Report for Week 8

I. Summary

1. I trained a NB Classifier on the **Twitter set** from Ken Benoit.
2. I built the 3 classes of features discussed.
3. I trained 3 XGB classifiers on **Reddit dataset,** for the 3 classes of features.

II. Actions performed

1. NB Classifier of Twitter Data

I followed exactly the methodology presented by Ken Benoit, in his paper. Please see the Final Report PDF, section 4.3, subsection "Data Preparation" for the steps involved in processing the data (exactly the steps followed by KB). I end up by having 11277 aggregated user texts.

I build the leave index, for each user, as No. Of Leave Hashtags – No Of Remain Hashtags and sort the users according to this index. I take the first 10% as Leave Labeled Training data and the last 10% of the users as Remain Labeled Training Data. Together, they form a total of 2200 documents, used for training the classifier.

I perform standard preprocessing (remove stopwords, @mentions, #hashtags), I perform stemming and keep only words that have at least 5 times. Important! I remove hashtags, so that the trained classifier will perform equally well on Reddit dataset.

I build a DTM, having 2200 documents and ASDAS features. I split this dataset in 80% training and 20% testing set. I train the Naive Bayes Classifier on the training set and test it on the new, never seen test set. **The accuracy obtained on the test set is about 93%** (very similar to Ken Bonoit accuracy reported in the paper).

This classifier will be used to generate ground truth labels on Reddit Dataset.

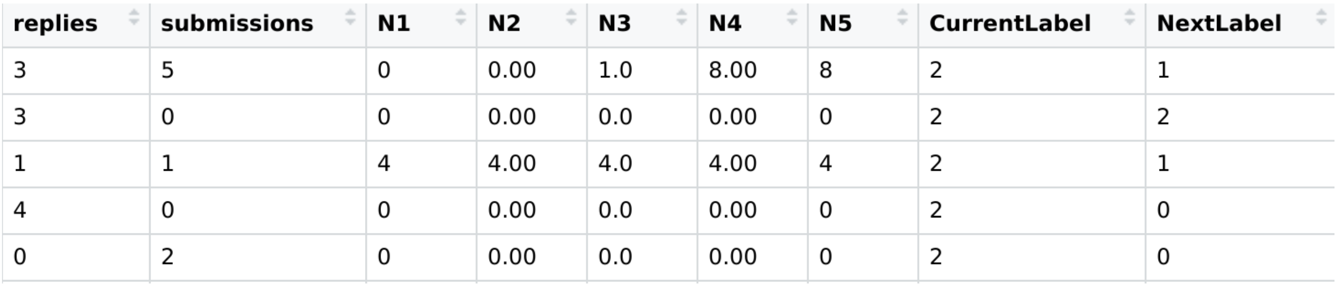
2. Reddit Dataset

I used the classifier trained on the Twitter Dataset on Reddit. To build the 3 classes of features, I consider the initial splitting in 15 timeframes. For every consequent timeframes, I consider the common authors. Thus, at every timeframe t, I have information about the current class of the author (position about the Brexit – **A**gainst, **B**rexit, **N**eutral) and his position in the following timeframe t+1. Usint this methodology, I collect all the common authors between every 2 consequent timeframes and their positions about brexit. Then, I build different features, taking into account the discussed situations: F1, F2, F3.

