to 30 because of the replacement.

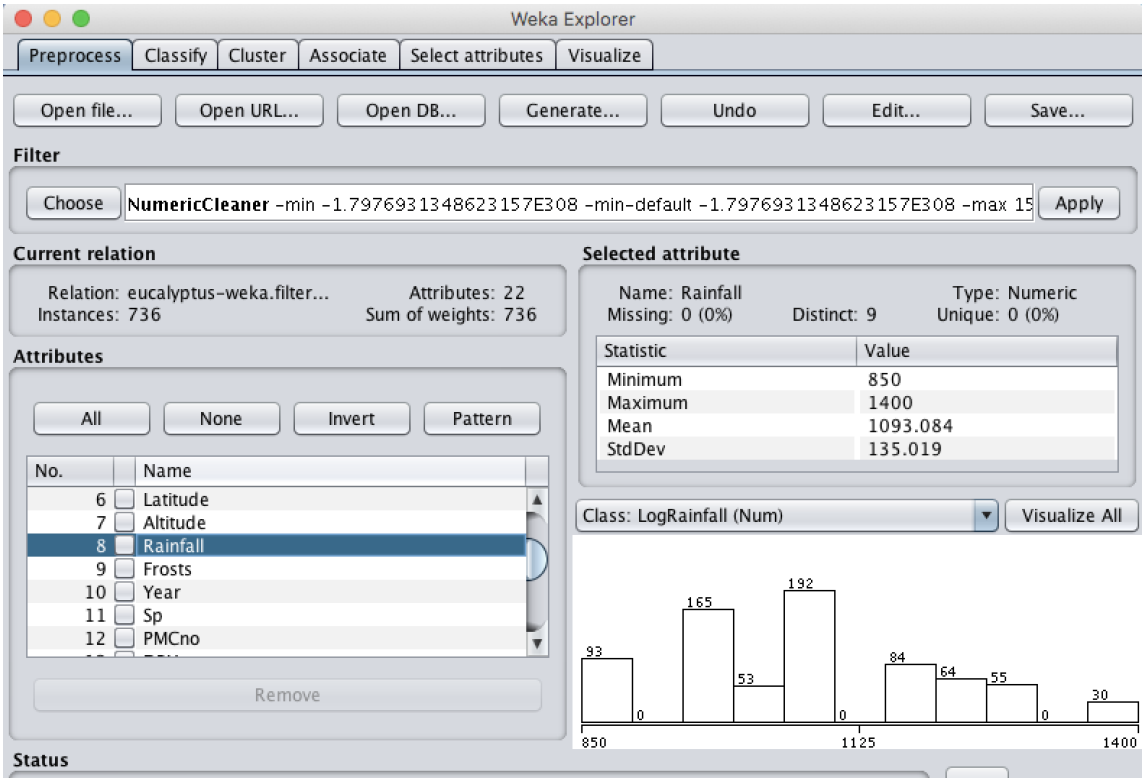


Fig 21: Rainfall attribute distribution after cleaning

iv. We can notice from tables 02 and 03 the attributes with missing values. We can observe that PMCno, DBH, Ht, Surv, Vig, Ins_res, Stem_Fm, Crown_Fm, Branch_Fm has missing values and the missing values can be filled either by using ReplaceMissingValues or ReplaceMissingValuesWithUserConstant filters. Both of them can be chosen by Weka → filters → Unsupervised → attribute → ReplaceMissingValues or ReplaceMissingValuesWithUserConstant.

Let us use the ReplaceMissingValues filter which replaces all the missing values of numeric and nominal attributes in a dataset with means and mode of the data. The displayed window is show in fig 22.

However, this filter doesn't take a list of indices instead it performs on all the attributes of a dataset. On the other hand, the ReplaceMissingValuesWithUserConstant takes just a single attribute index and replaces all the missing values in it with the user provided constant.

