**Software quality prediction using Ensemble Methods in software projects**

Report submitted in fulfilment of the requirements

for the Exploratory Project of

Second Year B.Tech. Second Year IDD

**Sanchit Tyagi** **Shikhar Sharma**

18075053 18074016

Under the guidance of

**Dr. Amrita Chaturvedi**



Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI

Varanasi 221005, India

June 2020

Dedicated to

My parents, professors, exploratory project convenor, mentor and everyone who helped and motivated us in successful completion of this report.

Declaration

I certify that-

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.
2. The work has not been submitted for any project.
3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi Sanchit Tyagi Shikhar Sharma

Date: 18/06/2020 B.Tech. IDD

Department of Computer Science and Engineering,

Indian Institute of Technology (BHU) Varanasi,

Varanasi, INDIA 221005.

Certificate

This is to certify that the work contained in this report entitled “Software quality prediction using Ensemble Methods in software projects" being submitted by Sanchit Tyagi (Roll No. - 18075053), and Shikhar Sharma (Roll No. – 18074016), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of my supervision.

Dr. Amrita Chaturvedi

Place: IIT (BHU) Varanasi Department of Computer Science and Engineering,

Date: 18/06/2020 Indian Institute of Technology (BHU) Varanasi,

Varanasi, INDIA 221005.

Acknowledgments

We want to express our sincere gratitude to the people who have helped us the most throughout our project. We are grateful to our project supervisor **Dr. Amrita Chaturvedi** for providing us an opportunity to implement the paper entitled “**Software Quality Prediction using Ensemble Methods in Software Projects**” and her constant support for the project. We wish to thank our parents for their support and attention. We would like to thank our friends who encouraged us and helped us out in finalizing the project. At last, we also thank the Almighty for his blessings.

Place: IIT (BHU) Varanasi Shikhar Sharma

Date: 18/06/2020 Sanchit Tyagi

Abstract

Value of quality in software is well recognised. Poor quality can incur a lot of cost during various phases in software cycle and can bring disaster to a business. Open source software can be used as a valid test case for the assessment of software characteristics.

Moreover, the powerful technique of data mining can be used to extract the characteristics of software and deal with possible bugs. A proper software testing can unveil a wide range of malfunctions that might happen to a software. Many tools have emerged to monitor the behaviour of a software. As more and more large and complex software are coming into picture, striving for the highest quality software without any defects is more important now than ever before.

Data Mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analysing data patterns in large batches of data using one or more software. Some of the common and effective techniques of data mining include Support Vector Machine (SVM), Random Forest (RF), Boosting and Naïve Bayes (NB). The output of these methods could help in detecting violation of patterns and find the bugs. This powerful way of Data Mining therefore perform the essential function of Software testing and maintenance.

vi

Contents

List of Figures viii

List of Tables ix

1) Introduction **1**

1.1 Overview 1

1.2 Motivation of the Research Work 1

2) Project Work **2**

2.1 About Datasets 2

2.2 Feature Selection 3

2.3 Classification Models 3

2.4 Workflow 4

2.5 Hypothesis Testing 5

3) Initial Assessment **7**

4) Conclusions **13**

5) References **14**

vii

List of Figures

1. Fig. 1 – Accuracy of NASA Dataset . . . . . . . . . . . . . . . . . . . . . . 7
2. Fig. 2 – Accuracy of Eclipse Datasets . . . . . . . . . . . . . . . . . . . . . 7
3. Fig. 3 – Accuracy of Android Datasets . . . . . . . . . . . . . . . . . . . . 7
4. Fig. 4 – Accuracy of Elastic Search Datasets . . . . . . . . . . . . . . . 7

viii

List of Tables

1. Table 1 – About Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
2. Table 2 – About Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
3. Table 3 – Overall P-value test . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
4. Table 4 – Naïve Bayes Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
5. Table 5 – Support Vector Machine Results . . . . . . . . . . . . . . . . . . . . . 9
6. Table 6 – K-Nearest Neighbours Results . . . . . . . . . . . . . . . . . . . . . . 9
7. Table 7 – Random Forest Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
8. Table 8 – AdaBoost Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
9. Table 9 – Majority Voting Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
10. Table 10 – Overall F-Score result . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

ix

1) Introduction

* 1. Overview

In this paper our aim is to develop an effective model which can be used for quality software predictions and bug detection. We will use various data mining techniques to achieve this. We have worked on those software metrics on which some papers have been published previously and we try to provide more efficient methods to produce better quality. We have collected these datasets from open source repositories, did processing of the metrics and provided the assessment using statistical methods and other data mining techniques.

There have already been discussions about practical aspect of software analysis. This paper aims at taking a deeper dive into the techniques of Data Mining and providing better results than those being already published. The most essential aspect our structured methodology is the powerful feature reduction techniques and other pre-processing techniques that have been applied before applying the actual techniques.

1.2 Motivation of the Research Work

In the previous studies done in this field most of them focussed on the NASA dataset[2] only. What we have done is we have included some other major datasets as a part of our study, namely Eclipse[3], Android[1] and Elastic[4] projects. Some authors had also included these datasets in their study but the main difference is our focus was on applying the data Mining techniques which focussed both on cleaning /pre-processing of data as well as the main classification techniques. The results we have been obtained beats the previous results because of these specific steps. The feature reduction techniques like Principal Component Analysis (PCA) have helped us in achieving some great results.

1

2) Project Work

2.1 About Datasets

We conducted research in the field of data mining to find datasets on which papers have already been published. Also, we used some other major datasets from open source repositories.

