Name: Sourabh Deshkulkarni
ID- ssd9930

Topics in Data Management assignment -4

Objective : To predict 'Rating' of an web-service API using 'Random Tree' classification technique.

Tools used: Weka, Java, MS Excel and Text editors.

Data Cleaning and Preparation performed:

1. First step was to convert the api.txt file into csv format to make it compatible with MS Excel and Weka tool.
2. There were several blank columns were present, in initial skimming through, removed blank columns -Author removed, version.
3. There were several columns which had over 80-90% data as blank and they added no significance and value to Ranking classification, hence I removed below columns
   i) Wsdl
   ii) Apigroup
   iii) Example
   iv) Blog
   v) Commercial
   vi) Readonly
   vii) Forum
   viii) Support
   ix) Managedby
   x) Limits (This is similar to fees, and fees had significant number of instances hence removed this)
   xi) Vendorapikit
   xii) Communityapikit
   xiii) noncommercial
   xiv) dataLicensing
   xv) Company
4. Serviceendpoint had 40% instances as blank, although it has no significance to Rating prediction, I decided to keep it and made all blank instances as 'Unknown'.

5. dataFormat contained 20% missing values but it is a very significant attribute for Ranking prediction, I found out that 'XML' was pre-dominant value value in dataFormat, hence I assigned it to to the missing values for dataFormat.

6. ClientInstall contained 60% data missing and it contained 39% rest values as ‘No’, hence I decided to annotate 60% missing values as ‘Unknown’.

7. Performed extensive data analysis of the ‘Authetication’ and did binning on Authentication column, and reduced the categories into ‘Required’, ‘None’ and ‘Unknown’.

8. Type column only had ‘1’ as constant value, this adds no value to building model, hence removed it.

9. ‘Protocol’ contained 20% missing values, but it contained 50% instances as ‘REST’ which is pre-dominant, hence I termed the 20% missing values as ‘REST’

10. I wrote a Java program to reduce entropy of ‘DateModified’ and ‘Updated’ column which contained date along with time. As only date is a sufficient information and time created more unique values, hence I trimmed the values to only keep date.

11. I also wrote a code to remove stop words from Description, to see if description adds any value to the prediction. I used publically available stopwords file from web for the same.

12. Fees is one of the most important attribute for Ranking prediction, hence after data analysis, I came up with ‘free’, ‘paid’ and ‘unknown’ as binned categories.

13. AccountReq is again important an attribute for ranking prediction, hence I categorized them into Yes, No and Unknown by handling missing values.

14. Created a separate category as ‘Unknown for ‘terms’ attribute and handled it.

15. Ssl is a significant attribute too, hence I categorized them into Yes, No and Unknown by handling missing values.
16. For ‘Provider’, I categorized missing values as ‘Unknown’.
17. Finally, for class variable ‘Rating’ I categorized it into ‘One’, ‘Two’, ‘Three’, ‘Four’ and ‘Five’. For this I converted values like this example for e.g. values above 1 and less than or equal to 2 will be categorized as ‘Two’.

With above data cleaning and preparation steps, below attributes remain completely handled.

1. **ID**
2. **Title**
3. **Summary**
4. **Rating**
5. **Name**

6. **Label**
7. **Description**
8. **SampleURL**
9. **DateModified**
10. **CommentsURL**
11. **Tags**
12. **Category**
13. **Protocol**
14. **ServiceEndPoint**
15. **DataFormats**
16. **ClientInstall**
17. **Authentication**
18. **SSL**
19. **AccountReq**
20. **Provider**
21. **Fees**
22. **Terms**
23. **Updated.**

