mrinal

by Misha Kakkar

Submission date: 23-Apr-2018 12:41PM (UTC+0530)
Submission ID: 476734988
File name: Copy_of_new.docx (801.84K)
Word count: 5616
Character count: 28274

1. INTRODUCTION

1.1 Introduction

Numerous models to predict defects have been published during the last 10 years. Classifiers used in these models are reported to be almost similar with the exception of rare cases where models perform above the ceiling of predictive performance which is about 80% recall. In this project the level of prediction uncertainty produced by the four classifiers under consideration and the individual defects that each classifier detects is analysed. Performance of three models under consideration, namely Random Forest, Logistic Regression and SVM are compared using sensitivity analysis. The prediction uncertainty of each classifier is compared by capturing the defect prediction made by each classifier in a confusion matrix. Even though the predictive performance of all the three classifiers is almost same, each detects different sets of defects. Consistency of classifiers also varies from model to model. Some are more consistent as compared to others. This project tries to confirm that different classifiers can detect unique subset of defects and also tries to see how effective the classifier ensembles are.

1.2 Project Objective

To predict the predication accuracy of three classifiers (Random Forest, Logistic Regression and SVM) and to investigate the categories of individual defects those are predicted by them. As from previous studies it is quite visible that the defects detected by each classifier are different even though the prediction accuracy is almost same. This project will try and categorise the defects detected by each classifier.

1.3 Scope

Defects in software cost industry in billions of dollars each year. If we can find these defects before delivery then these can be fixed in a fraction of post delivery fixed costs. In this

project attempt will be made to find out best performing techniques (classification) for software defect prediction. This project goes further by looking beyond the numbers, by looking at the specific defects detected or not detected by the specific classifiers. Even though the predictive power of almost all models is similar but there is a difference between defects detected and not detected by each of them. Flipping is also investigated. This project builds on the Panichella's study in 2014 where he detected classifiers and their associated detectable defect sets. This project further investigates whether different classifiers are equally consistent in their predictive performances. Results from here show that the way using ensembles and considering flipping is the future of building high performing models.

1.4 Constraints

- Certain datasets are difficult to learn from than others which implies that datasets can have a great impact on predictive performance.
- Quality of data should be considered, since normally datasets are noisy and contain outliers and missing values that can skew results.
- Features of data have to be considered. Normally related variables bias the prediction of models.
- Data balance is significant. As usually the number of non defective units is way more than that that of defective ones. PC2 NASA dataset is the example. In it defective class contains only 0.4% data points.

1.5 Risks

One step in a multistage process is classification. Especially activities that improve performance such as data pre-processing which includes the removal of non-informative features of discretion of continuous variables can elevate the performance of certain classifiers. However, technique addressing data imbalance will not be implemented because it does not necessarily impact all the classifiers in equal amount. Only partial feature reduction will be implemented. The effect of model building and data pre-processing techniques used are most probably not supposed to affect the results significantly. This could be due to the ceiling effect.

Our study will also be limited in the way that we investigated only three classifiers. It is a possibility that the classifiers we don't select may show different kind of variance in defect

subset as compared to these. We assume that this is very unlikely, as in terms of the prediction approaches used the four classifiers we have chosen are representative of different groups.

We will also use a limited number of datasets. Again, there is a possibility of other datasets behaving differently. We assume that this is very unlikely, as the 8 datasets we will investigate are very wide ranging in terms of their features.

1.6 Chapters of The Report

Second chapter of this report is 'Literature Review' which is further divided into subsections 'Existing Work', 'Datasets' and 'Classifiers'. This chapter tries to explain the evolution of the components on which the data science process is completely dependent and further the datasets and algorithms used are described in detail.

Third chapter is 'Tools and Methodology' which is further divided into 'Tools and Technology' and 'Modus Operandi of Data Science'. This chapter puts eye on the technology used during the project including hardware and software used and also explains the flow of the data science problem and steps involved in it.

Fourth chapter is 'Experiments and Results'. This explains the steps performed for the completion of project and finally has Venn diagrams thus made and the accuracy of prediction obtained across classifiers.

Fifth chapter is 'Conclusion and Future Scope'. As the name suggests it contains two sections having the conclusion of the results obtained and then the possibility of the improvement of the process and thus the results obtained.

2. LITERATURE REVIEW

2.1 Existing Work

From mere taking actions on the basis of instincts we have come a long way to using data science techniques for this judgement. Starting from the very basic mathematical techniques

to look at trends and then make future decisions we have started using various technological intense tools to understand the patterns/trends in data.