3. XGBoosting on Reddit Dataset

a. **F1 Features** -

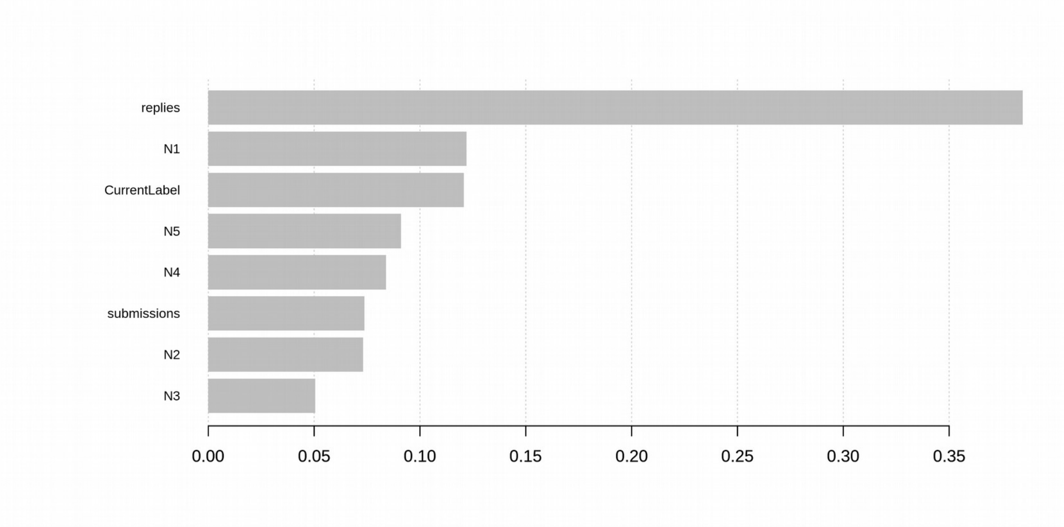
As we previously discussed, I built the first set of features comprising: the number of initial posts in period t, number of comments in period t, the 5 quantiles obtained from the number of comments / posts in period t. I changed a bit the specification that we discussed by turning the classes c1, c2, c3 … c9. I consider the current class as a feature (current position at period t: **A, B, N**) and I consider the next class (the position in the next time-frame t+1) as the unknown variable.

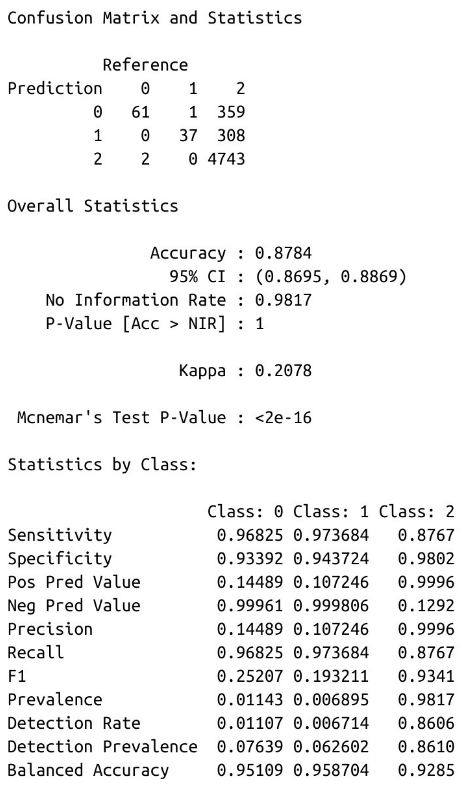


replies	submissions	N1	N2	N3	N4	N5	CurrentLabel	NextLabel
3	5	0	0.00	1.0	8.00	8	2	1
3	0	0	0.00	0.0	0.00	0	2	2
1	1	4	4.00	4.0	4.00	4	2	1
4	0	0	0.00	0.0	0.00	0	2	0
0	2	0	0.00	0.0	0.00	0	2	0

In previous figure, there is a snippet of the dataset using F1. I split the dataset in 2 parts, training respectively test set: ratio 75% - 25%. I use a XGBoost algorithm and 10-fold Cross Validation methodology. The dataset is very imbalanced (more neutral than A and B). Following a vignette online from the package, for imbalanced, multiclass classfication problems it is suitable to use: objective function multi-class softmax probability and the evaluation metric is multi- logloss. I have 10 fold validation, with 100 repteats.

