Applying Improved Random Forest Explainability (RFEX 2.0) on synthetic data

Sabiha Barlaskar, Dragutin Petkovic
SFSU CS Department
11/27/2018

1. Introduction

Our Random Forest (RF) [2] Explainability enhancement method (RFEX) [1] has already been applied to biomedical data [1] and to SW Engineering Teamwork Prediction and Assessment data (SETAP) [3,4] data where we found that it offers insights (e.g. explainability) into how RF performs its classification, as well as important factors in RF classification which may help in understanding of the underlying problem [3]. We improved the original RFEX method reported in [1,3] and created RFEX 2.0 method which we wanted to test it on variety of data. The improvements in RFEX 2.0 include:

- Adding *Man Whitney Wilcox* test [5] to establish if feature values for positive and negative class are independent
- Adding *feature cliques* (groups) for the assessment of predictive powers of combinations of groups of features (this replaces MFI in original RFEX)
- In addition, as in [3], instead of feature directionality in original RFEX summary report [1] we show average and standard deviation of feature values for positive and negative class as being simpler and easier to understand measure.

In order to test whether RFEX 2.0 pipeline can be successfully applied on other data sets to offer insights (e.gt. explainability) of RF classification, we have first experimented with it on a test database we created with carefully chosen synthetic data. This report summarizes our work which includes:

a) Creation of a synthetic test database in a way to offer easy to understand ground truth data (e.g. knowing which features have most predictive powers) which then offers verifiable test for RFEX 2.0.
b) Application of RFEX 2.0 on this test database to verify that it produces expected results consistent with the known ground truth in the test database.

In the text below, RFEX will denote RFEX 2.0. In our work we used R toolkit [6]

2. Description of synthetic test database

The data set in synthetic test database consists of 1000 samples or feature vectors, each with 10 features F_i (i=1..10)), and class assignment (1 for positive class, 0 for negative). In order to use this data for testing RFEX we created it in such a way that it has some easy to assess and obvious ground truth, that is, only F1 and F2 have predictive power and can together perfectly classify the data, while F3-F10 have no predictive power. Class balance (ratio of number of positive vs. negative class samples) is set to be about 50-50%. Specifically:

1. We created two features (F1, F2) which together offer perfect classification/prediction,
2. We created other features (F3-F10) with data which are irrelevant for prediction. The reason behind this is to observe if the RFEX pipeline will be able to identify the most important features as well features that are useless.

The feature values were created as follows using R toolkit [6]:

Feature F1: Range of data between 1:10, **`sample (1:10,1000, replace=TRUE)`**
Feature F2: Range of data between 1:20, **`sample (1:20,1000, replace=TRUE)`**

After creating these two features we created the target variable as follows:

```
if (data[i,'F1'] >= 1 & data[i,'F1'] <=6 & data[i,'F2'] >= 1 &
data[i,'F2'] <=14)
     target = '1'
else
     target = '0'
```

The data for F1 and F2 are therefore structured in a "box" fashion such that typical RF can easily classify them (but not linear classifier). See Fig 1 showing distribution of data for class 0 and 1 for F1 and F2.

We chose the ranges, [1-6] for F1 and [1-14] for F2, since doing this divides the data into approximately balanced sets for the two classes.

Number of data points belonging to class '0': 574
Number of data points belonging to class '1': 426

We then created features F3-F10 using an inbuilt Excel function in [6] to create 8 normal unimodal distributions for each feature *irrespective* of class values. For each feature we set a different mean and standard deviation for feature values with *random* assignment of class. This was done so as to make these features independent of the class values, hence making F3-10 irrelevant for classification consistent with the objectives of the whole experiment (stated above).

The function we used to generate features F3-10 is:

```
=ROUND (NORM.INV(RAND (),Mean, StandardDeviation),rounding
point)
```

Here is a sample distribution of two features, F1 and F2 along with the classes

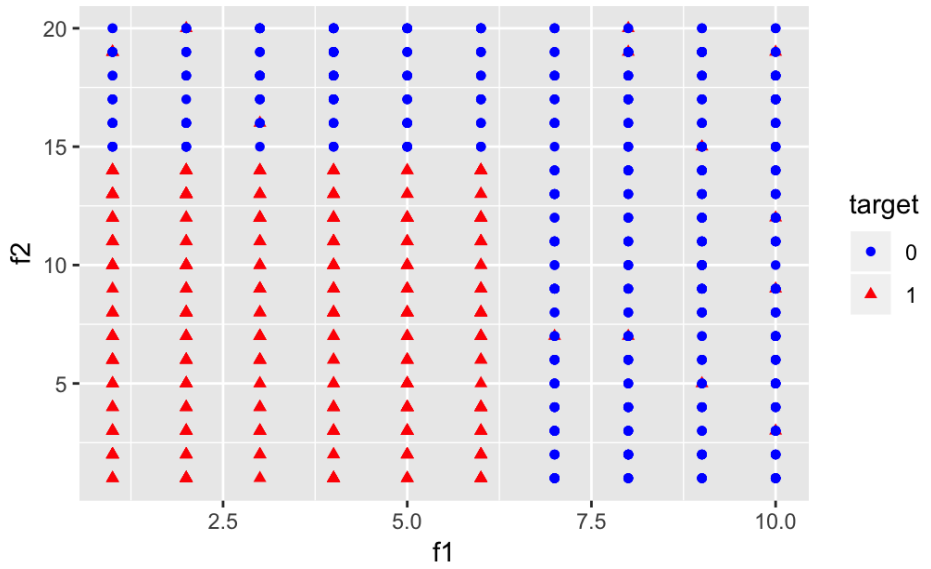


Figure 1: Scatter plot(boxy) distribution of two features in the data, with class 1 in triangles, and class 0 in circles

The plots below (Fig 2 and 3) display the histograms of feature values for F1 and F2 for each target class respectively:

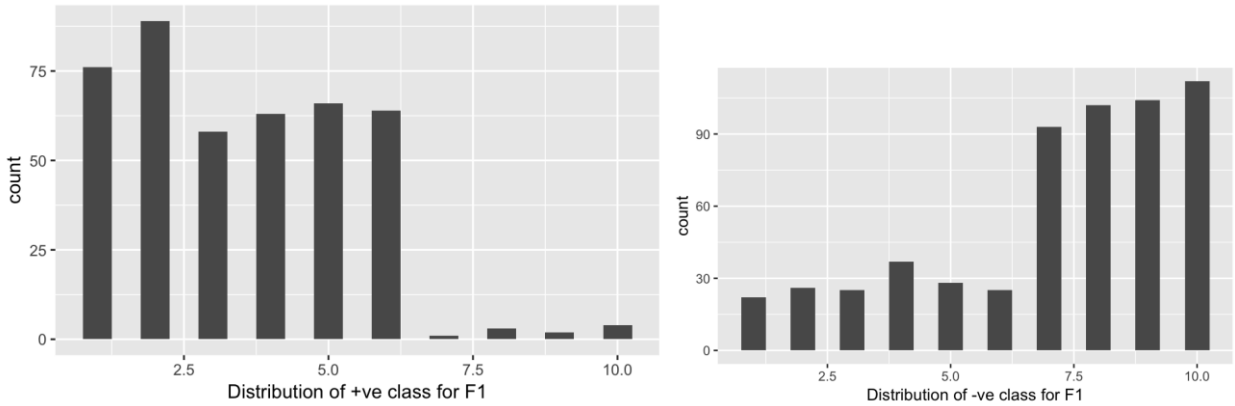


Figure 2: a) Histogram of feature values of F1, for class 1 b) Histogram of feature values of F1, for class 0

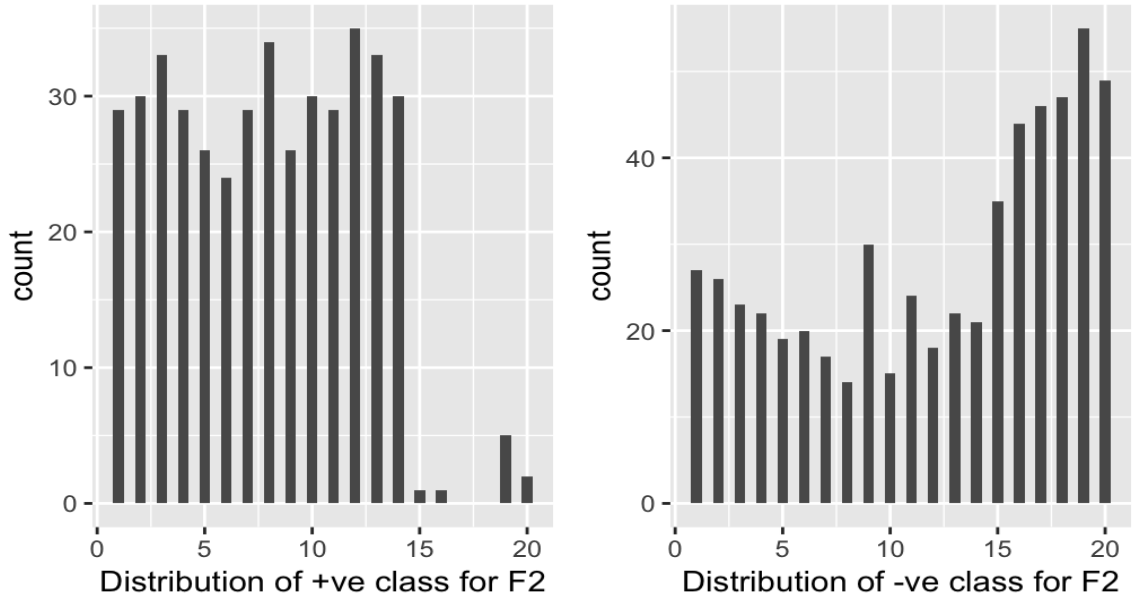
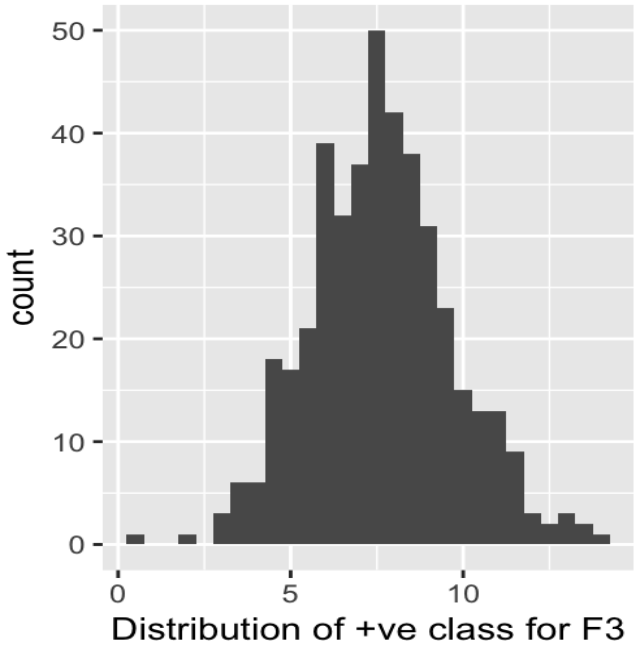


Figure 3: a) Histogram of feature values for F2, for class 1 b) Histogram of feature values for F2, for class 0

Below (Fig 4-11) are histogram plots of the features displaying unimodal distributions for features F3-F10 for each class. Note that the feature values for classes 1 and 0 have same distribution for F3-F10 making them useless for prediction.