The datasets used are-

* NASA Dataset
* Eclipse Dataset
* Elastic-Search Dataset
* Android Dataset

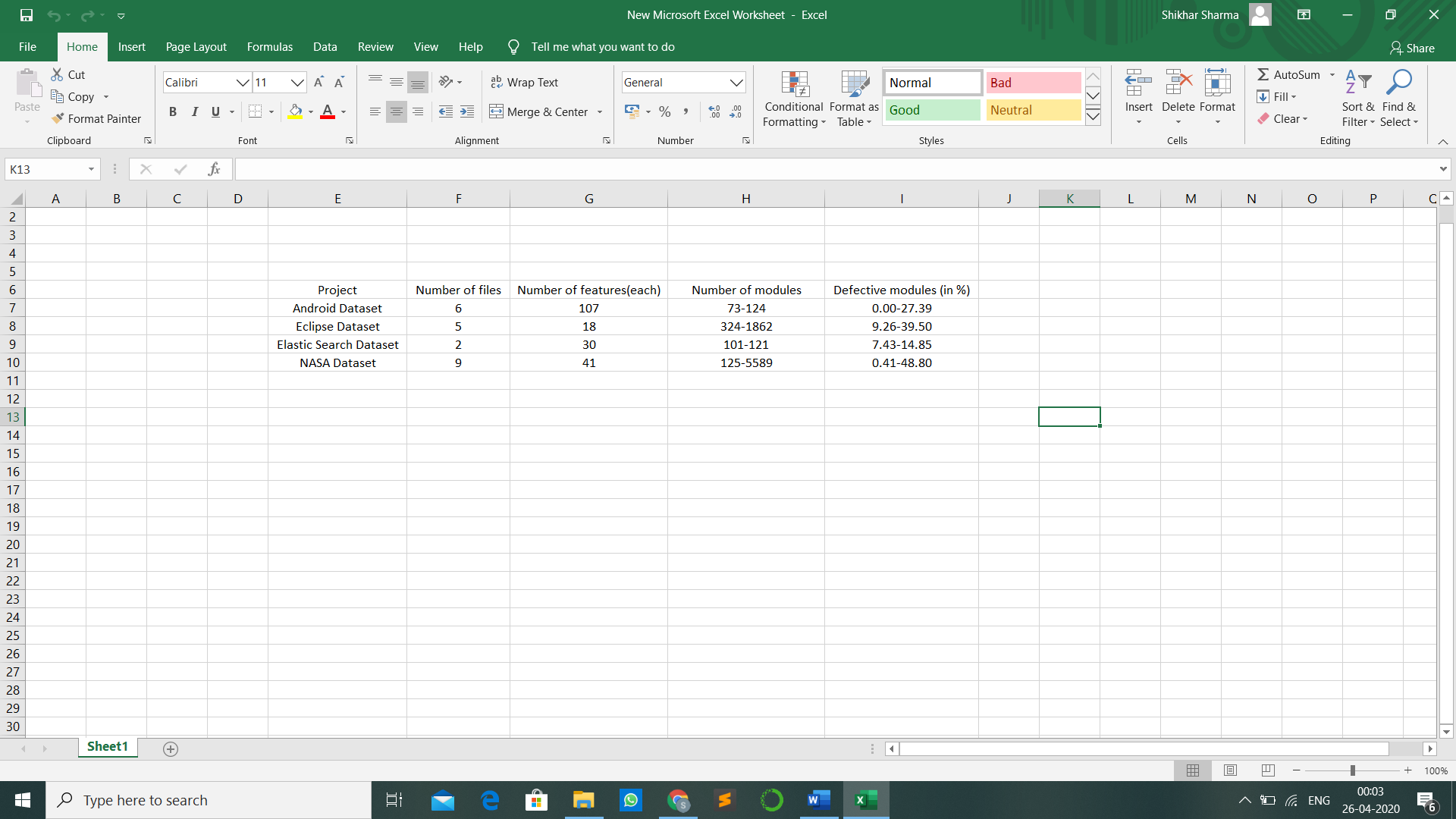


Table 1: Datasets

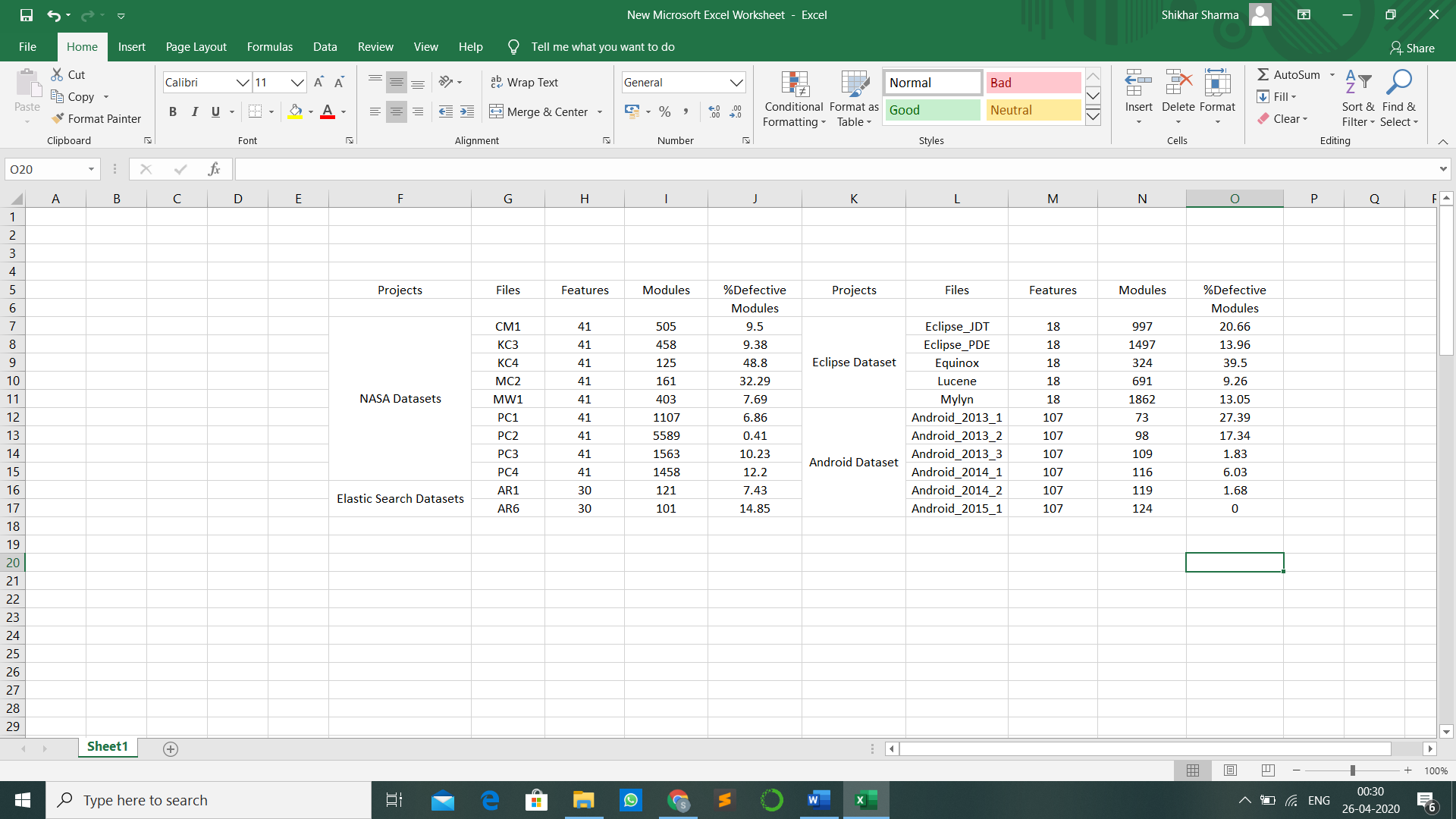


Table 2: Datasets

The details about the datasets are given in Table-1 and Table-2.

2

* 1. Feature Selection

Feature Selection is a very critical component in a Data Scientist’s workflow. When presented data with very high dimensionality, models usually choke because **training time** increases exponentially with number of features and models have increasing risk of **overfitting**with increasing number of features. Feature Selection methods helps with these problems by reducing the dimensions without much loss of the total information. It also helps to make sense of the features and its importance. We used the following data mining techniques on the defective modules before running the on the above models: -

* **SelectKBest**

The SelectKBest class just scores the features using a function (in our case chi2 but could be others) and then "removes all but the k highest scoring features".

* **Principal Component Analysis**

Principal component analysis is a technique for feature extraction - so it combines our input variables in a specific way, then we can drop the “least important” variables while still retaining the most valuable parts of all of the variables.

* **Variance**

We calculate the variance of data for every feature. Those which have less variance do not affect the result much since the data in those columns in somewhat similar. Hence the required number of features are selected starting with the ones having more variance.