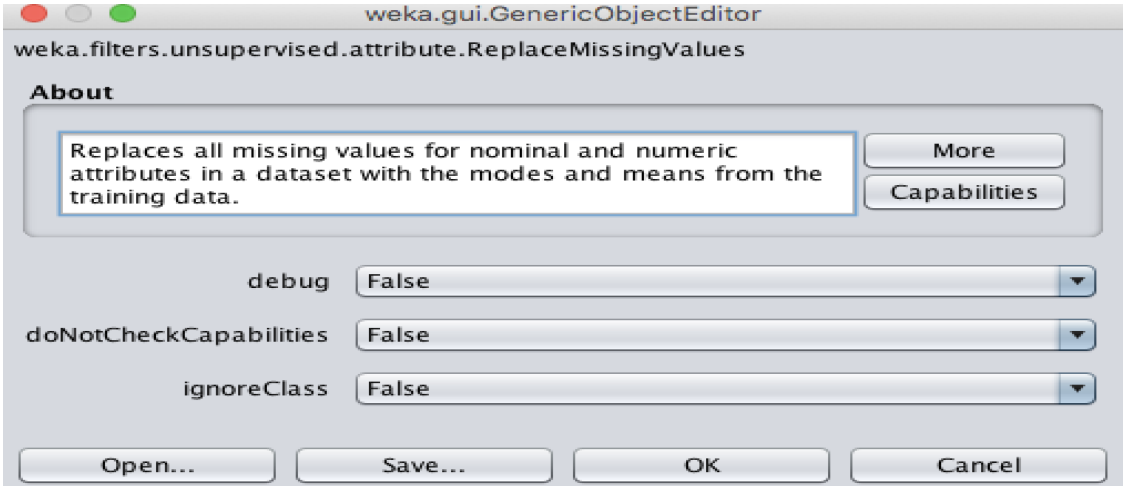


Fig 22: Replace Missing Values.

Now after replacing we can observe the change in the number of missing values for Crown_Fm attribute (Previously it was 69 now it is 0) as in fig 23.