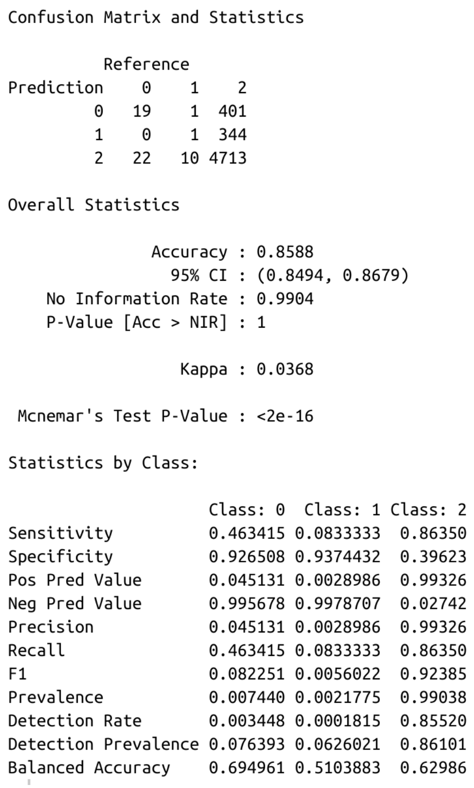
In the following figure, I plotted the features that influenced the most the learning process (Gini index)





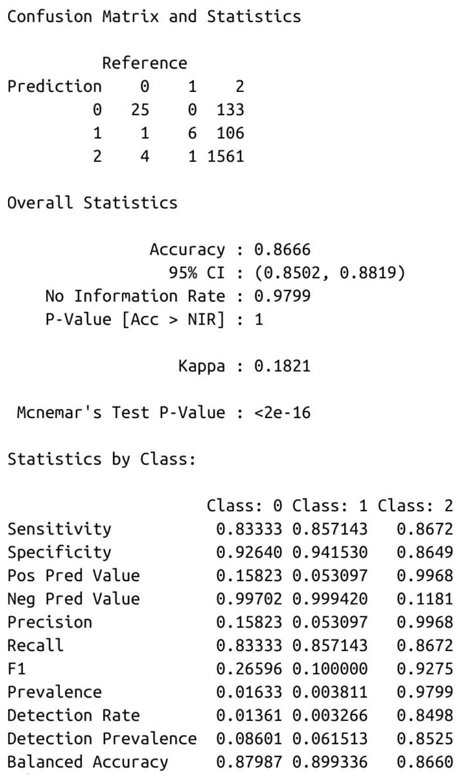
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   61    1  359
         1    0   37  308
         2    2    0 4743

Overall Statistics

               Accuracy : 0.8784
                 95% CI : (0.8695, 0.8869)
    No Information Rate : 0.9817
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2078

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.96825 0.973684   0.8767
Specificity           0.93392 0.943724   0.9802
Pos Pred Value        0.14489 0.107246   0.9996
Neg Pred Value        0.99961 0.999806   0.1292
Precision             0.14489 0.107246   0.9996
Recall                0.96825 0.973684   0.8767
F1                    0.25207 0.193211   0.9341
Prevalence            0.01143 0.006895   0.9817
Detection Rate        0.01107 0.006714   0.8606
Detection Prevalence  0.07639 0.062602   0.8610
Balanced Accuracy     0.95109 0.958704   0.9285
```

F1 – Training Statistics (after doing CV, I test the classifier on the test data)



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   19    1  401
         1    0    1  344
         2   22   10 4713

Overall Statistics

               Accuracy : 0.8588
                 95% CI : (0.8494, 0.8679)
    No Information Rate : 0.9904
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0368

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 0  Class: 1 Class: 2
Sensitivity          0.463415 0.0833333  0.86350
Specificity          0.926508 0.9374432  0.39623
Pos Pred Value       0.045131 0.0028986  0.99326
Neg Pred Value       0.995678 0.9978707  0.02742
Precision            0.045131 0.0028986  0.99326
Recall               0.463415 0.0833333  0.86350
F1                   0.082251 0.0056022  0.92385
Prevalence           0.007440 0.0021775  0.99038
Detection Rate       0.003448 0.0001815  0.85520
Detection Prevalence 0.076393 0.0626021  0.86101
Balanced Accuracy    0.694961 0.5103883  0.62986
```