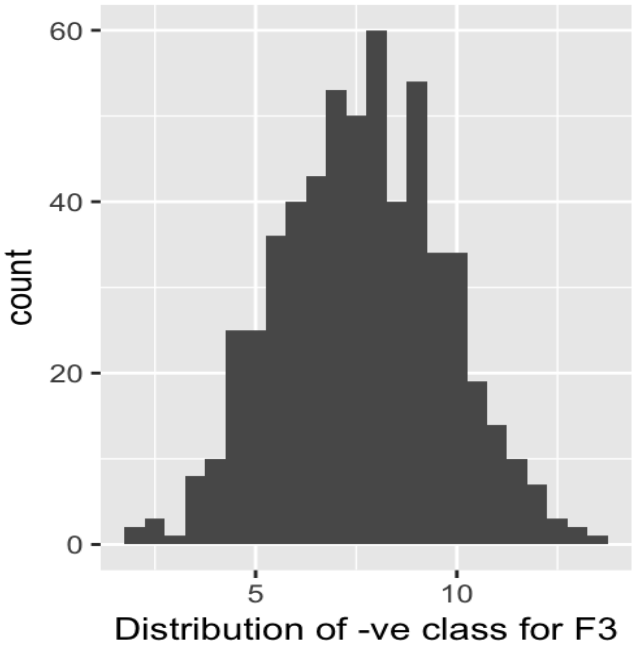
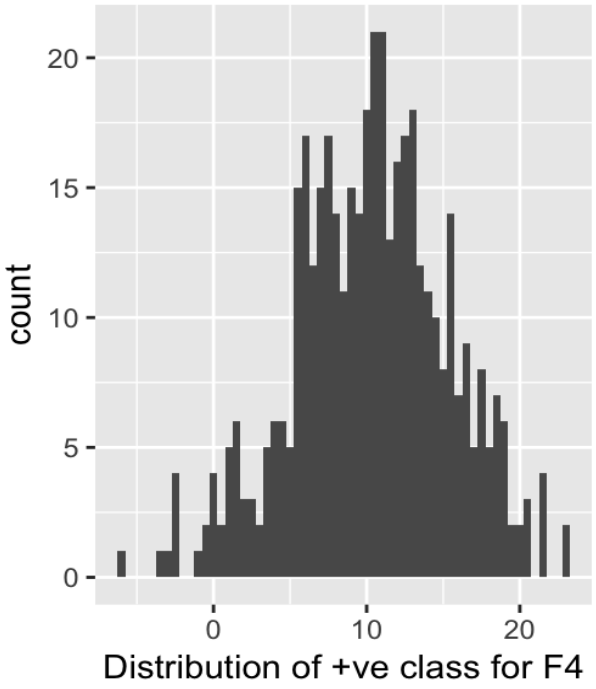


Figure 4: a) Histogram of feature values for F3, for class 1 b) Histogram of feature values for F3, for class 0



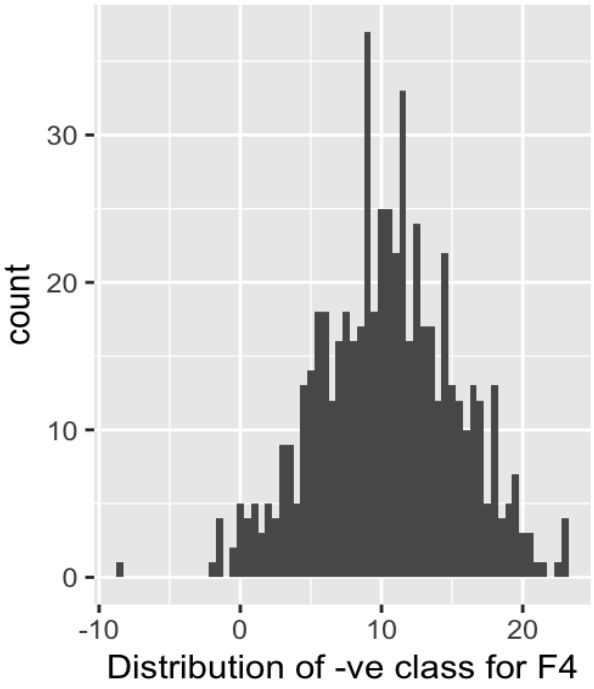
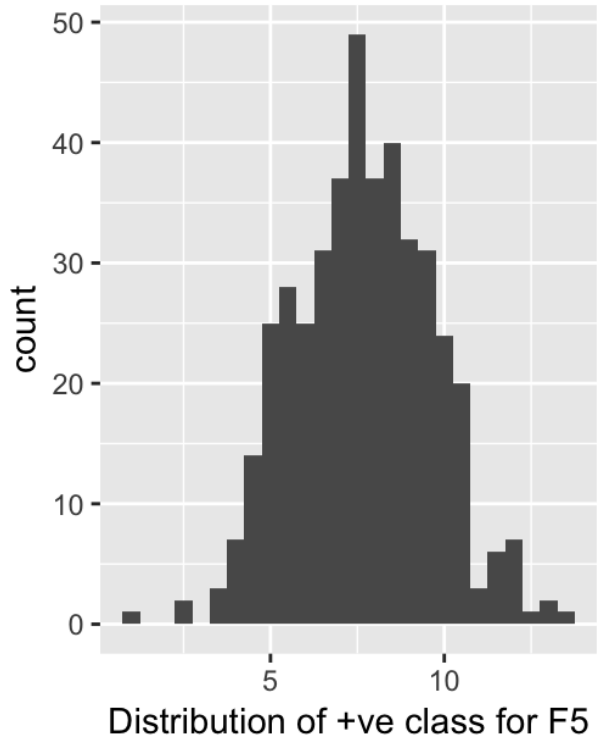


Figure 5: a) Histogram of feature values for F4, for class 1 b) Histogram of feature values for F4, for class 0