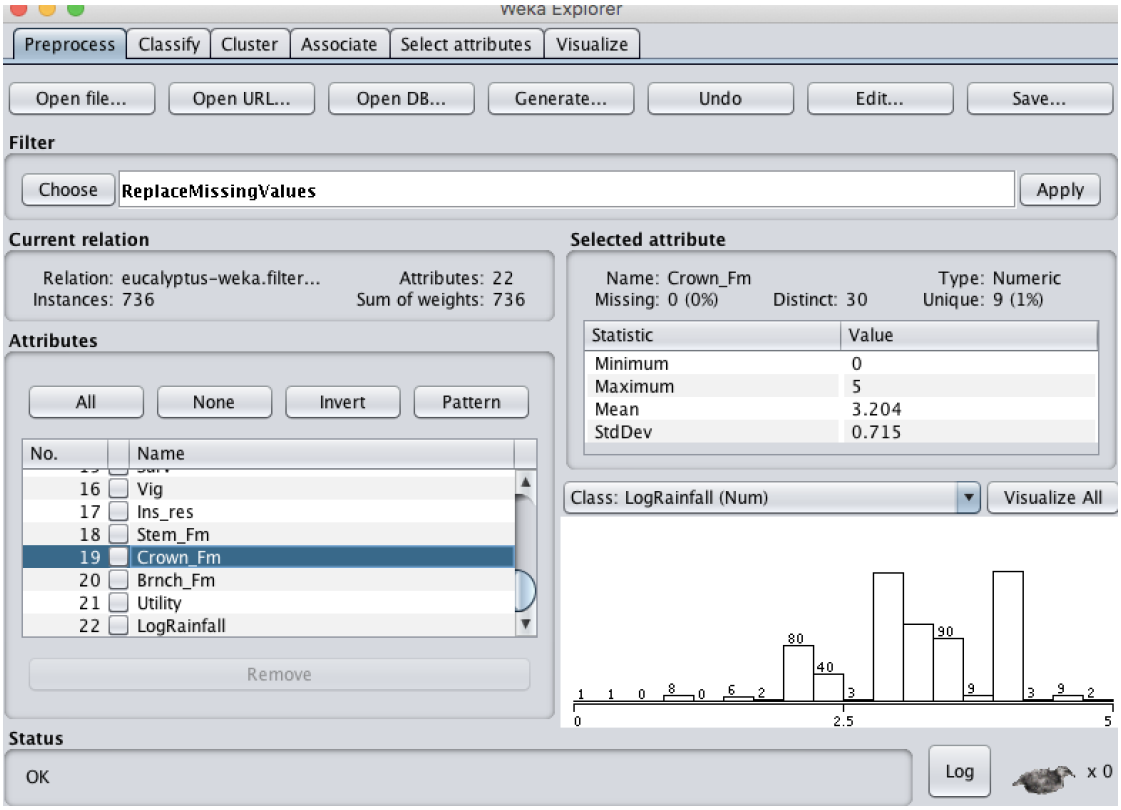


Fig 23: After Replacing Missing values in the data.

Now we can undo this action by clicking on the undo button in Weka tool.

v. Removing Instances with Missing Values:

We can remove all instances/records which contain missing values by using RemoveWithValues filter which is present in Weka → filters → Unsupervised → Instance → RemoveWithValues.

Now a window appears and enter the attribute index for which you want the missing values to be removed in the attributeIndex field. Also change the matchMissingValues value to true which is responsible for removing the missing values as shown in fig 24. Now click save and apply. Perform this operation for every attribute which has missing values.

One other approach is to select Multi-filter from the filter area and add the same matchMissingValues filter for every attribute to be considered and apply it.

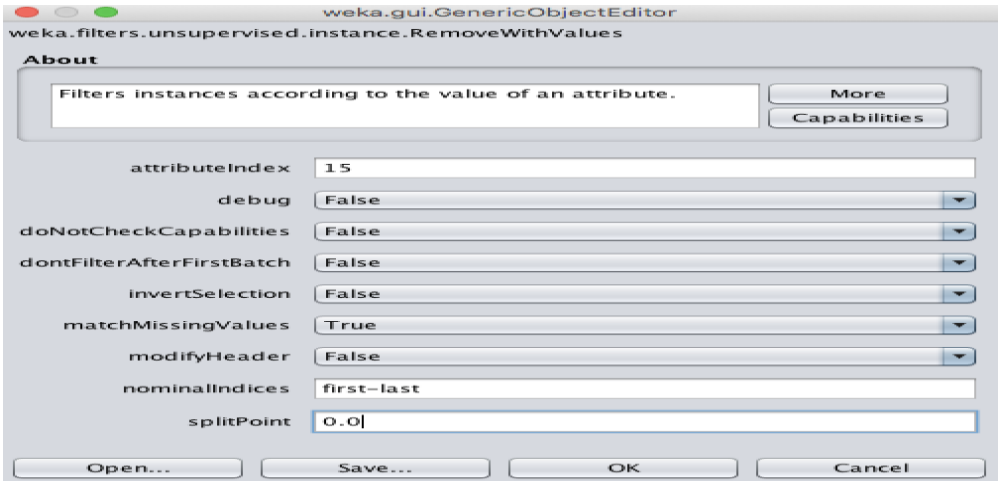


Fig 24: Remove with Values filter

The result consists of just 641 instances after removing all the instances containing at least 1 missing value which is shown in fig 25.