F1 – CV – Statistics (the stats on the hold out fold, during CV process)

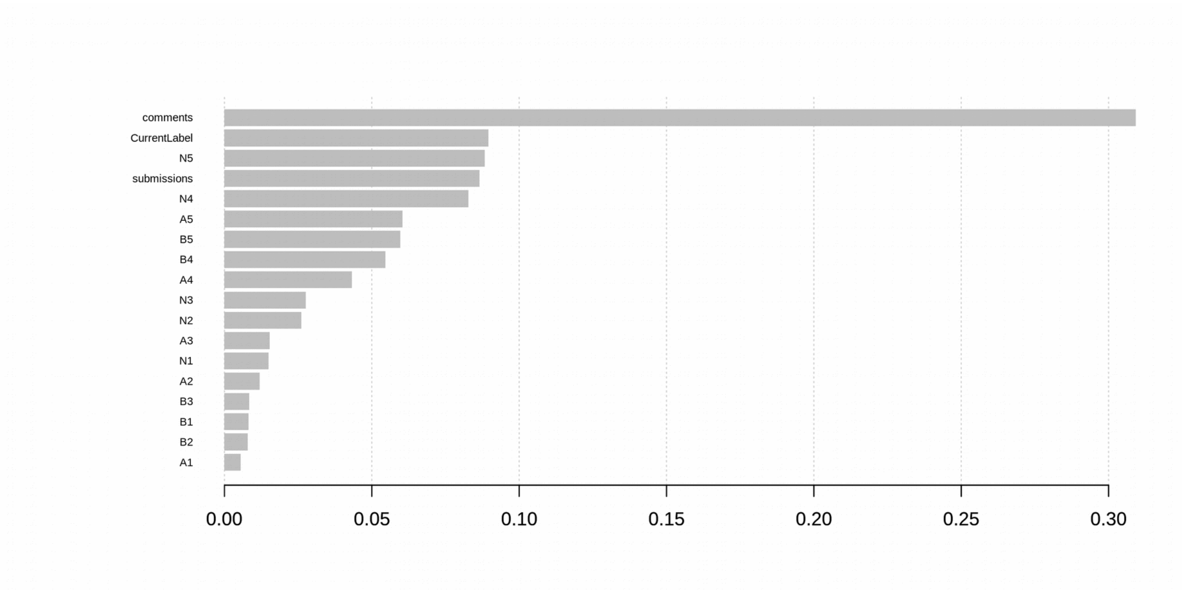


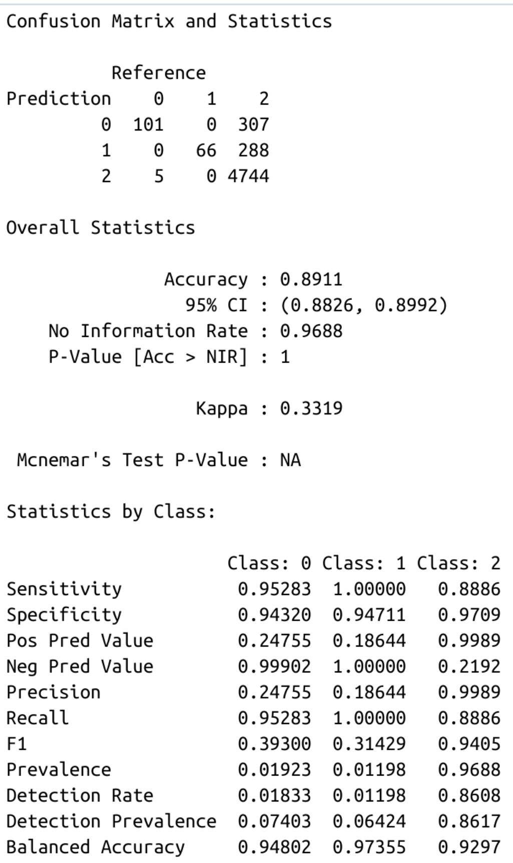
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   25    0  133
         1    1    6  106
         2    4    1 1561