2.3 Classification Models

We used the following most commonly used yet most effective machine learning models: -

* **Naïve Bayes**

A Naive Bayes Classifier is a supervised machine-learning algorithm that uses the Bayes’ Theorem, which assumes that features are statistically independent. The theorem relies on the naive assumption that input variables are independent of each other.

* **Support Vector Machine**

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

* **K-Nearest Neighbours**

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its *k* nearest neighbours.

3

* **Ensemble Classifier**

Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a weighted vote of their predictions. It generates various base classifiers from which a new classifier is derived which performs better than any constituent classifier.

**Sequential** **Parallel**

Dependence between base Independence between base

learners. learners.

Known as Boosting methods Known as Bagging methods

1. ***Random Forest***(Parallel Classifier)

Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

1. ***Adaboost***(Sequential Classifier)

AdaBoost assigns a “weight” to each training example. Examples with higher weights are more likely to be included in the training set, and vice versa. After training a classifier, AdaBoost increases the weight on the misclassified examples so that these examples will make up a larger part of the next classifiers training set, and hopefully the next classifier trained will perform better on them.

1. ***Majority Voting***

Every model makes a prediction (votes) for each test instance and the final output prediction is the one that receives more than half of the votes.

* 1. Workflow

After obtaining our datasets from their repositories, we have performed the necessary cleaning action on the them. There were some features in our datasets which had some missing values. It was necessary to deal with these missing values before doing anything. So, one efficient way we found out was to replace all the missing values in a column by the mean of all other values in that column.

The datasets were huge in size, and there some column which had all values in it exactly same. These columns were therefore removed without affecting our results.

Detecting the outliers is our next step. We have used the method of IQR (Interquartile Range). The first and the third quartile (*Q1, Q3*) are calculated. An outlier is then a data point xi that lies outside the interquartile range. These outliers are then removed. There is no definite number of rows to be removed, so we started by removing one row, the one which is farthest away from the interquartile range, and applied the three feature selection techniques on it independently.

4

Now, we continued deleting rows in this order and updating the results for different feature selection techniques after each step. To ensure that there is not a great loss of data, we have set an upper limit as to the number of removed rows.

We have done some initial pre-processing of our data. Now comes the step which has helped us in achieving great results: The feature reduction techniques and the way we have applied them.

First feature reduction method is SelectKBest (SKB). We have given score to each feature based on chi2 function. Now at this point we were not sure that how many columns should we select to obtain best results and without loss of much data. So, we started with removing one column and found out the result for each classifier. Then one by one we removed the columns with least score and simultaneously updated the results for each classifier. To avoid data loss, we have set a maximum limit for number of column removals.

Next, we have PCA and VAR techniques. The Principal Component Analysis (PCA) was the advanced feature reduction method which helped us in achieving results beyond our expectation. It combines our input variables in a specific way, then we can drop the least important variables and retain the important ones. Again, we have done it by dropping one by one and updating to get the best results. In the VAR technique, the statistical concept of variance was applied here to find the distribution of values. It finds the variance for each feature and the one with least variance contributes the least.

Standardisation is a technique wherein we convert the values of each feature such that mean is zero and standard deviation is 1. At every point after removing the columns we have standardised the data.

We now have obtained the results for best accuracy values for each classifier from each feature reduction technique and we pick the best of all techniques to get the final accuracy.

* 1. Hypothesis Testing

**Hypothesis testing** is type of statistical analysis in which we test an assumption regarding a population parameter. The method depends on the nature of the data used and the reason for the analysis. It is used to check the possibility of a hypothesis (assumption that we make about the population parameter) being true by using sample data.

A **hypothesis test**evaluates two mutually exclusive statements (Null Hypothesis and Alternate Hypothesis) about a population to determine which statement is best supported by the sample data. The **Null Hypothesis** is usually a hypothesis of equality between population parameters while the **Alternative Hypothesis** is the opposite of a null hypothesis.

The **P value**, or calculated probability, is the probability of finding the observed results when the null hypothesis is true. If our P value is less than the chosen significance level (generally 5%) then we reject the null hypothesis i.e. our sample gives reasonable evidence to support the alternative hypothesis.

5

**2-sample t-test** compares the means of two independent groups in order to determine whether there is statistical evidence that the associated population means are significantly different.

Our Null hypothesis is that all the classifiers are working same on the datasets. We tested this hypothesis independently on all four projects and then on the combination of all projects treated as one.

For every case, we took the result of each classifier for all the datasets in that case and applied 2-sample t-test for each pair of classifiers. The value of t-test for a classifier with itself was 1, since the mean difference is zero. The classifiers which had less p-values with most of the other classifiers were Random Forest and Majority Voting.