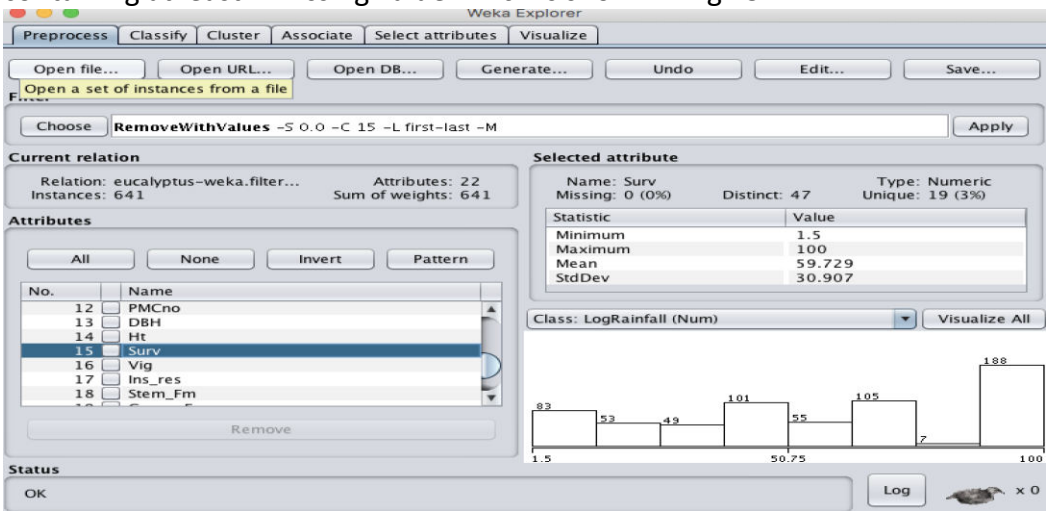


Fig 25: Instances after removing all Missing values

vi. We can save the obtained file by using the Save button in Weka tool.
Now, if we view it in a text editor we notice that the number of rows have been reduced and the data in the new file consists of no missing values. We also notice 2 additional attributes (ID, LogRainfall) added to the data. All the filters that have been done until now are stored at the top of the file as @relation.

There are many unsupervised filters for adding, removing, modifying, transforming, renaming, converting attributes or instances. These filters are useful in the pre-processing step of data mining where we need to remove noisy data or fill some noisy data or generate new attributes based on existing attributes or discretize numerical attributes or clean the data or sample the data.

a.) Weka → filters → Unsupervised → instance→RemoveMisClassified.

It is an instance filter which is used for removing the instances which are incorrectly classified. It is useful for removing outliers in the data. It can take several types of attributes to work on. We can specify the classifier of our choice to misclassify the instances based on a specific threshold. We can also specify the number of iterations to perform for misclassification. We also need to specify the index of the attribute which is to be considered as class. The filter also takes the number of folds to be considered for cross-validation.

b.) Weka → filters → unsupervised → attribute → Normalize

It is an attribute filter which is used for normalizing the values of an attribute. Normalization is an important pre-processing technique as it scales the value of an attribute in to a specific range of our choice. This Weka filter normalizes all the numeric attributes in a given dataset. By setting the Scale and Translation parameters we can adjust the range of the normalized values.
The resulting values of normalization are by default in [0,1] range.

The Normalize filter can be used on Numeric attributes, relational attributes, String attributes, Missing values, Binary attributes. The minimum number of instances required is 0. However, we can also include the class attribute in the normalization process by selecting the ignoreClass value to false.

4. Attribute Type Conversion:

This section performs conversion of attributes from Numeric to Nominal and from Nominal to Binary.

i. Numeric to Nominal:

All the Numeric attribute values in the data can be converted in to Nominal values by either equal-width binning or by equal-frequency binning. This method is available in Weka → filter → Unsupervised → attribute → Discretize.

In the Window that appears select the attribute-indices to 'first-last' as we want all the Numeric attributes within that range to be converted to Nominal attributes. Specify the number of bins to be 5. We can either choose to use equal frequency binning or leave it. We can also choose to display the bin numbers by selected true for useBinNumbers as shown in fig 26.
If we choose equal frequency binning then we also need to include the desired weight of instances per interval in appropriate field.