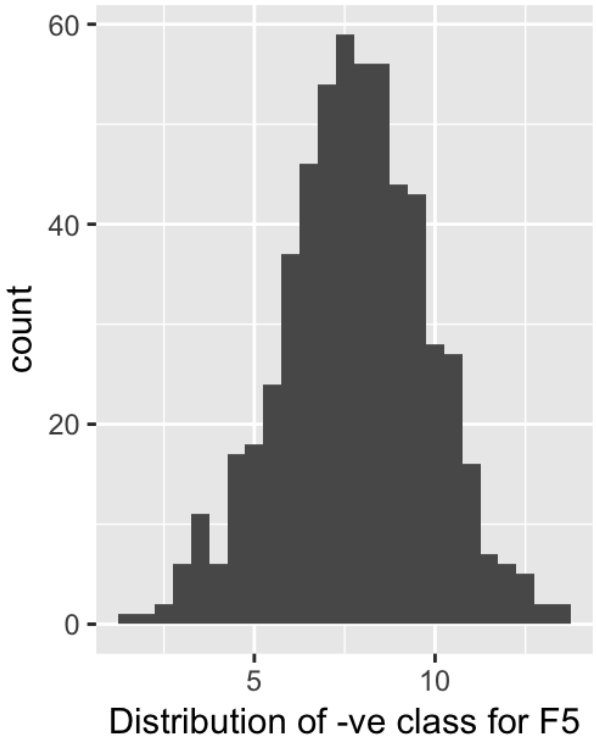
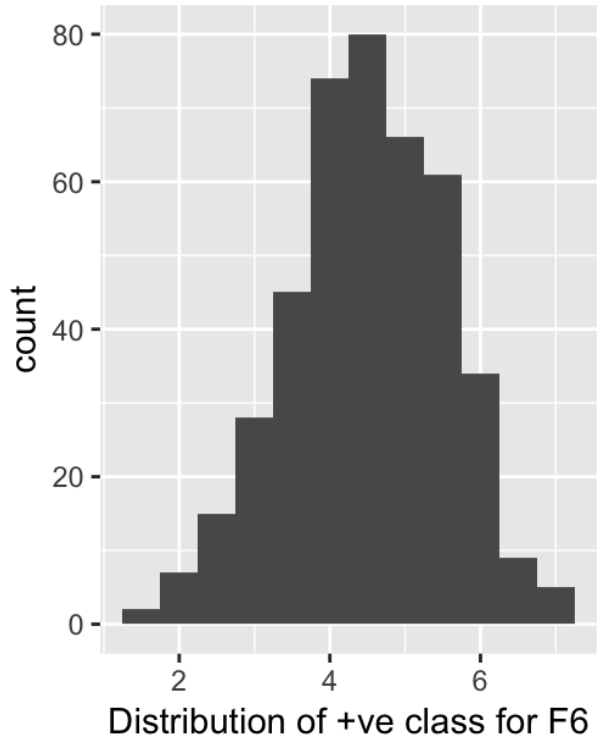


Figure 6: a) Histogram of feature values for F5, for class 1 b) Histogram of feature values for F5, for class 0



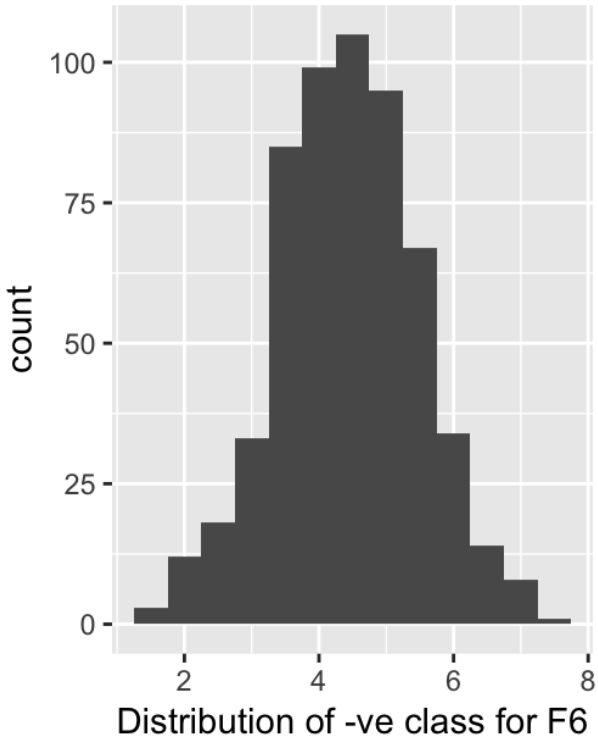


Figure 7: a) Histogram of feature values for F6, for class 1 b) Histogram of feature values for F6, for class 0

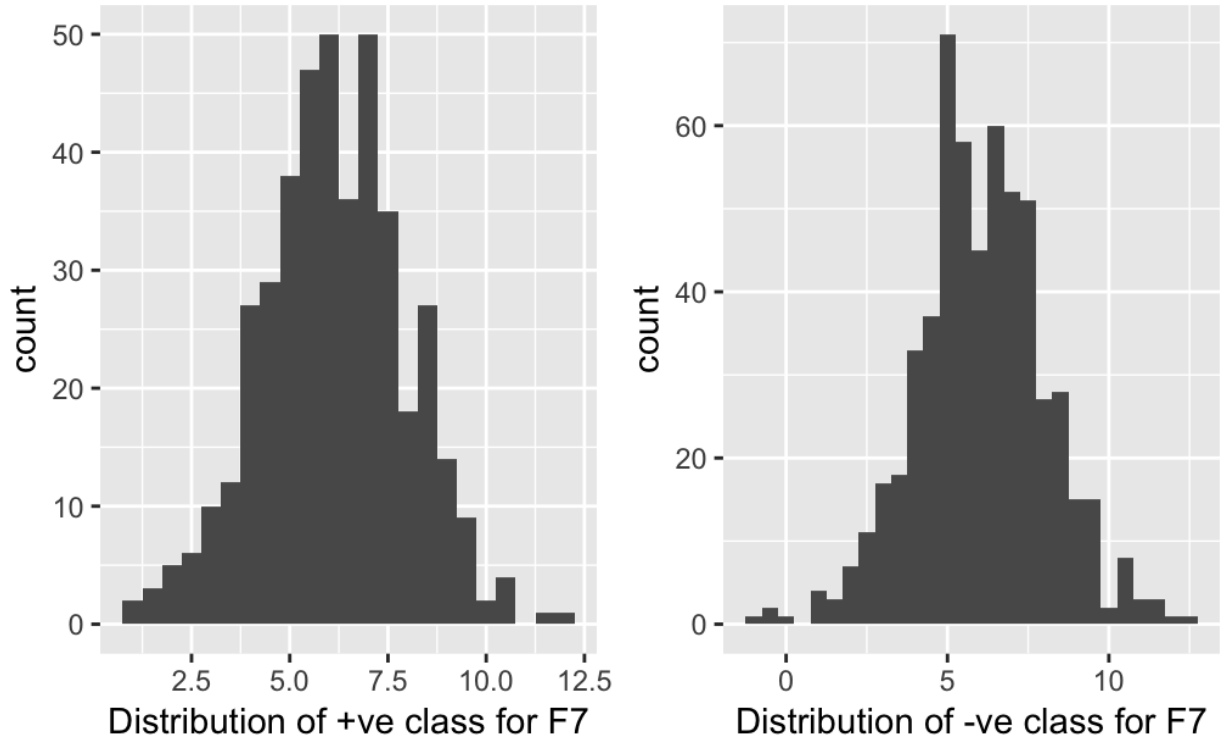


Figure 8: a) Histogram of feature values for F7, for class 1 b) Histogram of feature values for F7, for class 0