We now a day's even prepare models, machine learning models which once trained do not require further human efforts for the decision making among numerous alternatives and with the more possibility of the selected decision to be correct as compared to what we would have decided to do on the basis of the instincts. Guess what? It is still the learning phase. There is still a lot which we have not even considered rather thought of yet.

There are various aspects of defect models which have impact on predictive performance of the classifier.

Independent variables used previously are of the two categories, namely product metrics (e.g. static code) and process metrics(e.g. previous changes and defect data).

LOC is the most commonly used static code matrix. How effective LOC is as an independent variable is still unclear. Z Hang reported LOC to be usefulwhereas Bell reported LOC to have poor predictive power. Malhotra suggests object oriented metrics like response for a class and coupling between objects useful.

Process data in form of previous history data works well as suggested be previous studies. Previous bug reports are the best predictors as predicted by D'Ambros. Nagappan used more sophisticated 'change burst' metrics and reported good results. But conflicting results were reported by Ostrand. Bird reported positive results. Madeyski and Jureczko reported that process metrics like number of developers changing a file have"significant impact.

Predictive performance can be affected significantly by the Datasets. Certain datasets are difficult to learn from than others. One such dataset is PC2 NASA dataset which is specifically difficult to learn from. This was reported by Ktlubay and Menzies too. Hence PC2 remains most rarely used dataset. Ariholm and Briand used very good modelling practices and still reported poor results. This study demonstrates how data used can be significant.

Quality of data should be considered, since normally datasets are noisy and contain outliers and missing values that can skew results. Gray showed that predictive power can be impacted if the data is not cleaned. Jiang acknowledged the importance of data quality.

Features of data have to be considered. Normally related variables bias the prediction of models. It can be improved by feature selection on set of independent variables.

Data balance is significant. As usually the number of non defective units is way more than that that of defective ones. PC2 NASA dataset is the example. In it defective class contains only 0.4% data points. Classifiers are impacted to different extents by imbalanced data. Chawla, Arisholm and Briand, Arisholm reported decision tree performs badly with imbalanced data. Fuzzy based classifiers have been reported by Visa and Ralescu (2004) to perform robustly. Chen (2014) reported Random Forest to perform better with data balancing

but there is a risk of over fitting as reported by Gray (2012). Data balancing shows no impact on technique like Naive Bayes as reported by Rodriguez (2014).

Classifiers basically are mathematical techniques to construct models which have the ability to predict dependent variables. 22 classifiers' were studied and summarised by Lessmen to learn their prediction power. Ensembles, not so frequently, but surely are used for software defect prediction. Majority voting decision strategy is normally used in them. E.g. Misirli combined the use of Naive Bayes, Use of Artifical Neural Networks and Voting Feature Intervals had better predictive performance over the traditional models which use only one classifier. Conflict remain over which study is the best suited for software defect prediction. Bibi reported regression via classification works well. Naive Bayes and Logistic Regression perform best as seen in 19 of the studies. Arisholn predicted that classifier techniques have limited impact on predictive power. Lessmen also showed similar results suggesting no major difference in the performance of top classifiers.

Panichella reported that even though predictive power of different classifiers detects different defects. Panichella proposed CODEP model which works better than individual classifiers. Panichella conducted cross project validation study which differs from within project validation here.

This project builds on Panichella in number of ways. This analyse at module level as compared to class level. Learning algorithm used in both studies is different as Panichella uses regression where probability of being defective is calculated. This project deals with classification as it classifies as defective or not.

Confusion matrix is often used to measure the performance. The matrix reports the properly classified results as compared to their actual classification. Menzies on the other hand used pd & pf to show standard predictive performance.

There is no as such the best way to measure that how good the model is performing. As performance depends on how good the training data is distributed, how model is constructed and how it will be used.[1]

2.2 DATASETS

This project explores defects from 3,681 classes (containing over a million lines of code) from open source JAVA systems. This data is divided into 6 files. The name of the java systems from which this class data is taken are:

- Tomcat
- Synapse
- Ant
- Redaktor
- Ivy

- Jedit

This type of data is known comes under SeaCraft (Software Engineering Datasets Can Really Assist Future Tasks) datasets. This is a research dataset repository specialising in software engineering research datasets. The data has following attributes:

amc	average method complexity	e.g. number of JAVA byte codes
avg, cc	average McCabe	average McCabe's cyclomatic complexity seen in class
ca	afferent couplings	how many other classes use the specific class.
cam	cohesion amongst classes	summation of number of different types of method parameters in every method divided by a multiplication of number of different method parameter types in whole class and number of methods.
cbm	coupling between methods	total number of new/redefined methods to which all the inherited methods are coupled
cbo	coupling between objects	increased when the methods of one class access services of another.
ce	efferent couplings	how many other classes is used by the specific class.
dam	data access	ratio of the number of private (protected) attributes to the total number of attributes
dit	depth of inheritance tree	
ic	inheritance coupling	number of parent classes to which a given class is coupled (includes counts of methods and variables inherited)
lcom	lack of cohesion in methods	number of pairs of methods that do not share a reference to an case variable.
lcom3	another lack of cohesion measure	if m,a are the number of *methods,attributes* in a class number and $\mu(a)$ is the number of methods accessing an attribute, then $lcom3 = ((\frac{1}{a}\sum_j \mu(a,j)) - m)/(1-m)$.
loc	lines of code	
max, cc	maximum McCabe	maximum McCabe's cyclomatic complexity seen in class
mfa	functional abstraction	number of methods inherited by a class plus number of methods accessible by member methods of the class
moa	aggregation	count of the number of data declarations (class fields) whose types are user defined classes
noc	number of children	
npm	number of public methods	
rfc	response for a class	number of methods invoked in response to a message to the object.
wmc	weighted methods per class	

Table 1: Attributes of data used

2.3 Classifiers

2.3.1 Random Forest

It is a collection or ensemble of trees which when given predictor values are capable of producing results. It classifies or associates the independent variables with the category of

dependent variables associated to it. In regression problems estimate given by the dependent variables is the response of the tree. The Random Forest algorithm was developed by Breiman.

Outcome of a Random Forest is estimated by the collection of an arbitrary number of simple trees. Result of classification problems is the most voted class by the ensemble of trees used in the random forest whereas regression problems use the average of the responses of the individual tree. Ensemble of trees when used can show significant improvement in the results.[2]

2.3.2Logistic Regression

One of the classification algorithms is Logistic Regression which is more often than not used for the problems with binary outcomes. Logistic regression can also be thought of as the special case of linear regression where outcome variable is categorical, where log odds are used as the dependent variable.

It is used to separate kinds of data by using linear decision boundary. Number of errors in it can be reduced by gradient descent. But here the focus is on the reduction of log loss function which gives higher values to misclassified points as compared to that of the adequately classified ones.

Logistic regression is used for prediction of output which is binary, as stated above. For example, if a company issuing credit cards thinks of creating a model whether to give a person credit card or not then it will focus on building something which will predict whether he/she will default or not. This is called "Default Propensity Modeling" in banking lingo.[5]

2.3.3 Support Vector Machine

Support Vector Machines work on the very idea of decision planes which states decision boundaries. Members belonging to different classes are separated by using decision planes. An example is shown below. The objects belong either to class red or they belong to the green one in this example. Both the classes are separated by the separating line such that all objects to the right are red and to the left are green. The new object falling to the right will be labelled red and to the left will be labelled green.

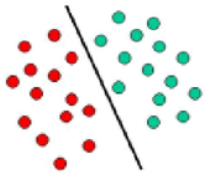


Fig 1

The example which is discussed above is of linear classifier i.e., it seperates the set of objects into red and green classes with a line. Most of the classification tasks we come across are complex than this one and often more complex decision structures are required to classify them properly, This is shown in the image below. AS compared to the last example the complete separation of red and green classes is not possible unless a curve is used. These type of classification tasks which use drawing separating lines are called hyperplanes. These are the tasks for which Support Vector Machines are well suited.

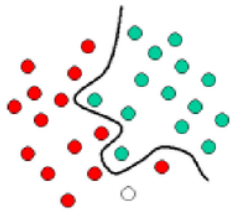


Fig 2

Reasons for choosing these classifiers:

- Previously used.
- Gives the option to compare with previous models.
- These use different techniques and so it is reasonable to detect whether different defects are detected by each and whether the prediction consistency is different among the classifiers.[3]

3. Tools and Methodology

3.1 Tools & Technologies

3.1.1 Technology: 'R'

- The R statistical programming language is a free open source package based on the S language developed by Bell Labs.
- It is an interpreted language.
- It is very powerful for writing programs.
- It has many statistical functions built in.
- It is free.

Fig 3: R Logo

3.1.2 Software: 'R Studio'

It is an IDE which allows the user to run R in more user friendly manner. It was founded by JJ Allaire. IT is available in two editions: RStudio Desktop and RStudio Server. It is written in C++ and for graphical framework it uses Qt framework. Development of it was started at the end of 2010. It's first beta version (v0.92) was released in February, 2011. On 1 November, 2016 Version 1.0 was released.