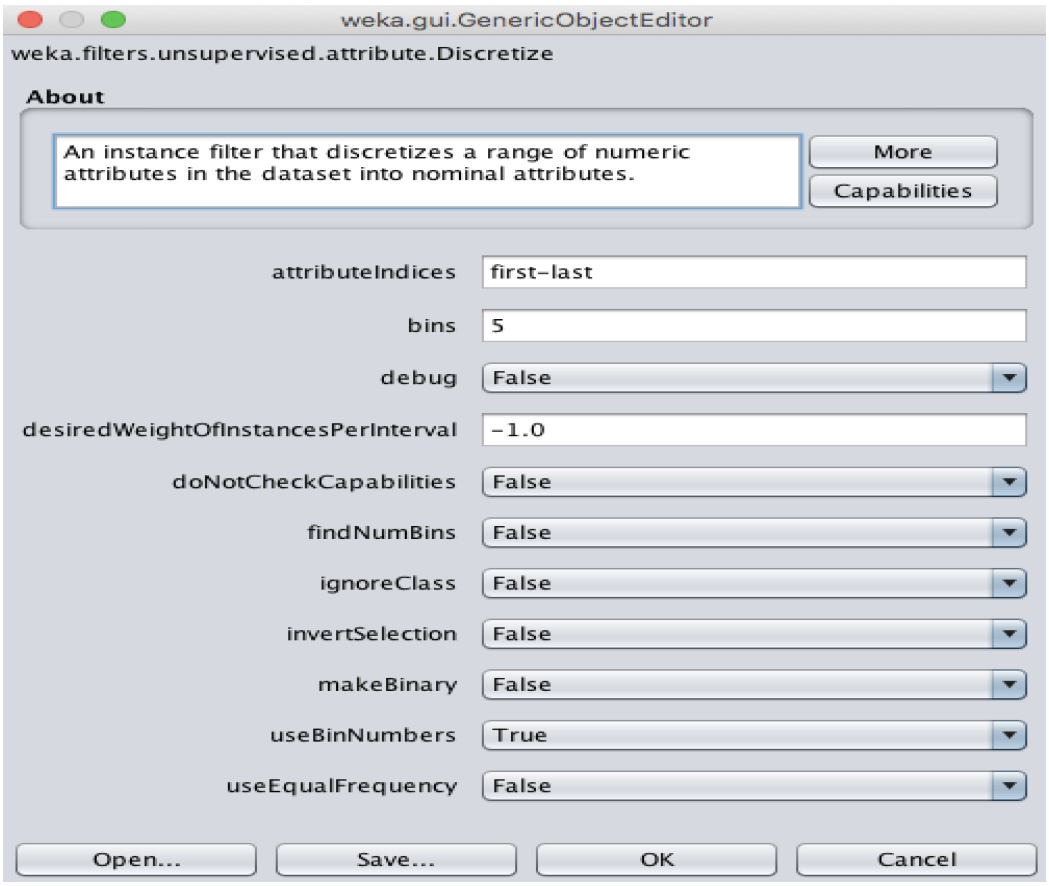


Fig 26: Discretize the numeric attributes to nominal attributes

Now save your parameters and click apply. We can now observe that all the numeric attributes are changed to nominal attributes which each consisting of 5 bins of equal size as shown in fig 27. Now save the obtained results as 'Eucalyptus Nominal.arff'