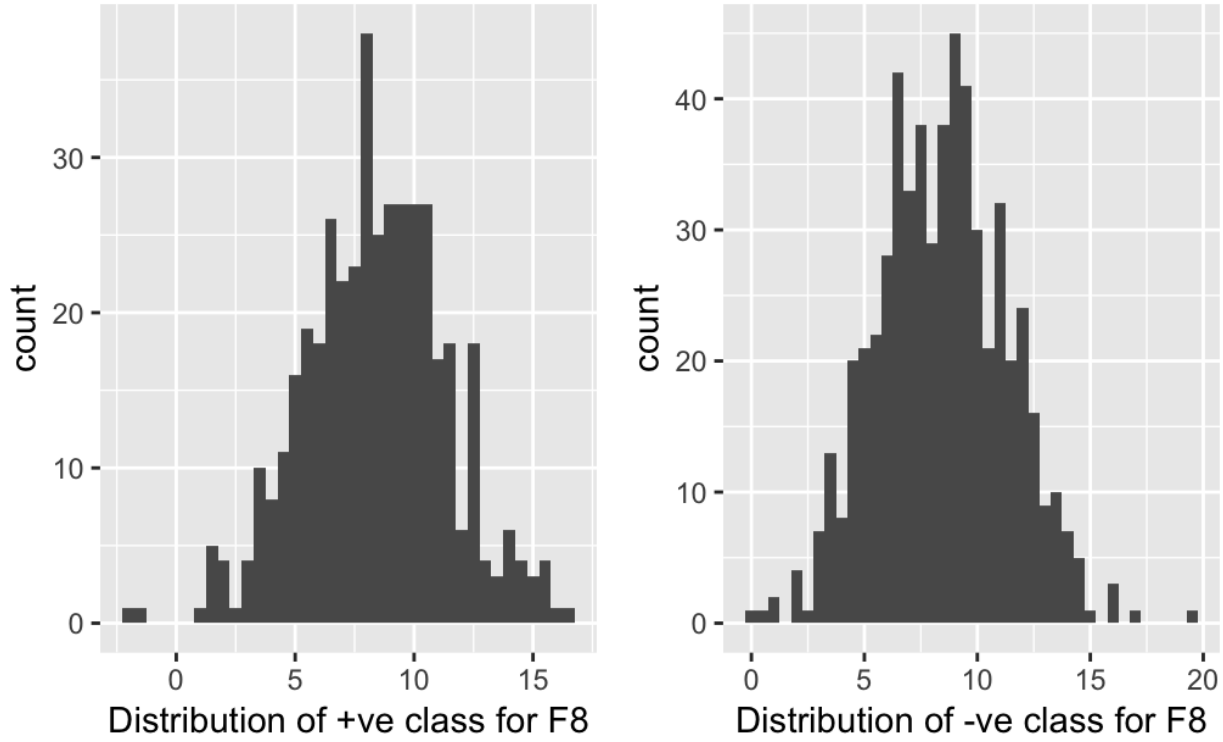


Figure 9: a) Histogram of feature values for F8, for class 1 b) Histogram of feature values for F8, for class 0

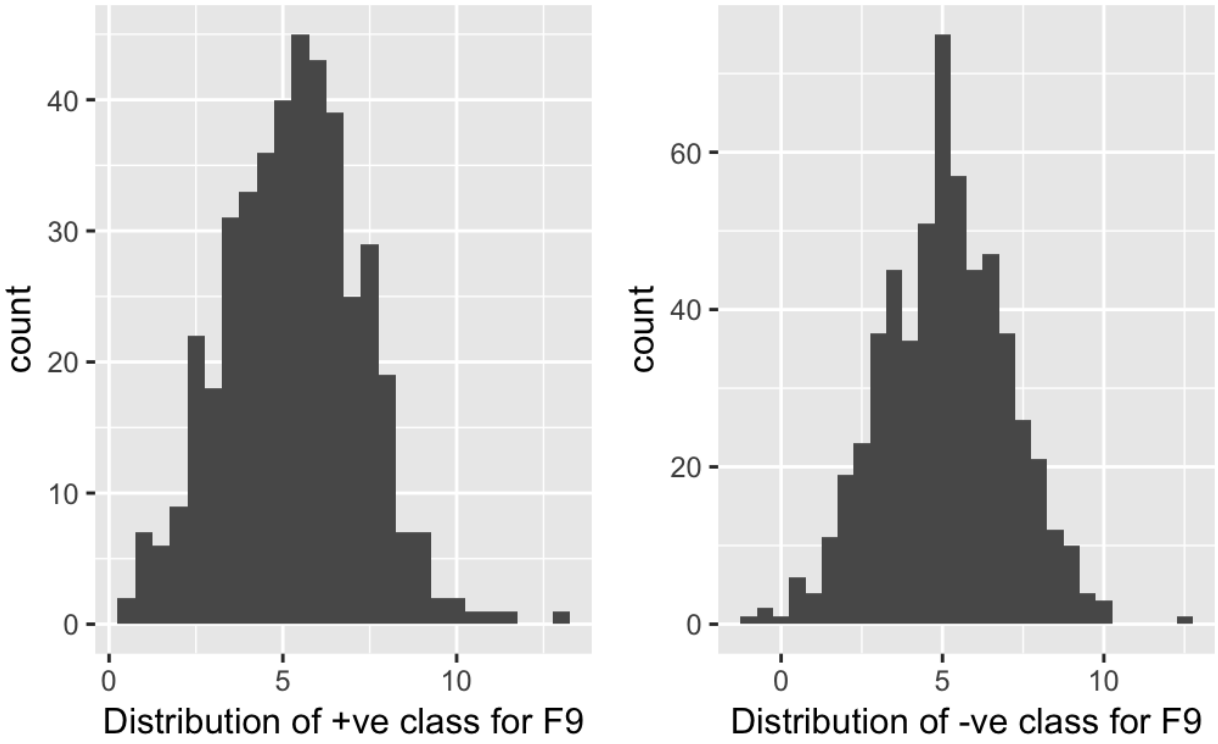


Figure 10: a) Histogram of feature values for F9, for class 1 b) Histogram of feature values for F9, for class 0