It has four components:

- File editor
- Console
- Workspace
- Help/Package

It aids users by providing features such as Code completion (it predicts the possible arguments, functions, braces etc.), Command history search(it gives you the liberty to look for previous commands), Command history to R script/file, Function extraction from Rscript etc.[4]

Fig 4: R_Studio Logo

3.2 Modus Operandi of Data Science

Fig 5: Flow of data science project

3.2.1 Data Acquisition

It is the first and the foremost step of any Data Science project. Usually people are not able to find the data at one place because more often than not it is distributed across the line of business. There are numerous ways to acquire data e.g. entering it manually in a spread sheet, downloading data files, streaming data on demand from online sources via APIs, generation of data by computer softwares.

3.2.2 Data Preparation

Data scientists roughly spend 50-80% of their time in gathering and cleaning data i.e trying to get it in the form on which they can perform desired operations upon. What makes it so important to devote such amount of time to this activity is the diverse nature of data that they encounter from their size to formats to ordering.

Data cleaning provides the direction to our research. It is extent to which we cleaned the data which decides how fruitful our algorithms will be. This quite clearly tells that cleaning the

data is as significant to any firm as making good algorithms is. More often than not this turns out to be the point of differentiation between great enterprise data scientist and moderate enterprise data scientist.[5]

According to the survey which was performed on 80 data scientists San Francisco based crwodsourcing and data mining company 'CrowdFlower'

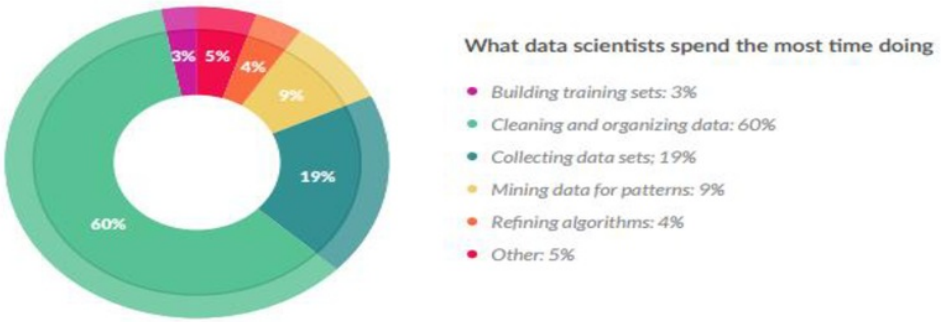


Fig6: Time distribution of Data Scientists

Even though it is the one of the least glamorous tasks but most of the time a data scientist is spent in data cleaning which we also call data munging and data wrangling. The amount of time needed to prepare data (clean it) is dependent on the fact that how healthy data is i.e how complete it is, how many inconsistencies it has, how many missing values are there and many such things.[6]

Data cleaning is essential because given data can have discrepancies in the names or codes, it may also have outliers or errors, to summarize the initial dataset is not qualitative rather it is quantitative.

Steps involved in data preparation are:

- **Data cleaning:**

This step includes dealing with missing and inconsistent values and thereby smoothing out noisy data. The missing values can either be treated by removing them or by filling them with appropriate values depending upon the situation. Noisy data is either is tackled manually or using various regression and clustering techniques.

- **Data integration:**

It involves things like data conflict resolving, handling of redundancies in data if there are any & schema integration

- **Data transformation:**

In this step noise is removed from the data if there is any and it also does normalization, aggregation and generalization.

- **Data reduction:**

This step is done with the motive to reduce the size of data keeping but at the same time maintaining it's utility i.e. making sure that it has the same statistical inference qualities as it used to have. Depending the requirement we can use either of the three approaches: dimensionality reduction, data cube aggregation and numerosity reduction.

- **Data discretization:**

This step is mainly used for the algorithms which only accept categorical attributes. This step helps data scientists to divide continuous data into intervals and thereby also assisting in reducing the size of the data in hand.

3.2.3 EDA: Exploratory Data Analysis

It is the approach to analyze data with the motive to summarize the main characteristics of the data and get vital insights. Mostly graphical methods like scatter plots, histograms, bar graphs, line plots etc. are used but it does have certain quantitative techniques as well. EDA's main motive is to help us:

- Select the appropriate tool and technique that should be used.
- Design hypothesis tests.
- Evaluate the assumptions which will be the basis of our statistical inferences.
- Assess the requirement of data collection that has to be done in future in order to make proper conclusions or may even predict things.

3.2.4 Modeling

In machine learning we make computers learn from experiences. For this purpose we training data to train our models which is called modelling and then use the learning to estimate or classify results. There are various models which can be used depending upon the problem. Few of them are mentioned below:

- **Decision tree**: Each node of the tree reduces the possible options by looking at the trend followed in historic data. Hence, decision tree. E.g. recommendation of clothes on e-commerce site can be made on the basis of the info of the current user and this can pretty well be aided by the 'decision tree'.