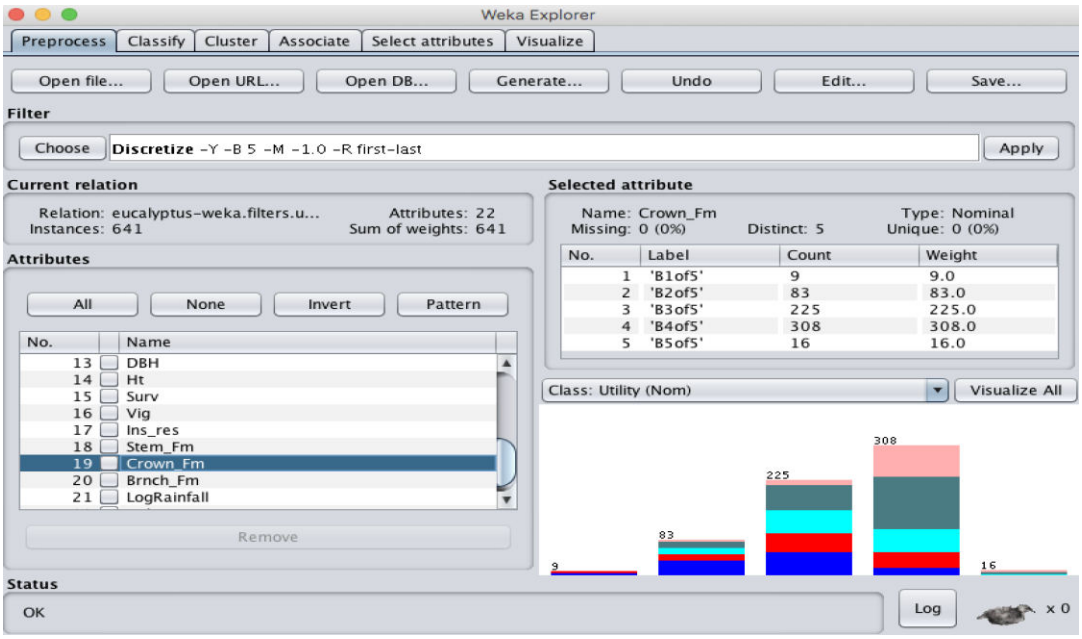


Fig 27: Numeric to Nominal conversion result

Now to make sure that utility is the class variable we need to move it to last, as last variable in a dataset is usually considered as Class Variable. To obtain this we use the reorder filter in weka. We provide the indices of attributes (except class attribute) along with class attribute to the filter in an order and it outputs the reordered dataset.

ii. Nominal to Binary:

To convert all the Nominal attributes in to Binary we can use the Nominal to Binary filter which is available in Weka → filters → Unsupervised → attribute →NominalToBinary.

In the window that appears we select the value for transformAllValues to be true just to make sure that every Nominal attribute (except the class attribute) is made binary irrespective of whether it contains more than 2 class labels as shown in fig.28. Now save the filter and apply it on the data set.

We can now see that every Nominal attribute is converted to binary attribute with values of either 0 or 1. A total of 158 attributes are present after applying the filter including the class variable.