Overall Statistics

               Accuracy : 0.8666
                 95% CI : (0.8502, 0.8819)
    No Information Rate : 0.9799
    P-Value [Acc > NIR] : 1

                  Kappa : 0.1821

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.83333 0.857143   0.8672
Specificity           0.92640 0.941530   0.8649
Pos Pred Value        0.15823 0.053097   0.9968
Neg Pred Value        0.99702 0.999420   0.1181
Precision             0.15823 0.053097   0.9968
Recall                0.83333 0.857143   0.8672
F1                    0.26596 0.100000   0.9275
Prevalence            0.01633 0.003811   0.9799
Detection Rate        0.01361 0.003266   0.8498
Detection Prevalence  0.08601 0.061513   0.8525
Balanced Accuracy     0.87987 0.899336   0.8660
```

F1 – Test – Statistics (stats on the test, unseen data)

b. F2 – Features

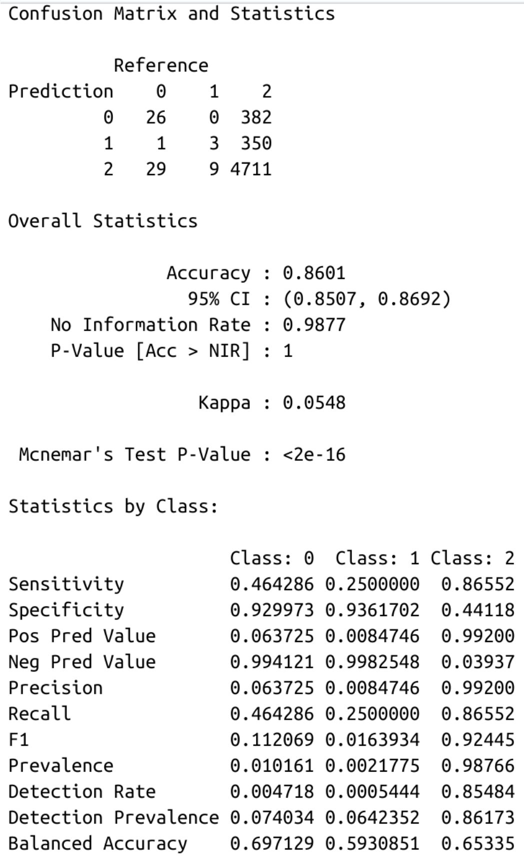
We have again the number of initial posts, comments and now we split the number of replies per post in 3 categories – From A, from B and from N. Basically, I take every post of a user (initial or intermediate comment) and I count for it how many replies (posts that have as parent the analyzed post) of each kind it has. I built the quantiles for Against, Neutral and Brexit.





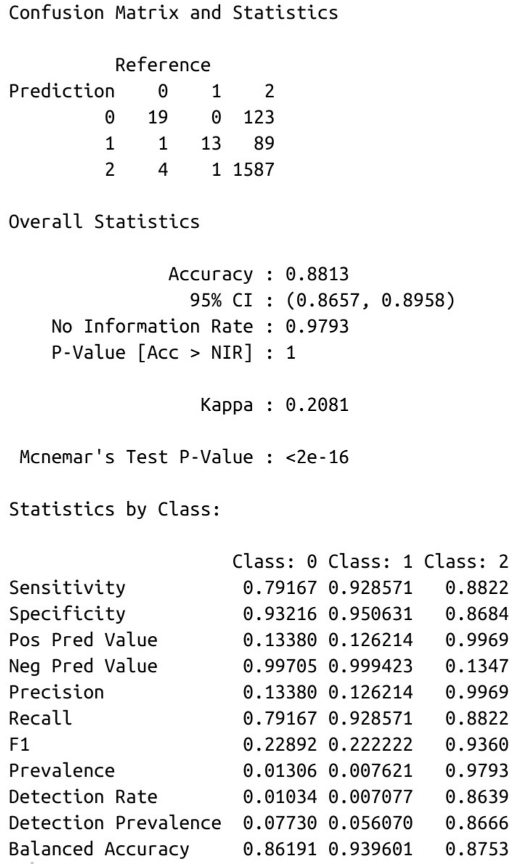
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0  101    0  307
         1    0   66  288
         2    5    0 4744

Overall Statistics

               Accuracy : 0.8911
                 95% CI : (0.8826, 0.8992)
    No Information Rate : 0.9688
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3319

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.95283  1.00000   0.8886
Specificity           0.94320  0.94711   0.9709
Pos Pred Value        0.24755  0.18644   0.9989
Neg Pred Value        0.99902  1.00000   0.2192
Precision             0.24755  0.18644   0.9989
Recall                0.95283  1.00000   0.8886
F1                    0.39300  0.31429   0.9405
Prevalence            0.01923  0.01198   0.9688
Detection Rate        0.01833  0.01198   0.8608
Detection Prevalence  0.07403  0.06424   0.8617
Balanced Accuracy     0.94802  0.97355   0.9297
```