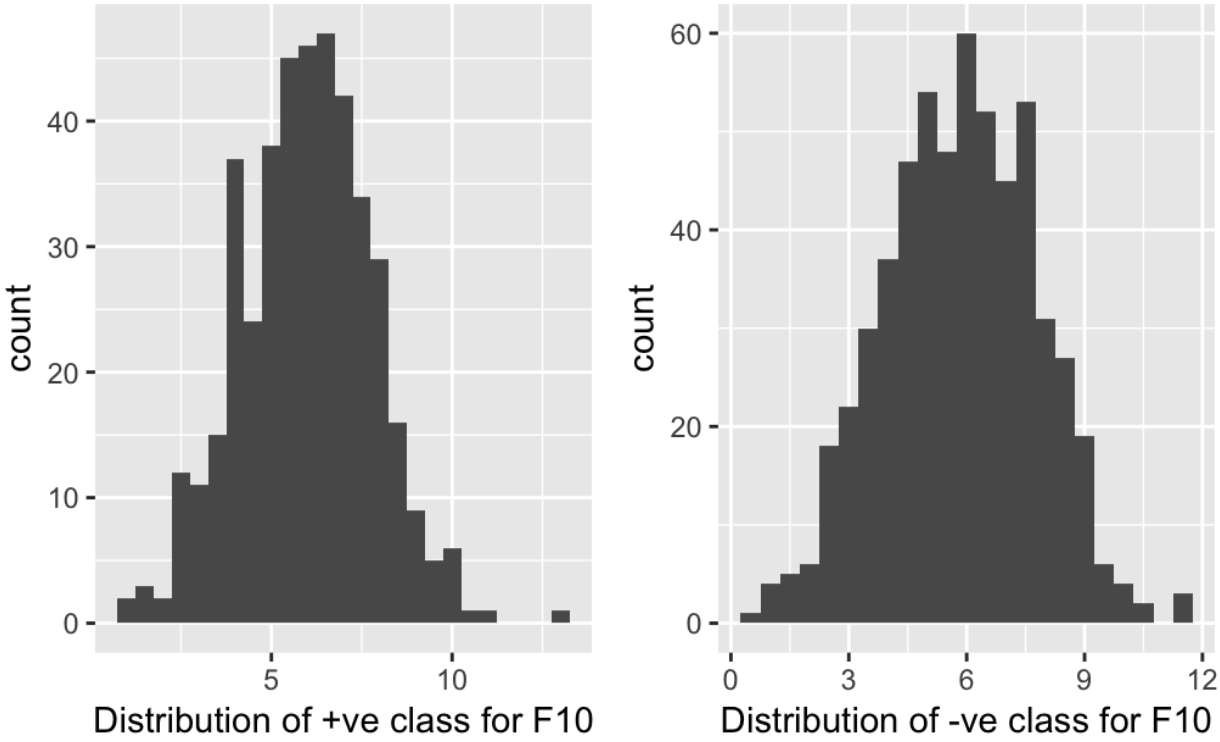


Figure 11: a) Histogram of feature values for F10, for class 1 b) Histogram of feature values for F10, for class 0

3. Methods and Algorithms used for the RFEX process

RFEX 2.0 pipeline is summarized below in 6 steps:

1. Apply Random Forest (RF) to establish base RF accuracy using all features of the data set for classification.
2. Identify the most important features using Mean Decrease in Accuracy (MDA) (or MDA specific to + class, e.g. MDA+, in case of imbalanced data) and rank the features according to the predictive power indicated by MDA.
3. Provide tradeoff between accuracy and the number of top ranked features used.
4. Apply Man Whitney White (MWW) to each feature to test if the features values for + and – class are independent of each other (NEW in RFEX 2.0).
5. Identify the most important combination of features e.g. cliques of 2 features with highest predictive powers (NEW in RFEX 2.0).
6. Produce a one-page RFEX summary showing above information in easy to understand way (includes AV/SD of feature values for positive and negative class)

Below we describe each of these steps in more details.

Step 1: Apply Random Forest (RF) to establish base Accuracy using all features of the data set for classification

The first step is to apply Random Forest algorithm [2] to find out the best accuracy using f1-score as our main evaluation measure, and also report recall, precision and OOB error. This is achieved by grid search over Ntree, Mtry and Cutoff parameters [2]. These parameters are briefly described below.

Ntree: Number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times.

Mtry: Number of variables randomly sampled as candidates at each split.

Cutoff: Cutoff is the vector of length equal to the number of classes (in this case 2).
In the data set, the number of data points belonging to the positive class (1 class) is 574 and number of data points belonging to the negative class (0 class) is 426.
Changing the cutoff changes, the voting ratio in the ensemble of Ntrees, which then changes the "sensitivity" e.g. recall and precision or f1 score.

f1-score: The f1-score is used to evaluate a classification algorithm's performance. It is the harmonic mean of **precision** (the number of true positives divided by the sum of true and false positives) and **recall** (the number of true positives divided by the number of true positives and false negatives) for calculating the score**.** The formula for calculating f1-score is:

```
2*((recall*precision)/(precision+recall))
```

We performed grid search varying the mtry, ntree and the cutoff in order to record the best f1 scores. The table below records best f1 score values and also lists all the combinations we have used:

Confusion Matrix:

	Predicted 0	Predicted 1	OOB Error
Actual 0	574	0	0.00
Actual 1	15	411	0.035

Table 1: Confusion Matrix after training the data with RF classifier

f1-score	0.982
Accuracy	0.985
Precision	1
Recall	0.964
OOB Error	1.5%
RF Parameters	
Best ntree	500
mtry range	2,3,4,5,6
Best mtry	5
cutoff range	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Best cutoff	0.5 for class 0, 0.5 for class 1

Table 2: Scores after fitting the data with RF classifier

***Observation*:** It can be observed that we get a high f1-score and a precision of 1, which reflects the fact that the algorithm is performing very well which is to be expected by the way we created the data. This step validates our expectation that RF should be able to predict (classify) well the data in this test database.

Step 2: Identify the most important features using Mean Decrease in Accuracy (MDA) and rank the features according to the predictive power.

