Predicting post-release defects from Eclipse

Vinod Rajendran

The Eclipse 3.0 dataset lists the number of pre- and post-release defects along with other complex metrics for every package and file in the Eclipse. Although this dataset has scope for wider research, in this work a classifier models are build to predict how many post-release bug defects can be found from a given file and package.

Initially, the data was analyzed using RapidMiner. The dataset has totally 10593 samples with 201 features and 1 label. From RapidMiner, the following observations are made.

- The histogram representation of post-defects indicates that the number of defects biased between 0 to 3. The histogram representation of post defects is depicted in Fig 1.
- 42 features present in this datasets has only 0 values and 2 other features has same values for all samples. The complete list of features can be found in Table 1.

As a first step of pre-processing, all the irrelevant features i.e 44 features are removed from the dataset. With 157 features, the initial classifier models are designed. The train and test set split are performed using cross validation.

Firstly, the tree-like graphical model decision tree was experimented. Following that some of its variants like Random Forest and Extra Tree classifier models are experimented. Compared to decision tree, the Random Forest and Extra Tree classifiers were better and able to obtain 85 % accuracy. In oder to improve the accuracy three complex methods namely Adaboost, SVM and Deep neural network (DNN) are performed. However, all three results were close to 85 % accuracy. It is important to note that, the data were normalized for these three methods. Complete results of each classifier can be found in Table 2.

Since the above experimented model doesn't contribute much to the accuracy, the selection of relevant features for classifiers seems to be an obvious choice. Principal component analysis (PCA) is one such technique which able to reduce the dimension of data with minimal loss of information. Using PCA with randomized SVD, the input data has been transformed to 60 features. Therefore, all of the above mentioned classifier methods are re-evaluated with the transformed data. But the results are not as expected, only a minor changes were noted. The complete results can be found in Table 4.

The main problem lies with this dataset is the imbalanced classes of the post-defects between 0 to 3 numbers. The classification report of Random Forest classifier is generated for analyzing the imbalances. As shown in Table 3, from post-defect number 4 onwards 0 cases were reported in the classifier.

In order to overcome this problem, following steps were performed.

- the data were grouped according to each label.
- With 17 independent sets, cross validation is applied on each of them to obtain train and test sets.
- Finally, all the train sets are joined to perform training and similarly for test sets to do testing.

As shown in Table 6, only some minor changes in selection of training and testing data were obtained. Apart from that in terms of precision,recall, F1 score and accuracy not much change is noticed. The complete result can be found in Table 5.

Future Work

To overcome the problem of imbalanced classes, methods like SMOTE,costSensitive classifier, etc. needs to be investigated.

Although the accuracy of Deep Learning method - DNN was close to 85 %, it needs more training data to get better results. Therefore data augmentation techniques need to be investigated.