F2 – Training Stats



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   26    0  382
         1    1    3  350
         2   29    9 4711

Overall Statistics

               Accuracy : 0.8601
                 95% CI : (0.8507, 0.8692)
    No Information Rate : 0.9877
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0548

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 0  Class: 1 Class: 2
Sensitivity          0.464286 0.2500000  0.86552
Specificity          0.929973 0.9361702  0.44118
Pos Pred Value       0.063725 0.0084746  0.99200
Neg Pred Value       0.994121 0.9982548  0.03937
Precision            0.063725 0.0084746  0.99200
Recall               0.464286 0.2500000  0.86552
F1                   0.112069 0.0163934  0.92445
Prevalence           0.010161 0.0021775  0.98766
Detection Rate       0.004718 0.0005444  0.85484
Detection Prevalence 0.074034 0.0642352  0.86173
Balanced Accuracy    0.697129 0.5930851  0.65335
```

F2 – CV Stats

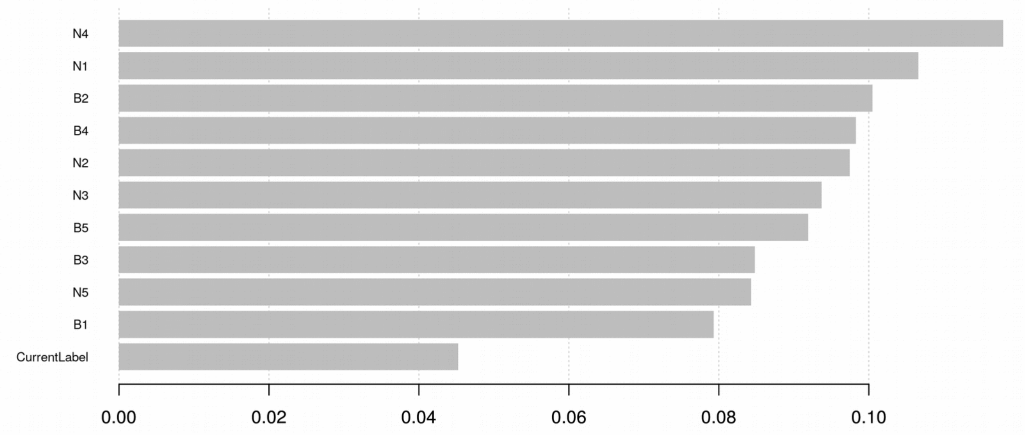


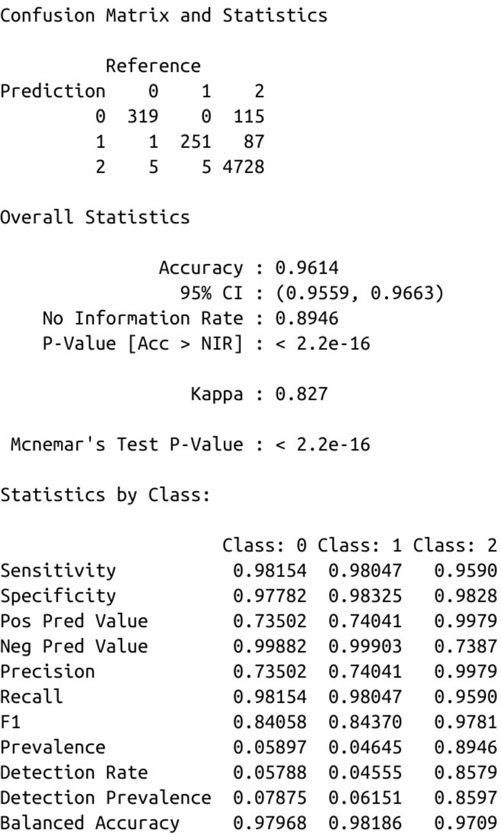
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   19    0  123
         1    1   13   89
         2    4    1 1587

Overall Statistics

               Accuracy : 0.8813
                 95% CI : (0.8657, 0.8958)
    No Information Rate : 0.9793
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2081

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.79167 0.928571   0.8822
Specificity           0.93216 0.950631   0.8684
Pos Pred Value        0.13380 0.126214   0.9969
Neg Pred Value        0.99705 0.999423   0.1347
Precision             0.13380 0.126214   0.9969
Recall                0.79167 0.928571   0.8822
F1                    0.22892 0.222222   0.9360
Prevalence            0.01306 0.007621   0.9793
Detection Rate        0.01034 0.007077   0.8639
Detection Prevalence  0.07730 0.056070   0.8666
Balanced Accuracy     0.86191 0.939601   0.8753
```