The mean decrease in accuracy (MDA) is calculated during the Out of Bag error calculation. If by permuting a feature value, the accuracy of the random forest decreases then it indicates the importance of that feature. This is tried in all trees and MDA is averaged for each feature. The more important the feature is, the higher its MDA value.

In R [6], MDA is calculated using "importance" variable in [6] and using the following syntax:

```
importance (x, type=NULL, class=NULL)

x is an object belonging to the RF class.
Type = 1 specifies MDA
Class is the target class of the data. Class specific MDA can be calculated
by specifying the target class in the class variable.
```

Note that MDA can be computed for each class separately (denoted as MDA+ and MDA -) which, based on our previous results in [1] we recommend if classes are imbalanced (not our case here). Following are the MDA values for the 10 features recorded in the tables below. By applying the RFEX pipeline step of predicting the important features, it was found that F1 and F2 are indeed the most important features indicated by them having by far the largest MDA

values. (In our previous work we calculated MDA of all the features and also for each class separately and repeated it 5 times and found that the MDA values and rankings are very similar).

Features	MDA general
F1	377.28
F2	293.7
F3	5.73
F6	3.55
F8	1.34
F4	0.986
F5	0.15
F9	-0.13
F7	-2.72
F10	-4.59

Table 3: MDA ranking of features for both the classes

Features	MDA Value(+)
F1	312.8
F2	234.27
F9	3.94
F3	3.26
F7	1.93
F4	0.14
F8	0.05
F6	-0.059
F5	-1.122
F10	-1.47

Table 4: MDA+ ranking of features for positive (1) class

Features	MDA Value(-)
F1	306.22
F2	244.35
F3	5.11
F6	4
F8	1.4
F4	1.02
F5	0.759
F9	-1.489
F7	-4.15
F10	-4.48

Table 5: MDA- ranking of features for negative (0) class

***Observation*:** Observing the above tables for MDA+ and MDA- compared to MDA general, we can see that F1 and F2 are by far the most and only important features for both the positive and negative class as well, as it is to be expected from the way we generated test data (note they have by far the largest MDA values). We also see that MDA+, MDA – and MDA ranking are almost similar, which is to be expected in case of balanced data set.

Step 3: Provide tradeoffs between number of features and f1-score:

The goal of this crucial step is to reduce complexity (using feature reduction) and to provide a tradeoff where the user can observe the f1-score of the subsets of K top ranked features and thus

can choose less features for a given loss of accuracy (e.g. drop in f1 value from the base accuracy when all features are used). This step is performed as follows:

Let TopN be the number of features ranked highly by MDA in Step 2 (TopN = 10 in our case)

```
For K = 2 to K=TopN
 Train RF using only K top ranked features (vary ntree, mtry
 and cutoff to get max f1-score)
 Record maximal f1-score(K) for set of K top ranked features
```

The figure 12 below displays the tradeoff.

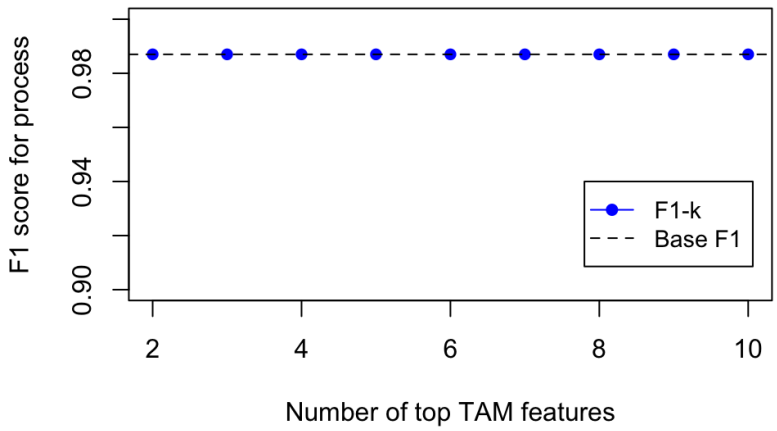


Fig 12: Tradeoff between f1-score(K) using top K ranked features. Base f1-score (0.98) using all features is at dotted line

***Observation*:** Here, we can observe that we get the same f1-score using only top two features (F1 and F2) as using all the features. This proves that F3-F10 are irrelevant for prediction as expected, and that we can get the desired f1-score using only the top 2 important features. This is consistent with the way we generated our test database

Step 4: Apply of Man Whitney Wilcox (MWW) test [5] to identify if the top ranked features are independent of each other

Man-Whitney-Wilcox Test is a test of a hypothesis of the distribution of data [5]. It is a non-parametric test of assuming a null hypothesis of class specific samples not being independent from each other. It doesn't assume a normal distribution and can also be used to assess if two independent samples were selected from populations having the same distribution. We applied MWW test for each feature to determine if feature values from one class are independent from feature values of another class.

Null Hypothesis: The values of the MDA ranked features of class 0 and 1 are not independent of each other.

Alternative Hypothesis: The values of the MDA ranked features of class 0 and 1 are independent of each other.

To test the independence, we use the **Wilcox.Test** function in R [6], to calculate the p-values. If p-value < 0.05, we can reject the null and can say that the feature values are independent of each other.

The syntax of the MWW function in [6] is:

wilcox.test(pos_class_process[,i,], (neg_class_process[,i]))

where, pos_class and neg_class are the both ‘1’ and ‘0’ classes for each feature.

Features	p-values	Independent
F1	6.655816e-84	Yes
F2	1.303722e-31	Yes
F3	0.7208	No
F4	0.9052	No
F5	0.3677	No
F6	0.2554	No
F7	0.477	No
F8	0.8033	No
F9	0.0984	No
F10	0.2906	No

Table 6: MWW test results displaying whether the features are independent of each other for both the classes

***Observation*:** It can be seen that indeed the most important features, F1 and F2 are independent of each class as expected and mentioned in the objective of the MWW test data above. The rest of the 8 features are not independent of each class (as expected in the way we generated the data), which causes them to be irrelevant for prediction.

Step 5: Identify the most important combination of features e.g. cliques of 2 features on the top most important features.

In order to get some idea which combinations of features are most powerful we considered a clique of 2 features which is a set of all combinations of 2 features. In our case we could try all combinations e.g. for a total of 45 feature combinations for our set of 10 features (e.g. 10 choose 2). We retrained RF Classifier for each of these combinations in order to find the pair of features (clique of 2) with the highest f1-score. Results are presented in the table below.