Appendix

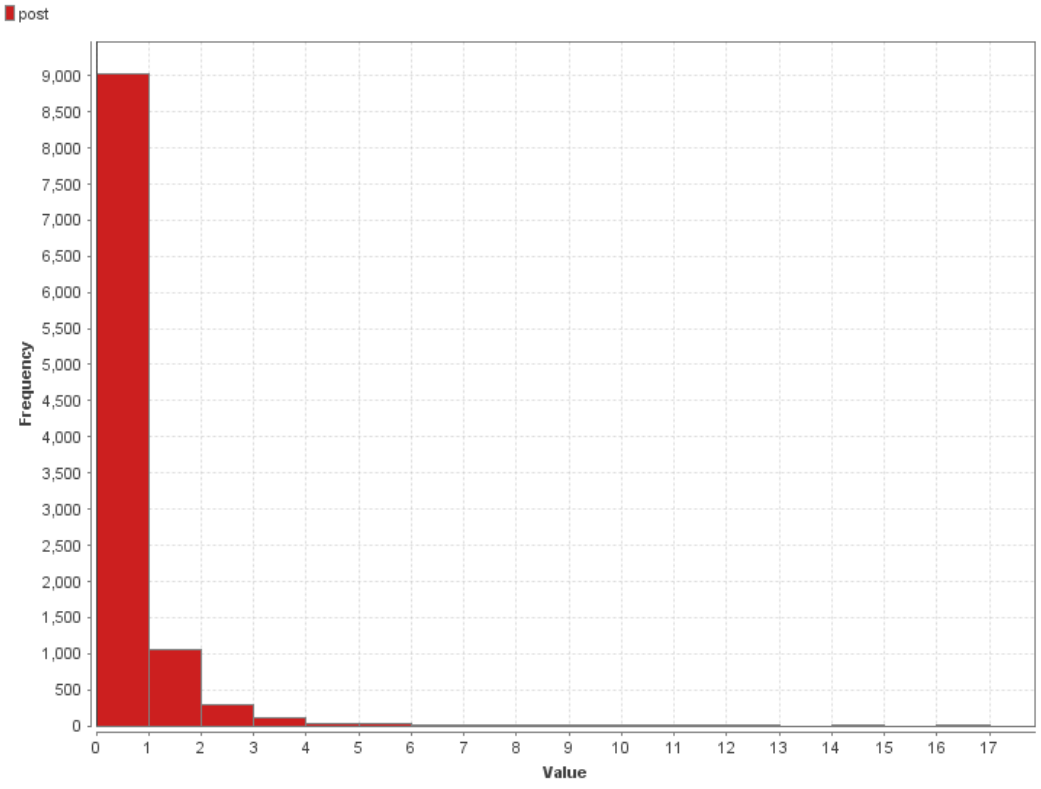

Fig. 1. Histogram representation of post defects

Table 1. List of irrelevant features

LineComment
BlockComment
TagElement
TextElement
MemberRef
MethodRef
MethodRefParameter
EnhancedForStatement
EnumDeclaration
EnumConstantDeclaration
TypeParameter
ParameterizedType
QualifiedType
WildcardType
NormalAnnotation
MarkerAnnotation
SingleMemberAnnotation
MemberValuePair
AnnotationTypeDeclaration
AnnotationTypeMemberDeclaration
NORM_LineComment
NORM_BlockComment
NORM_TagElement
NORM_TextElement
NORM_MemberRef
NORM_MethodRef
NORM_MethodRefParameter
NORM_EnhancedForStatement
NORM_EnumDeclaration
NORM_EnumConstantDeclaration
NORM_TypeParameter
NORM_ParameterizedType
NORM_QualifiedType
NORM_WildcardType
NORM_NormalAnnotation
NORM_MarkerAnnotation
NORM_SingleMemberAnnotation
NORM_MemberValuePair
NORM_AnnotationTypeDeclaration
NORM_AnnotationTypeMemberDeclaration
AssertStatement
NORM_AssertStatement
PackageDeclaration
CompilationUnit

Table 2. Results of classifier models

Learning algorithm	**Accuracy**	**Precision**	**Recall**	**F1 score**
Decision Tree	77.62	77.80	77.62	77.71
Random Forest	85.05	79.08	85.05	79.47
Extra Tree	85.11	79.92	85.11	79.94
Adaboost	84.64	71.75	84.64	77.67
SVM	85.02	80.4	85.02	79.72
DNN	84.64	-	-	-

Table 3. Random Forest classification report - Random Sampling

Post (target)	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.86	1.00	0.92	2690
1	0.44	0.06	0.11	335
2	0.57	0.05	0.09	85
3	0.25	0.03	0.05	34
4	0.00	0.00	0.00	10
5	0.00	0.00	0.00	12
6	0.00	0.00	0.00	3
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	1
9	0.00	0.00	0.00	2
10	0.00	0.00	0.00	1
12	0.00	0.00	0.00	1
17	0.00	0.00	0.00	1
avg/total	0.79	0.85	0.79	3178

Table 4. Results of classifier models with PCA

Learning algorithm	**Accuracy**	**Precision**	**Recall**	**F1 score**
Decision Tree	75.95	77.1	75.95	76.51
Random Forest	85.05	80.10	85.05	79.27
Extra Tree	85.17	81.13	85.17	79.51
Adaboost	84.64	71.64	84.64	77.60
SVM	83.32	74.80	83.32	78.07
DNN	84.70	-	-	-

Table 5. Results of classifier with even sampling of data

Learning algorithm	**Accuracy**	**Precision**	**Recall**	**F1 score**
Decision Tree	77.76	78.80	77.76	78.26
Random Forest	85.74	80.77	85.74	80.58
Extra Tree	85.61	80.70	85.61	80.58
Adaboost	84.98	72.37	84.98	78.17
SVM	84.79	79.71	84.79	78.68
DNN	84.83	-	-	-

Table 6. Random Forest classification report - Even Sampling

Post (target)	**Precision**	**Recall**	**F1-Score**	**Support**
0	0.86	1.00	0.93	2708
1	0.55	0.09	0.15	315
2	0.45	0.06	0.10	87
3	0.50	0.06	0.11	33
4	0.00	0.00	0.00	12
5	0.00	0.00	0.00	11
6	0.00	0.00	0.00	5
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	2
9	0.00	0.00	0.00	2
10	0.00	0.00	0.00	1
11	0.00	0.00	0.00	1
12	0.00	0.00	0.00	1
14	0.00	0.00	0.00	1
16	0.00	0.00	0.00	1
17	0.00	0.00	0.00	1
avg/total	0.81	0.86	0.81	3184