- **Random forest**: It is a collection or ensemble of trees which when given predictor values are capable of producing results. It classifies or associates the independent variables with the category of dependent variables associated to it. In regression problems estimate given by the dependent variables is the response of the tree

- **Logistic regression**: It is used to separate kinds of data by using linear decision boundary. Number of errors in it can be reduced by gradient descent. But here the focus is on the reduction of log loss function which gives higher values to misclassified points as compared to that of the adequately classified ones. This is further explained in depth in next chapter.

- **Support Vector Machines**: Out of the two possible regression lines the best is the one which is optimistically away from boundary point of each section. For this now we don't use 'Gradient Descent' rather we use 'linear optimization'. And on the whole this method is called 'Support Vector Machines'. Kernel Trick is used in Support Vector Machines when line is not enough to split. We either use different geometric curves or we can either use different planes for different sets (both of them are same).

3.2.5 Evaluation

There are various methods used by data scientists to measure the accuracy of the model:

- ROC Curve: The plot between false positive rate and True Positive rate.
- Gini coefficient: This is the ratio of area between the ROC curve and the diagonal line & the area of the above triangle
- Cross Validation: In it we split the dataset into two parts one is used to train and the other is used to test. Test on data which is not seen by the model before. This helps in checking how the model will behave with the data it is supposed to chek afterwards.
- Confusion Matrix: A table that shows the count of predictions made for each class as opposed to the count of instances that really are in each class. This helps to know the kind of mistakes made by the algorithm. Accuracy, true positive, false positive, Sensitivity & specificity are shown by this for a given model.

4. Experiment and Results

Fig 9: Flow and actions done throughout the project

4.1 Problem statement

To detect the software defects by building a model and investigate if all the classifiers detect same defects or not and show the results using venn diagrams.

4.2 Data Acquisition

It is the first and the foremost step of any Data Science project. Usually people are not able to find the data at one place because more often than not it is distributed across the line of business. There are numerous ways to acquire data e.g. entering it manually in a spread sheet, downloading data files, streaming data on demand from online sources via APIs, generation of data by computer softwares. Data acquired here is in the spreadsheet format.

4.3 Data Preparation

Data preparing provides the direction to our research. It is extent to which we cleaned the data which decides how fruitful our algorithms will be. This quite clearly tells that cleaning the data is as significant to any firm as making good algorithms is. More often than not this turns out to be the point of differentiation between great enterprise data scientist and moderate enterprise data scientist. Here data provided was already in the useable format and it did not require cleaning.

4.4 EDA (Exploratory Data Analysis)

It is the approach to analyze data with the motive to summarize the main characteristics of the data and get vital insights. Mostly graphical methods like scatter plots, histograms, bar graphs, line plots etc. are used but it does have certain quantitative techniques as well. Below is the screenshot of EDA on 'redaktor' dataset:

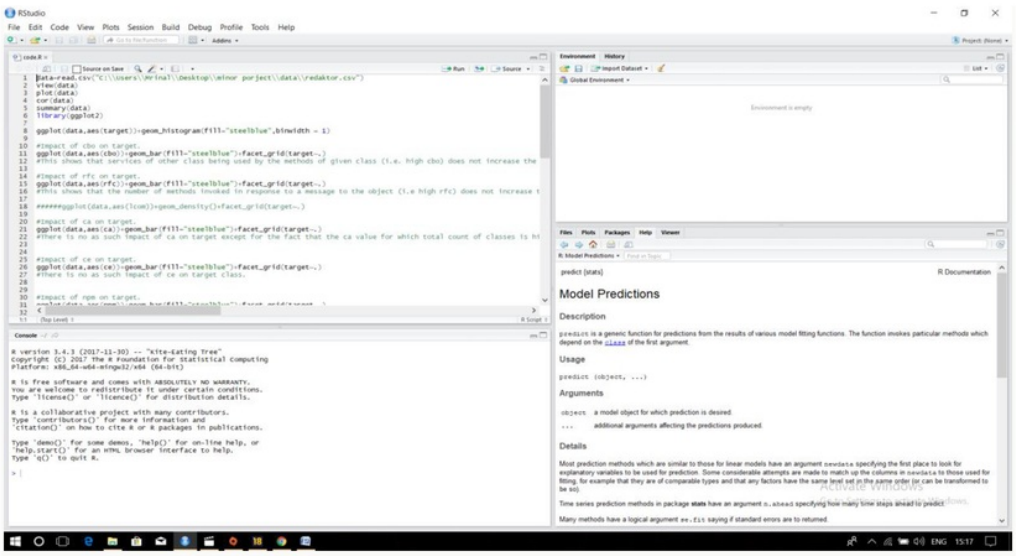


Fig 8: Screenshot of R_Studio showing EDA

4.5 Modelling

In machine learning we make computers learn from experiences. For this purpose we trainingdata to train our models which is called modelling and then use the learning to estimate or classify results. Below is the screenshot of modelling on 'redaktor' dataset:

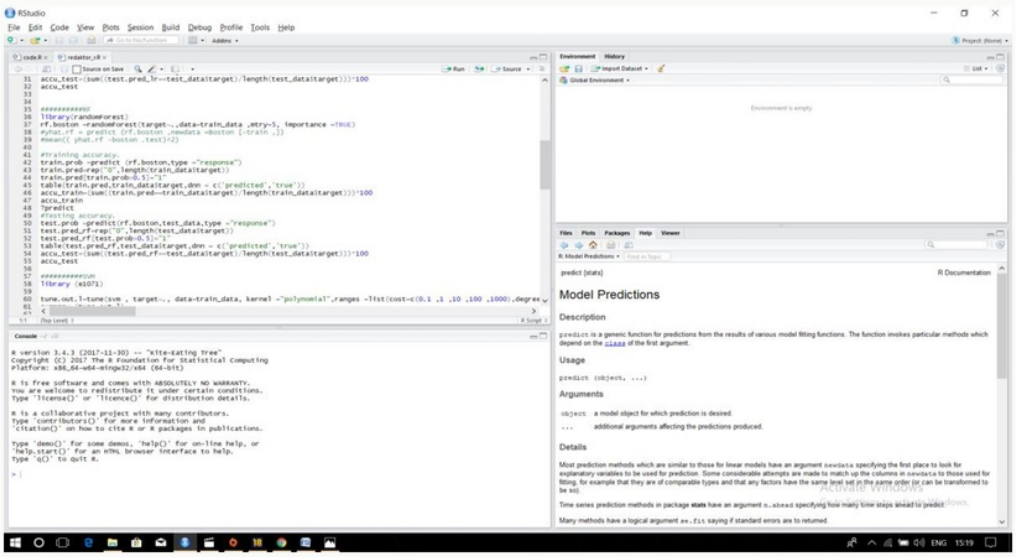


Fig 9: Screenshot of R_Studio showing modelling

4.6 Results

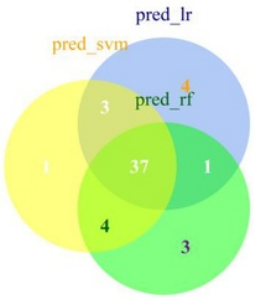
After building the models on every dataset and running them on test data wee emphasised on test accuracy and noted it down in a table as shown below:

	Logistic Regression (in %)	***Random Forest (in %)***	***SVM (in %)***
Ant	***69.57***	***75.36***	***63.77***
Ivy	***91.51***	***90.57***	***91.51***

Jedit	***96.62***	***96.62***	***97.97***
Redaktor	***81.13***	***84.90***	***86.79***
Synapse	***71.42***	***74.02***	***75.32***
Velocity	***79.71***	***79.71***	***75.36***

Table 2: Test accuracy for various classifiers on each dataset

Below are the Venn diagrams on different datasets divided on the basis of target class showing the count of classes detected defective or not by one of the classifiers or two of them or three of them. The Venn diagram on the left is for the non defective class. As expected the count is larger across all the datasets whereas it is less to the right.



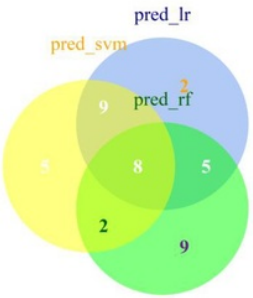
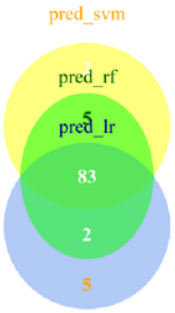


Fig 10: 'ant' dataset when target=0

Fig 11: 'ant' dataset when target=1

Interpretation:

Venn diagram on the left has 37 out of the total predicted as non defective by any of the predictor, predicted as non defective by all three of them whereas 3,4 &1 as non defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as non defective by one of the classifier i.e. 1, 3 & 4 by SVM, RF & LR respectively. The one on the right has 8 out of the total predicted defective by any of the predictor, predicted as non defective by all three of them whereas 9, 2 & 5 as defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as defective by one of the classifier i.e. 5, 9 & 2 by SVM, RF & LR respectively.