F2 – Test Stats

c. F3 – Features

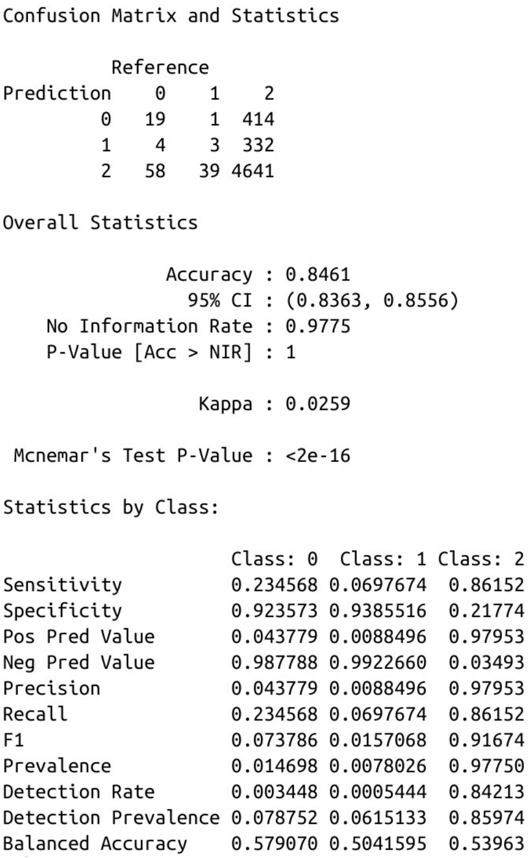
Now, we have no more number of posts and comments and we remove also the number of replies. Now, for each common author, between two conseqtive periods, I take all the diffusions that he started (he wrote the first message) and I compute the percentage of posts pro and against brexit. I built the quantiles for Brexit percentages and for Neutral.