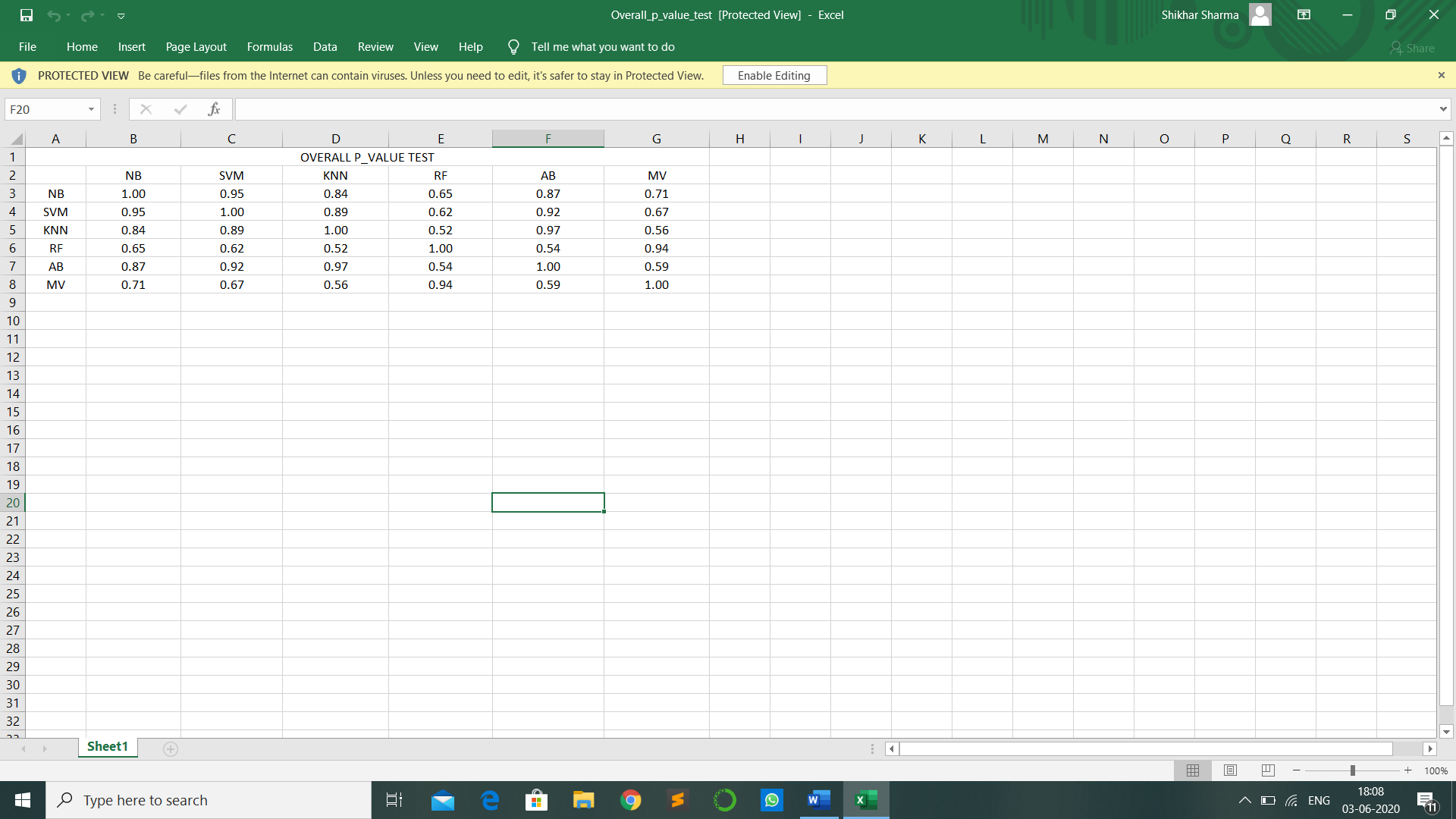


Table 3: Overall P-value Test

From our analysis (Table 3) we can conclude that Random Forest and Majority Voting provided the extraordinary results on every dataset, which were different and better than other classifiers, which provided almost the same results.

6

3) Initial Assessment

We have used accuracy and F-score as the 2 performance indicators for our model.

The bar graphs show the average accuracy that has been calculated for our datasets after all the cleaning and pre-processing. The classification techniques used are Naïve Bayes, Support Vector Machine, K-Nearest Neighbour, Random Forest, Adaboost and Majority Voting.

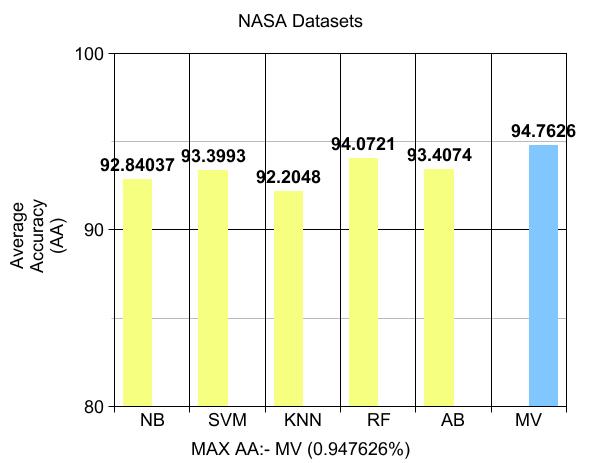
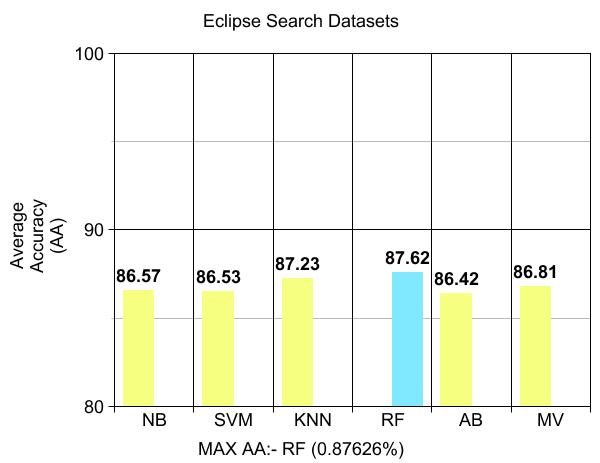
 

Fig 1: Accuracy of NASA Datasets Fig 2: Accuracy of Eclipse Datasets

Figures 1 and 2 show the accuracy obtained by exploiting the NASA datasets and the Eclipse datasets. Majority Voting are the techniques that have worked best for the NASA dataset. Random Forest has given best results for Eclipse projects.

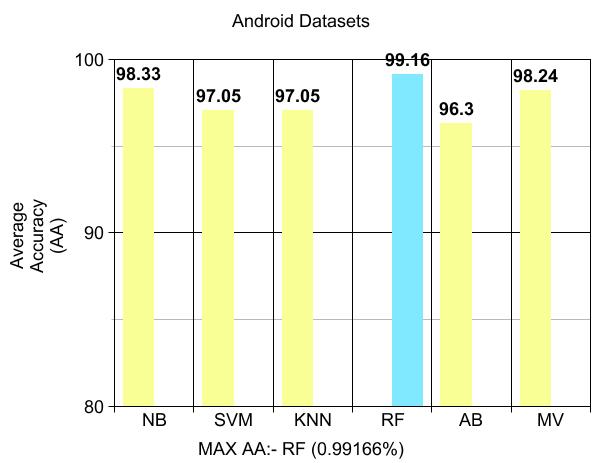
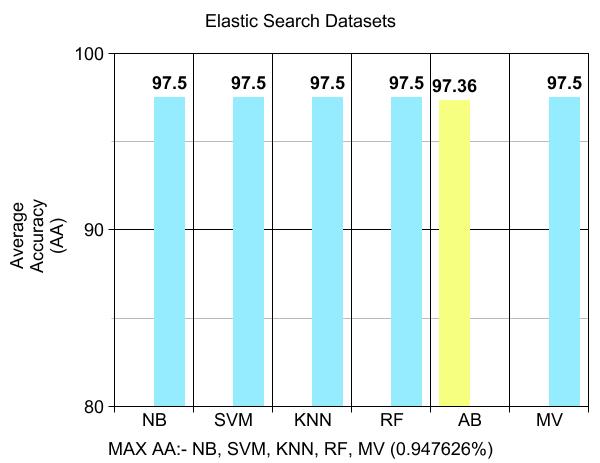
 

Fig 3: Accuracy of Android Datasets Fig 4: Accuracy of Elastic Search Datasets

7

Figures 3 and 4 show the accuracy results for Android and Elastic Projects. The techniques that have performed best are Random Forest and Majority Voting. We see that in general the best technique has been Majority Voting.