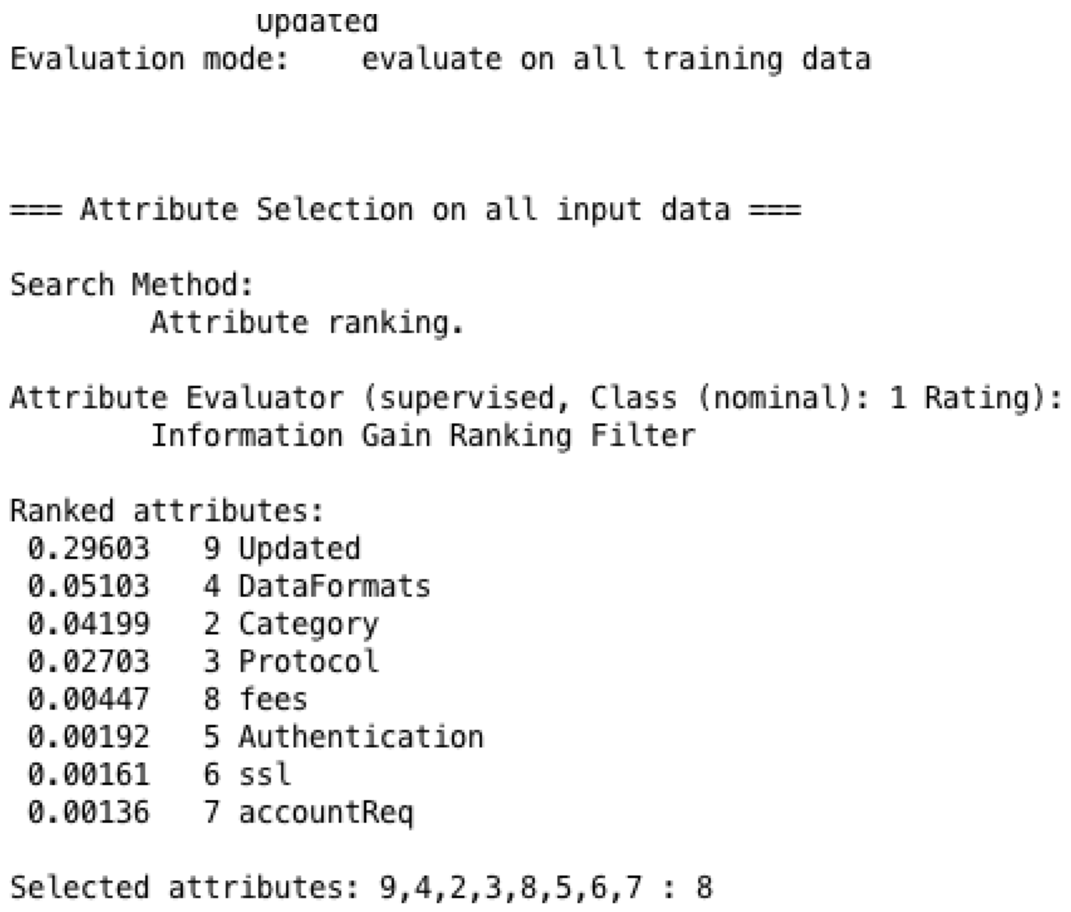
Out of these attributes, after careful observation I came to conclusion of using below 8 attributes for 'Rating' prediction.

1. **Category** – This attribute defines what is the type of an API. For a particular category which is more useful to the user, could be used more and receive more rating.
2. **Protocol** – This attribute defines what protocol is used by the api, rating depends on this attribute as for e.g. some people may find REST service easy to use than SOAP and can rate it more.
3. **DataFormats**- This attribute also affects the rating. For e.g. due to ease of use some may find JSON format better and rate service based on that.
4. **Authentication**- If an API doesn't require API key or further authentication to use the API, it makes API use easier and an API may get more rating.
5. **SSL**- If an API provides SSL security then it will be rated higher.
6. **AccountReq**- If an API doesn't require sign up and account creation then it saves user's time and API will be rated more.
7. **Fees** – This is foremost important. If an API is free and provides necessary functionality it will be liked by all and receive good rating
8. **Updated** – If the API is updated regularly, then we can assume that it is improved continuously by the developers can receive good rating.

Experiment and results

Finding Information Gain of the attributes:

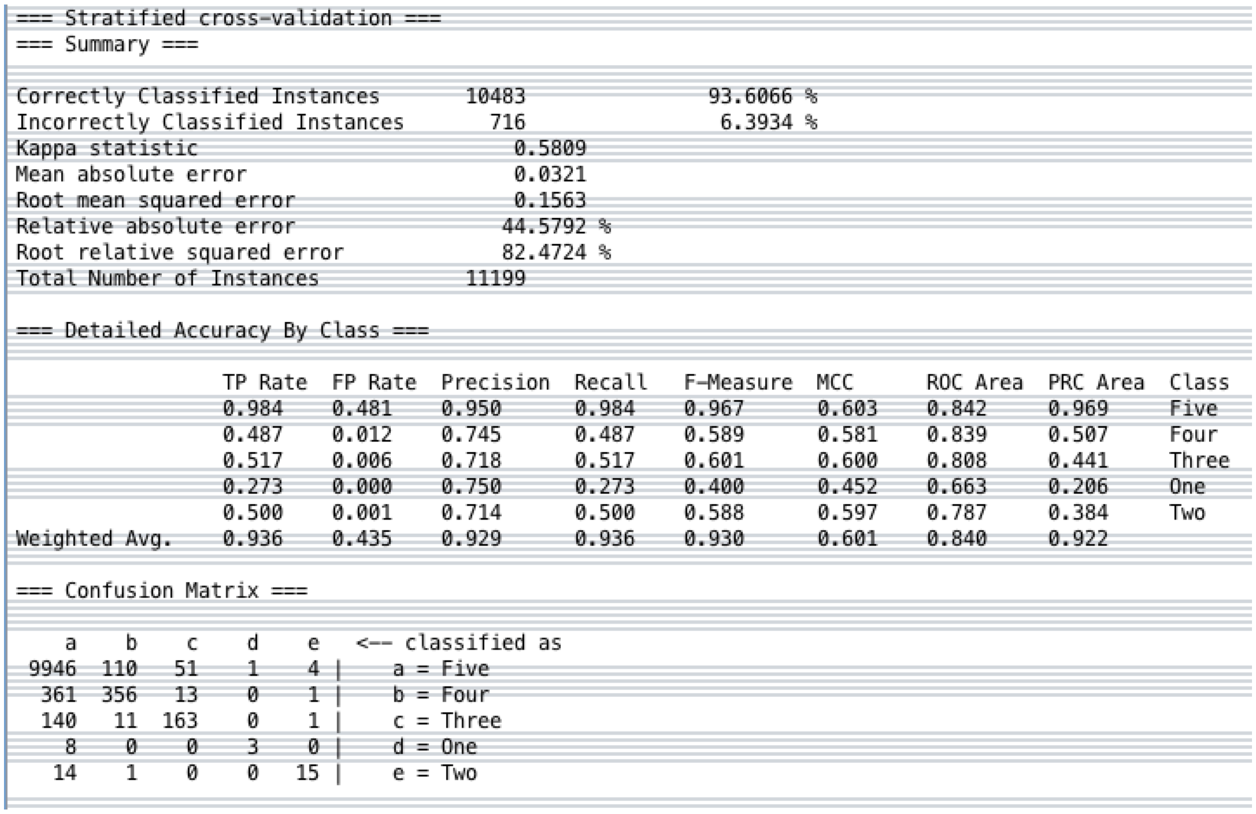
1. Information gain function shows the information gain ranking of the attributes, starting from higher to lower.
2. The results of info gain filter confirmed the selection of attributes and root node to be Updated.



```
               updated
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 Rating):
        Information Gain Ranking Filter

Ranked attributes:
 0.29603   9 Updated
 0.05103   4 DataFormats
 0.04199   2 Category
 0.02703   3 Protocol
 0.00447   8 fees
 0.00192   5 Authentication
 0.00161   6 ssl
 0.00136   7 accountReq

Selected attributes: 9,4,2,3,8,5,6,7 : 8
```

Model Building

1. I used Weka and **Random Tree** algorithm with 10-fold cross validation to predict the rating and received over **93%** accuracy for my model. Random tree selects K different attributes at each level and it requires no pruning.
2. We can observe that most of the class variables were distributed in class ‘Five’. So the prediction would have been biased to class ‘Five’. To overcome this problem, I used stratified sampling filter ‘Resample’ filter from Weka, which takes equal number of instances from each class variable and uses them to build the model. Hence the results we see below are unbiased results.



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       10483               93.6066 %
Incorrectly Classified Instances       716                6.3934 %
Kappa statistic                          0.5809
Mean absolute error                      0.0321
Root mean squared error                  0.1563
Relative absolute error                 44.5792 %
Root relative squared error             82.4724 %
Total Number of Instances            11199

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.984    0.481    0.950      0.984    0.967      0.603    0.842     0.969     Five
                 0.487    0.012    0.745      0.487    0.589      0.581    0.839     0.507     Four
                 0.517    0.006    0.718      0.517    0.601      0.600    0.808     0.441     Three
                 0.273    0.000    0.750      0.273    0.400      0.452    0.663     0.206     One
                 0.500    0.001    0.714      0.500    0.588      0.597    0.787     0.384     Two
Weighted Avg.    0.936    0.435    0.929      0.936    0.930      0.601    0.840     0.922

=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 9946  110   51    1    4 |    a = Five
  361  356   13    0    1 |    b = Four
  140   11  163    0    1 |    c = Three
    8    0    0    3    0 |    d = One
   14    1    0    0   15 |    e = Two
```

3. Pleas find **WekaResultRandomTree.txt** fro complete decision tree and detailed **Weka results.**
4. **From confusion matrix, we can observe that we got highest accuracy for rating 5 but at the same time accuracy for group 'Three', 'Two' and 'Four' is around or more than 50%. This could not have been possible without the help of Stratified sampling.**

Conclusion

I believe, with appropriate data cleaning with binning and missing value handling techniques, we can achieve good accuracy and model.