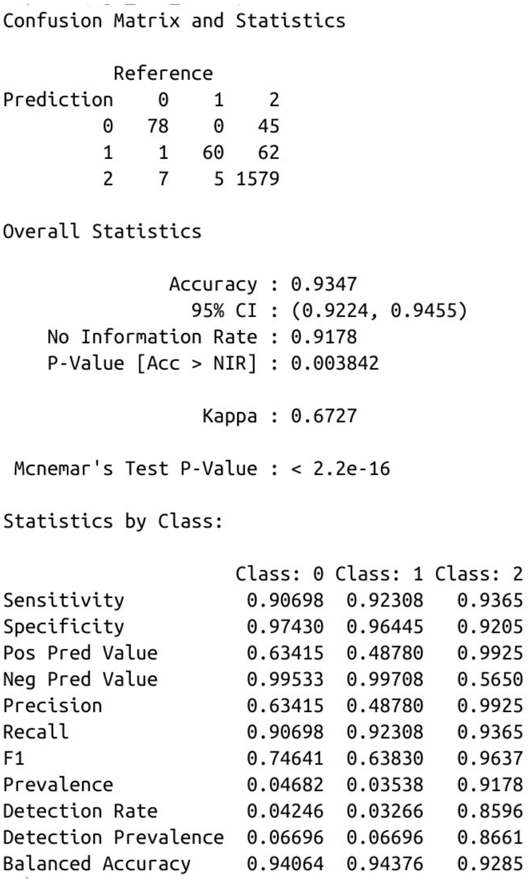
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0  319    0  115
         1    1  251   87
         2    5    5 4728

Overall Statistics

               Accuracy : 0.9614
                 95% CI : (0.9559, 0.9663)
    No Information Rate : 0.8946
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.827

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.98154  0.98047   0.9590
Specificity           0.97782  0.98325   0.9828
Pos Pred Value        0.73502  0.74041   0.9979
Neg Pred Value        0.99882  0.99903   0.7387
Precision             0.73502  0.74041   0.9979
Recall                0.98154  0.98047   0.9590
F1                    0.84058  0.84370   0.9781
Prevalence            0.05897  0.04645   0.8946
Detection Rate        0.05788  0.04555   0.8579
Detection Prevalence  0.07875  0.06151   0.8597
Balanced Accuracy     0.97968  0.98186   0.9709
```

F3 – Training Stat



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   19    1  414
         1    4    3  332
         2   58   39 4641

Overall Statistics

               Accuracy : 0.8461
                 95% CI : (0.8363, 0.8556)
    No Information Rate : 0.9775
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0259

 Mcnemar's Test P-Value : <2e-16

Statistics by Class:

                     Class: 0  Class: 1 Class: 2
Sensitivity          0.234568 0.0697674  0.86152
Specificity          0.923573 0.9385516  0.21774
Pos Pred Value       0.043779 0.0088496  0.97953
Neg Pred Value       0.987788 0.9922660  0.03493
Precision            0.043779 0.0088496  0.97953
Recall               0.234568 0.0697674  0.86152
F1                   0.073786 0.0157068  0.91674
Prevalence           0.014698 0.0078026  0.97750
Detection Rate       0.003448 0.0005444  0.84213
Detection Prevalence 0.078752 0.0615133  0.85974
Balanced Accuracy    0.579070 0.5041595  0.53963
```

F3 – CV Stat



```
Confusion Matrix and Statistics

          Reference
Prediction    0    1    2
         0   78    0   45
         1    1   60   62
         2    7    5 1579

Overall Statistics

               Accuracy : 0.9347
                 95% CI : (0.9224, 0.9455)
    No Information Rate : 0.9178
    P-Value [Acc > NIR] : 0.003842

                  Kappa : 0.6727

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 0 Class: 1 Class: 2
Sensitivity           0.90698  0.92308   0.9365
Specificity           0.97430  0.96445   0.9205
Pos Pred Value        0.63415  0.48780   0.9925
Neg Pred Value        0.99533  0.99708   0.5650
Precision             0.63415  0.48780   0.9925
Recall                0.90698  0.92308   0.9365
F1                    0.74641  0.63830   0.9637
Prevalence            0.04682  0.03538   0.9178
Detection Rate        0.04246  0.03266   0.8596
Detection Prevalence  0.06696  0.06696   0.8661
Balanced Accuracy     0.94064  0.94376   0.9285
```

F3 – Testing Stat

In conclusion, F3 – Features have the best F1 score on the testing set (data is imbalanced so we don’t count on Accuracy).