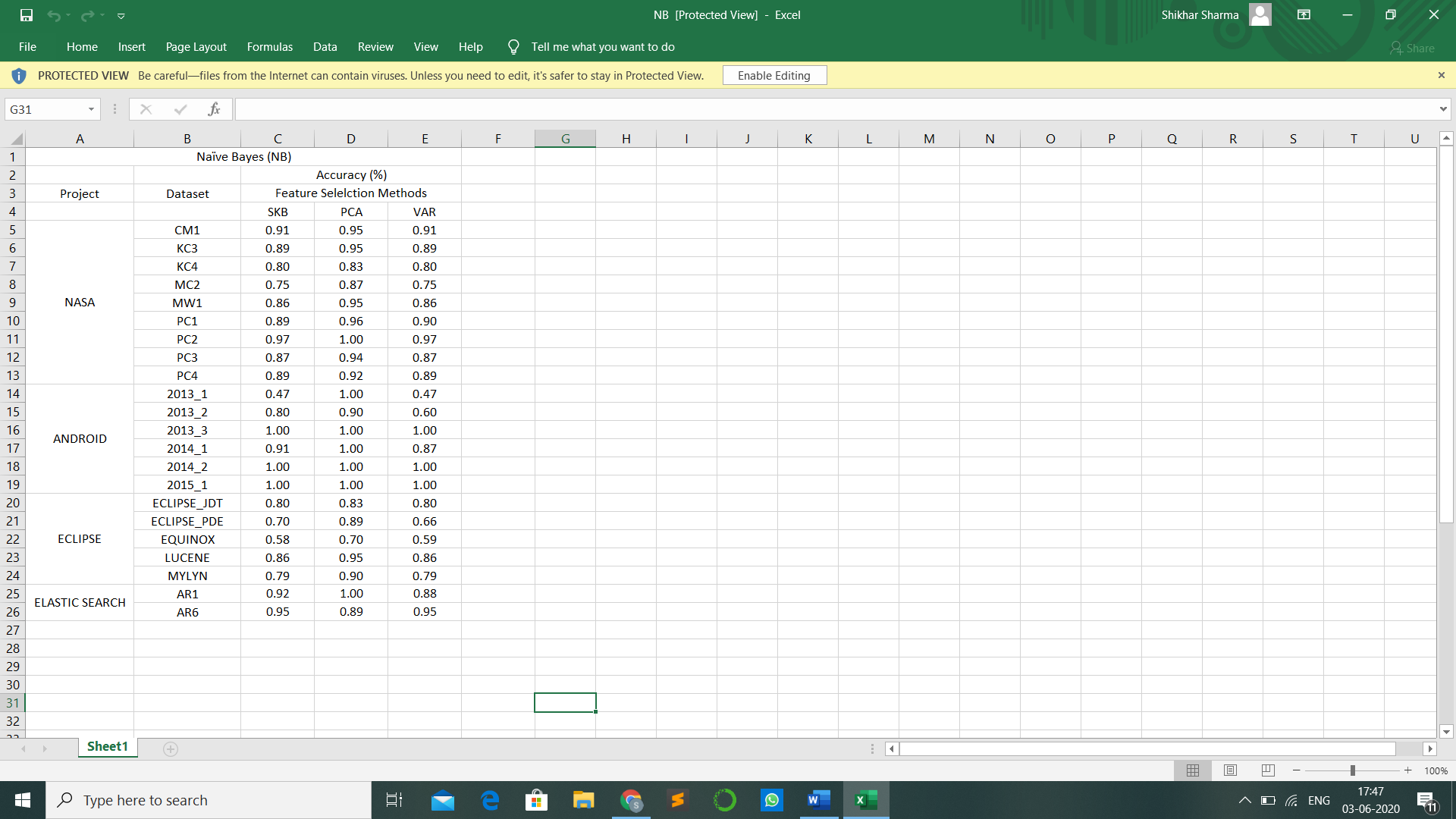


Table 4: Naïve Bayes Results

8

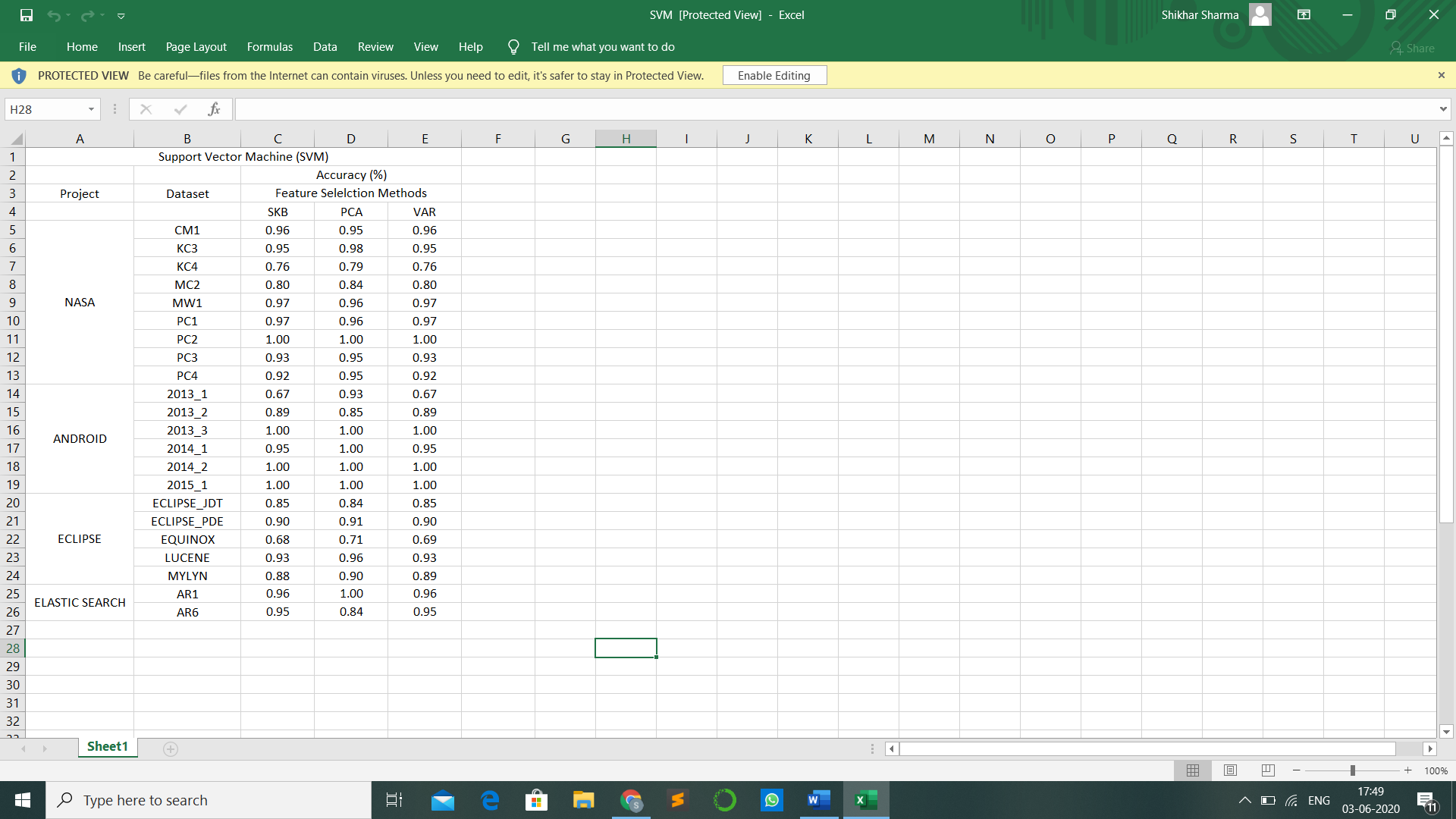


Table 5: Support Vector Machine Results

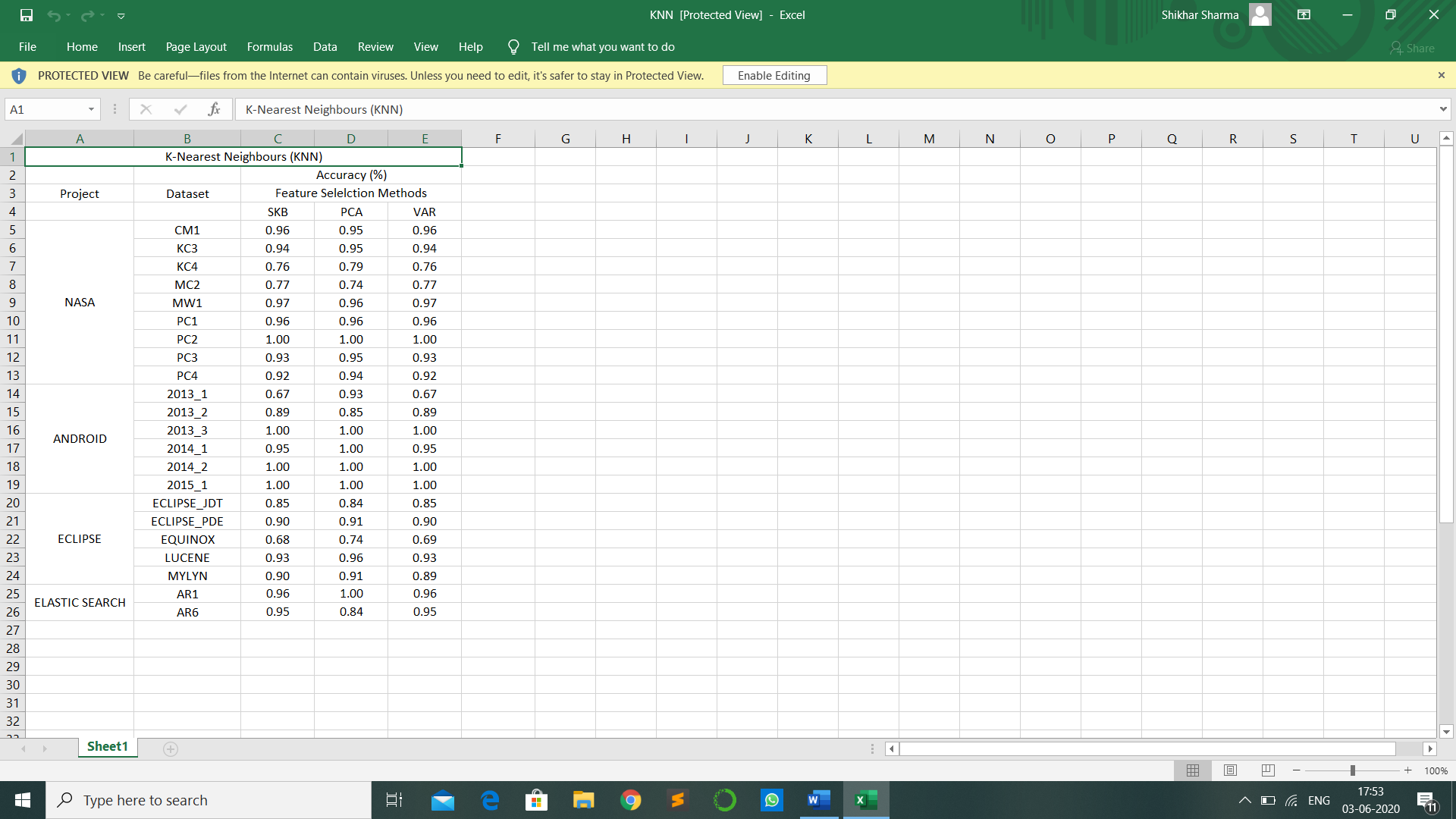


Table 6: K-Nearest Neighbours Results

9

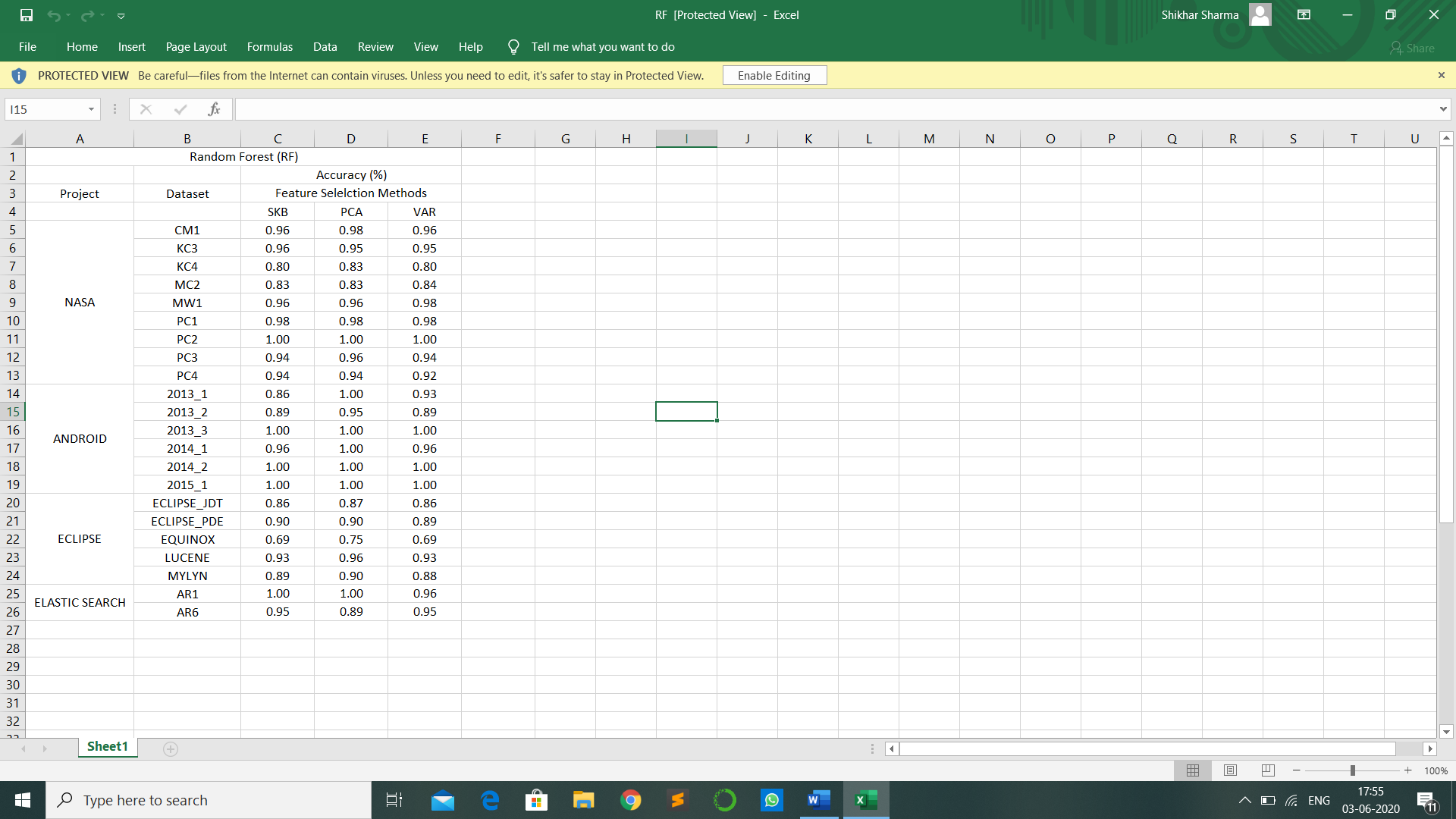


Table 7: Random Forest Results

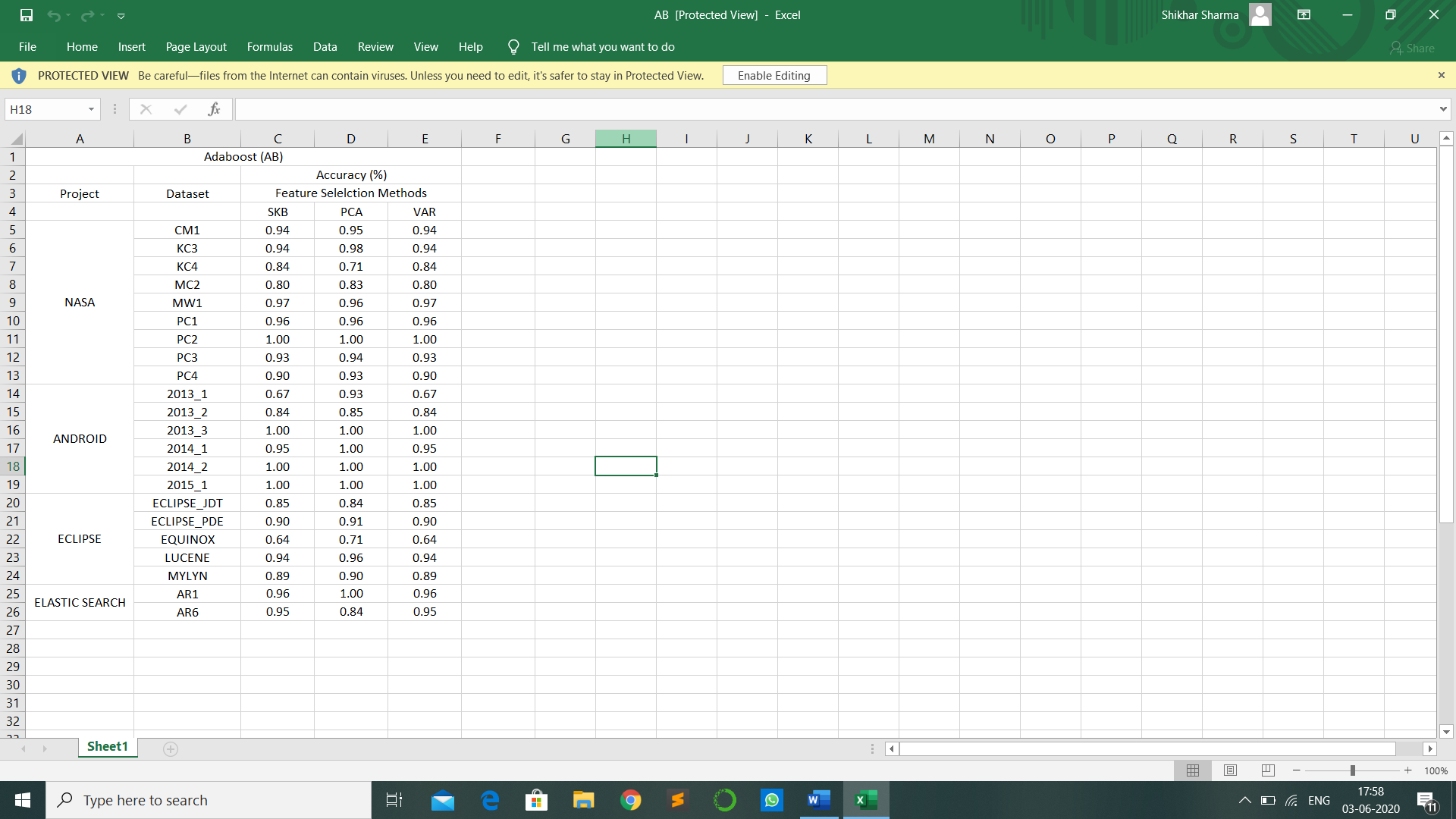


Table 8: Adaboost Results

10

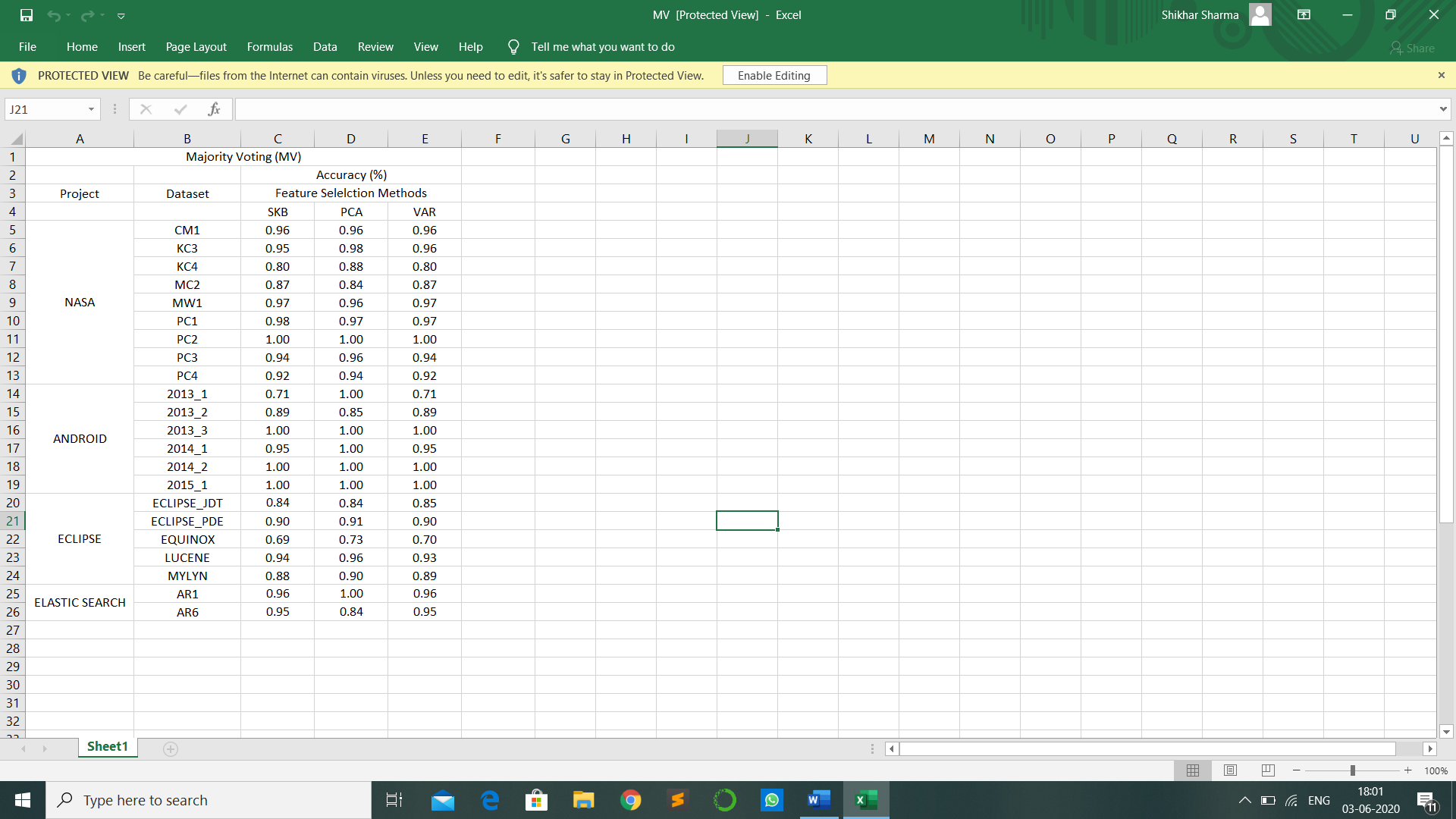


Table 9: Majority Voting Results

Table-4 to table Table-9 shows the **accuracy** results for all 6 classifiers on our datasets including all feature selection methods.

**F-score**- It is a performance metric which includes both precision and recall. F-score is defined as harmonic mean of precision and recall i.e.

F-score= 2\*(Recall\*Precision)/(Recall+Precision).