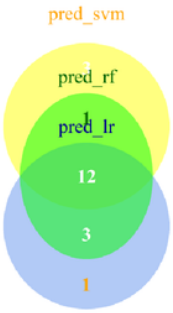


Fig 12: 'ivy' dataset when target=0

Fig 13: ivy' dataset when target=1

Interpretation:

Venn diagram on the left has 83 out of the total predicted as non defective by any of the predictor, predicted as non defective by all three of them whereas 5 & 2 as non defective by the combination of SVM & RF and LR & RF respectively. Rest of the classes are individually detected as non defective by one of the classifier i.e. 2 & 5 by SVM & LR respectively. The one on the right has 8 out of the total predicted defective by any of the predictor, predicted as non defective by all three of them whereas 1 & 3 as defective by the combination of SVM & RF and LR & RF respectively. Rest of the classes are individually detected as defective by one of the classifier i.e. 3 & 1 by SVM & LR respectively.

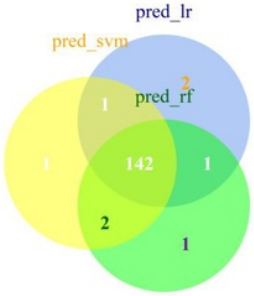
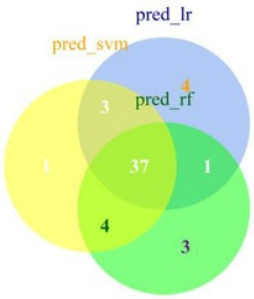


Fig 14: 'jedit' dataset when target=0

Fig 15: 'jedit' dataset when target=1

Interpretation:

Venn diagram on the left has 142 out of the total predicted as non defective by any of the predictor, predicted as non defective by all three of them whereas 1, 2 & 1 as non defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as non defective by one of the classifier i.e. 1, 1 & 2 by SVM, RF & LR respectively. The one on the right has only 2 elements and are bothe detected by LR and SVM.



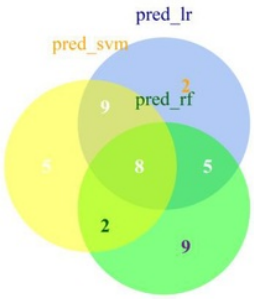
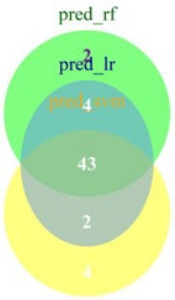


Fig 16: 'redaktor' dataset when target=0

Fig 17: 'redaktor' dataset when target=1

Interpretation:

Venn diagram on the left has 37 out of the total predicted as non defective by any of the predictor, predicted as non defective by all three of them whereas 3, 4 &1 as non defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as non defective by one of the classifier i.e. 1, 3 & 4 by SVM, RF & LR respectively. The one on the right has 8 out of the total predicted defective by any of the predictor, predicted as non defective by all three of them whereas 9, 2 & 5 as defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as defective by one of the classifier i.e. 5, 9 & 2 by SVM, RF & LR respectively.



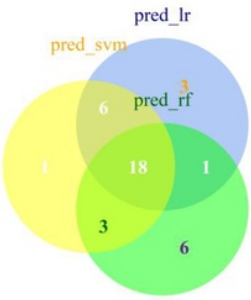
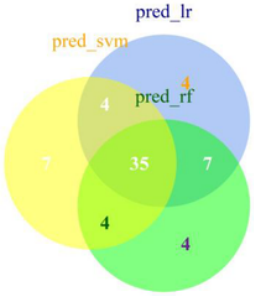


Fig 18: 'synapse' dataset when target=0

Fig 19: 'synapse' dataset when target=1

Interpretation:

Venn diagram on the left has 43 out of the total predicted as non defective by any of the predictor, predicted as non defective by all three of them whereas 4 & 2 as non defective by the combination of SVM & RF and LR & RF respectively. Rest of the classes are individually detected as non defective by one of the classifier i.e. 2 & 4 by SVM & LR respectively. The one on the right has 18 out of the total predicted defective by any of the predictor, predicted as non defective by all three of them whereas 6, 3 & 1 as defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as defective by one of the classifier i.e. 1, 6 & 3 by SVM, RF & LR respectively.