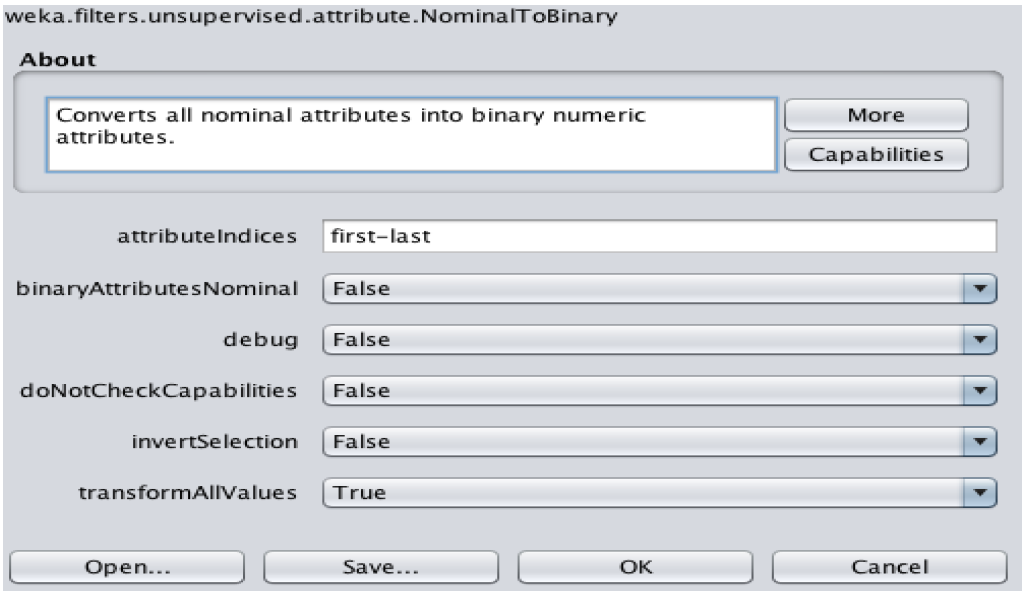


Fig 28: Nominal to Binary Filter.

All attributes except the class attribute are converted to binary and is shown in fig 29. It is now saved as 'Eucalyptus Binary.arff'

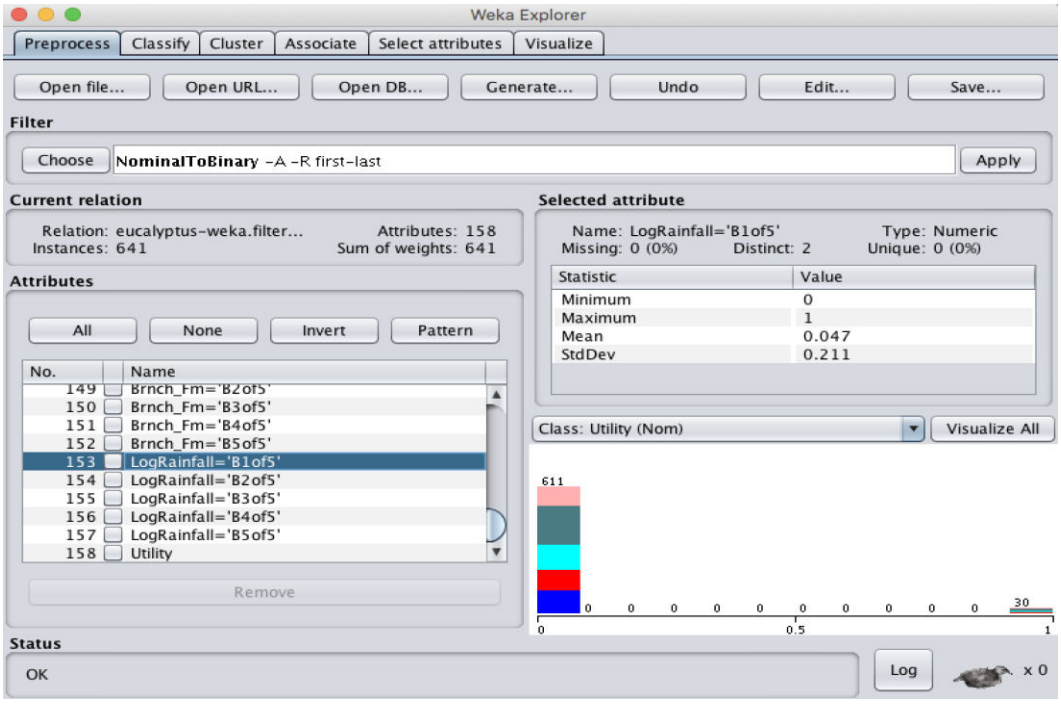


Fig 29: Nominal to Binary Result

CONTRIBUTIONS:

All the tasks have been done with efforts from both team mates. Zetao helped in capturing the screen shots while Krishna helped in writing down the report.

For task 3.6, Zetao looked at the filters for instances in Unsupervised learning while Krishna looked at the filters for attributes. We summarized these filters by discussing among ourselves.

Venkata Krishna Mohan Sunkara	45%
Zetao Zhao	55%

ACKNOWLEDGMENTS:

1. Weka Documentation.