Performing grid search, we varied the parameters for each pair individually (as below) for each of 45 cliques of 2 features, and the parameters that gave the best f1-scores are given below:

Ntree = [100-500], Best ntree = 100
Mtry = 1,2, (Best mtry = 2)
Cutoff = [0.1-0.9] for both the classes, Best cutoff = (0.5,0.5) and (0.6,0.4)

Feature 1	Feature 2	f1-score
F1	F2	0.98207885
F1	F3	0.82703777
F1	F4	0.82703777
F1	F8	0.82669323
F1	F9	0.82621648
F1	F10	0.82517483
F1	F6	0.8243513
F1	F7	0.82317682
F1	F5	0.822
F2	F8	0.69419862
F2	F5	0.69253152
F2	F4	0.68560235
F2	F7	0.67826087
F2	F10	0.6744186
F2	F3	0.67116683
F2	F9	0.66995074
F2	F6	0.66601563
F5	F10	0.5042735
F4	F5	0.50207469
F4	F6	0.50104822
F4	F7	0.49361702
F6	F8	0.48940678
F4	F8	0.48427673
F3	F4	0.48275862
F3	F10	0.48082902
F6	F7	0.47767394
F5	F9	0.47748691
F7	F8	0.47668394
F4	F10	0.4748954
F9	F10	0.47144341
F5	F7	0.47046414
F4	F9	0.47034339
F3	F7	0.47010309
F7	F10	0.4697286
F3	F5	0.46589717
F8	F9	0.46218487
F5	F6	0.45828933
F3	F8	0.45726496
F5	F8	0.45483528
F3	F6	0.45228216
F7	F9	0.44536082
F6	F9	0.4375645
F6	F10	0.4375645
F8	F10	0.434238
F3	F9	0.4338843

Table 7: Clique = 2 results for features and their f1 scores in decreasing order

***Observation*:** The highest f1-score for cliques of 2 observed by far is (F1, F2), which is as expected. The likelihood of a high f1-score for feature F1 with other features is better as compared to F2. Since the MDA value of F1 is higher than F2 (MDA(F1) > MDA(F2)), these results are in line with the objectives and goals of the experiment. Note that it examples with real data, when one has large number of features, thanks to our RFEX step 3 where we perform drastic feature reduction, it is feasible to test cliques of 2 features for a reasonable number of top K ranked features, where K can be on the order of 10-30 or so.

Step 6: Produce a One-page easy to read and understand RFEX Summary of the data

RFEX 2.0 summary table is presented below. It differs from RFEX 1.0 summary table in the following:

- Columns 4 and 5 contain AV/SD of feature values for positive and negative class. We felt that this is more informative and natural than presenting *feature direction* as in RFEX 1.0
- Column 6 contains YES for the case when feature values for positive vs. negative class are independent, and NO otherwise, as measured by MWW p test.
- Column 7 shows which feature forms best clique of 2 with current feature, and f1 score for that clique.

Feature Name	MDA value	f1 score-current and all features ranked above it	AV/SD for positive class	AV/SD for negative class	Positive and negative class feature values are independent (MWW test; p<0.05)	Best feature forming cliques of 2 with current feature and its f1
F1	377.28	N/A	3.48/1.91	7.16/2.58	Yes	F2 (0.982)
F2	293.70	0.982	7.842/4.34	12.27/6.13	Yes	F1 (0.982)
F3	5.73	0.982	7.61/2.071	7.63/2.04	No	F1 (0.827)
F4	0.986	0.982	10.29/5.08	10.38/4.9	No	F1 (0.827)
F5	0.152	0.982	7.67/1.987	7.77/2.033	No	F1 (0.822)
F6	3.55	0.982	4.493/1.037	4.42/1.04	No	F1 (0.824)
F7	-2.72	0.982	6.122/1.84	6.04/2.031	No	F1 (0.823)
F8	1.34	0.982	8.369/3.00	8.45/2.86	No	F1 (0.826)
F9	-0.13	0.982	5.304/1.96	5.08/1.97	No	F1 (0.826)
F10	-4.59	0.982	5.96/1.805	5.82/1.929	No	F1 (0.825)

Table 8: RFEX Summary showing the predictive features and all the metrics, AV/SD, p-value, Clique combination

Knowing the ground truth for our test database we easily see that RFEX summary report reflects it well:

- MDA values for F1 and F2, the only predictive features, are much higher than any other MDA value
- Using only top 2 ranked features, namely F1 and F2, produces accuracy as if all features were used, meaning F3-10 are useless for prediction
- Feature values for positive and negative class are independent only for F1 and F2, and are not independent for F3-10, as we set them in the test database
- Clique of F1, F2 is by far the best combination of 2 features.

4. Conclusions

In this report we outlined improved RFEX pipeline (RFEX 2.0) and applied it to carefully designed synthetic test database whose ground truth we can assess easily. Our results confirm that new RFEX 2.0 method produces summary consistent with ground truth in the test database. Hence, this report indicates validity of RFEX 2.0 and also to help explain how it works. Our next work will focus on testing RFEX 2.0 on other real data sets.

5. References:

[1]. D. Petkovic, R. Altman, M. Wong, A. Vigil: "Improving the explainability of Random Forest classifier – user centered approach", Pacific Symposium on Biocomputing PSB 2018, Hawaii

[2]. L. Breiman, "Random Forests," Machine Learning 45(1). 2001. pp. 5–32.

[3]. D. Petkovic, S. Barlaskar, J. Yang, R. Todtenhoefer: "From Explaining How Random Forest Classifier Predicts Learning of Software Engineering Teamwork to Guidance for Educators" Frontiers of Education FIE 2018, October 2018, San Jose CA

[4]. SETAP database at UC Irvine Machine learning Archive ([https://archive.ics.uci.edu/ml/datasets/Data+for+Software+Engineering+Teamwork+Assessment+in+Education+Setting]).

[5]. Mann Whitney – Wilcox U test – Wikipedia – accessed 11/15/18 https://en.wikipedia.org/wiki/Mann%E2%80%93Whitney_U_test

[6] R toolkit: R Core Team, R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. 2013. http://www.R-project.org