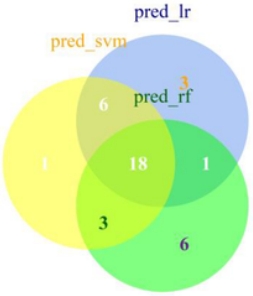


Fig 20: 'velocity' dataset when target=0

Fig 21: 'velocity' dataset when target=1

Interpretation:

Venn diagram on the left has 35 out of the total predicted as non defective by any of the predictor, predicted as non defective by all three of them whereas 4, 4 & 7 as non defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as non defective by one of the classifier i.e. 7, 4 & 4 by SVM, RF & LR respectively. The one on the right has 18 out of the total predicted defective by any of the predictor, predicted as non defective by all three of them whereas 6, 3 & 1 as defective by the combination of SVM & LR, RF & SVM and LR & RF respectively. Rest of the classes are individually detected as defective by one of the classifier i.e. 1, 6 & 3 by SVM, RF & LR respectively.

The reason for the sudden dip in performance of classifiers when applied on certain datasets and the split of the count detected by all three of them, any 2 of them and anyone exclusively is out of the scope of this project.

5. Conclusion and Future Prospects

We see that the test accuracy for a particular dataset is almost same for each and every classifier used, with slight difference in performance across them. Such as in the case of 'ant' Random Forest performs the best whereas SVM the worst. Same goes in the case of 'velocity' where Logistic Regression and Random Forest perform the same but performs significantly better than SVM.

Results also show that predictive performance of classifiers is dependent on the quality of the data used. As it can be seen that it is high in case of 'ivy' & 'jedit' whereas it is moderate in case of 'velocity' and 'redaktor'but it is poor in case of 'ant' and 'synapse'.

In accordance to the study done by Bowes, Hall and Petric our research also suggest that even though the predictive performance of various classifiers is similar but there is a difference between the defects detected by various classifiers, which is only obvious as the three classifiers approach classification using very different techniques. So to conclude that each classifier performs similar is wrong as they differ in the individual defects being detected. This study quite clearly suggests that ensemble of heterogeneous classifiers will perform best.

Future scope:

- Out of all the possible additions the main would be to consider flipping in the future work and also to look for the quality in datasets due to which the classifiers perform badly on them.
- Future work also includes implementing ensemble of classifiers.
- To look for the pattern in defects detected by a certain classifier is one of the important things which have not been investigated by anyone yet.
- Also to look for the set of classifiers which perform the best and are the most consistent.
- Future work also includes to investigate whether there is some global ensemble which perform the best or does it remain local to the dataset.
- Stacking is also the option which can be used. It doesn't use majority voting system rather it uses another classifier to make the final prediction.

So basically it all comes down to that we used the selected models for defect prediction on each dataset and found that the predictive performance of the datasets is not so different but the defects detected by each classifier is different i.e. a software which is labelled as defective by one classifier may or may not be labelled as defective by the other two. Hence by looking at these observations the best possible way to improve the software defect prediction accuracy and thereby reducing the cost occurred due to defective software being operated is by using the ensemble of classifiers which in classification problems normally use majority voting technique.

6. References

[1]David Bowes1, Tracy Hall, Jean Petric(2017). Software defect prediction: do different classifiers find the same defects? The Author(s) 2017. This article is published with open access at Springerlink.com

[2]Random Forests, http://www.statsoft.com/Textbook/Random-Forest

[3]What is logistic regression?, Sandeep Dayananda https://www.quora.com/What-is-logistic-regression

[4]Over 16 years of R project history, David Smith, http://blog.revolutionanalytics.com/2016/03/16-years-of-r-history.html

[5]Why data preparation is an important part of data science?, https://www.dezyre.com/article/why-data-preparation-is-an-important-part-of-data-science/242

[6]2016 Data Science Report, http://visit.crowdflower.com/rs/416-ZBE-142/images/CrowdFlower_DataScienceReport_2016.pdf

mrinal

ORIGINALITY REPORT

5%	4%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.dezyre.com Internet Source	1%
2	simplified-analytics.blogspot.co.uk Internet Source	1%
3	Beecham, S, Bowes, D, Counsell, S, Gray, D and Hall, T. "A systematic review of fault prediction performance in software engineering", Brunel University Research Archive (BURA), 2011. Publication	1%
4	www.people.vcu.edu Internet Source	1%
5	Submitted to Humboldt-Universitat zu Berlin Student Paper	<1%
6	www.analyticsvidhya.com Internet Source	<1%
7	S. Bibi. "BBN based approach for improving the software development process of an SME-a case study", Journal of Software Maintenance	<1%

and Evolution Research and Practice, 03/2010
Publication

8	digbib.ubka.uni-karlsruhe.de Internet Source	<1%
9	upcommons.upc.edu Internet Source	<1%

Exclude quotes	On	Exclude matches	< 8 words
Exclude bibliography	On		