11

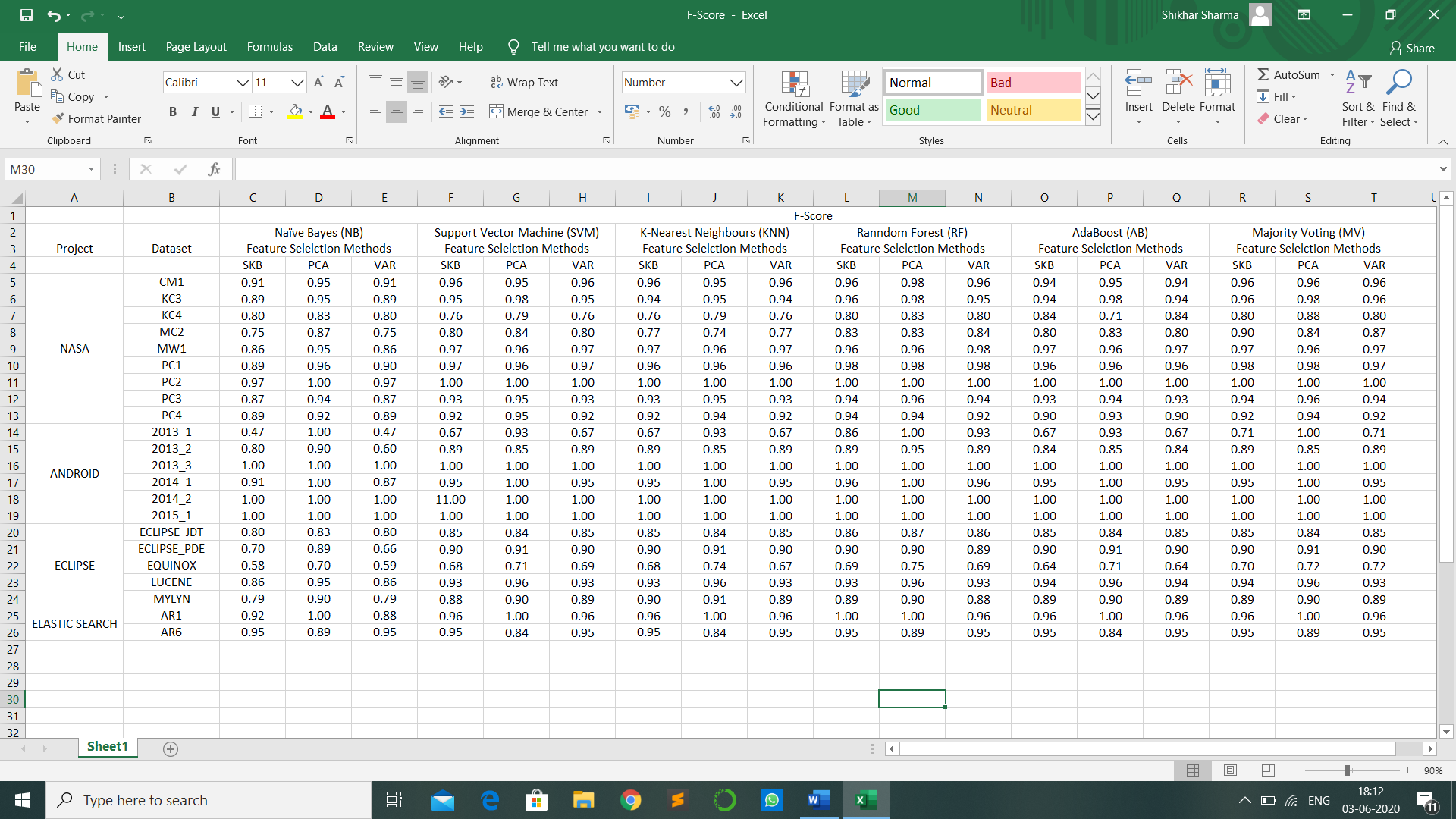


Table 10: Overall F-score Results

Table-10 gives the combined results of F-score for all classifiers.

12

4) Conclusions

In our study we have tried to find a way to detect bugs in software and improve software quality by applying some techniques on some open source projects. We have used currently existing techniques and performance criteria to come to a conclusion. Unlike the previous studies, we have widened our field of study and applied more concrete methods which have yielded better results.

We conclude from our work that Random Forest and Majority Voting have performed the best results over all datasets. We have also tried to show how data mining can constitute a great deal in predicting software quality.

Future Scope

* Other ensemble classifiers can be made which would yield better accuracy than those classifiers which are known today, just like majority voting was a future scope for researches in the past.
* In the future, better feature selection methods could be found which would select the features most optimally without needing to look at each and every case.
* One can use the depth of Neural Network and Deep Learning to improve some aspects of this research.

13

5) References

1. *Android Dataset*. (n.d.). Retrieved from, http://www.inf.uszeged.hu/~ferenc/ papers/UnifiedBugDataSet/
2. *Eclipse Dataset*. (n.d.). Retrieved from, http://bug.inf.usi.ch/download.php
3. *Elastic Search Dataset*. (n.d.). Retrieved from http://www.inf.uszeged.hu/ ~ferenc/papers/UnifiedBugDataSet/
4. *NASA Dataset*. (n.d.). Retrieved from, https://github.com/klainfo/ NASADefectDataset
5. Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2825--2830.
6. Shaikh, R. (2018, October 28). *towards data science*. Retrieved from Feature Selection Techniques in Machine Learning with Python: https:// towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e
7. Sharma, N. (2018, May 22). *towards data science*. Retrieved from Ways to Detect and Remove the Outliers: https://towardsdatascience.com/ways-to-detect-and-remove-the-outliers-404d16608dba

14