B. *Training and Testing Phase*

In the WEKA Classification section, data belonging to our previous WMC and CBO based classification were used as training data and machine learning was performed. Afterwards, RFC, NOC, NIM, Ratio C / C metrics were added to CBO and WMC and the classification was made by Navie Bayes method and the results are shown in Fig. 3.

Projects are correctly classified by 90% accuracy by WEKA. In terms of maintainability, two projects that were previously marked as successful were considered unsuccessful and 2 projects that were previously unsuccessful were classified as successful.

```
Correctly Classified Instances          36               90      %
Incorrectly Classified Instances         4               10      %
Kappa statistic                          0.7721
Mean absolute error                      0.1352
Root mean squared error                  0.288
Relative absolute error                 30.6223 %
Root relative squared error             61.4892 %
Total Number of Instances               40

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,846    0,074    0,846      0,846    0,846      0,772    0,943     0,872     0
                 0,926    0,154    0,926      0,926    0,926      0,772    0,943     0,975     1
Weighted Avg.    0,900    0,128    0,900      0,900    0,900      0,772    0,943     0,942

=== Confusion Matrix ===

  a  b   <-- classified as
 11  2 |  a = 0
  2 25 |  b = 1
```

Fig. 3. Classification with Naive Bayes

V. Conclusion

The key contribution of this study is the investigation of the software metrics that are required to measure maintainability characteristic. After researching scientific literature for the recommended metrics to measure the maintainability characteristic, 40 object-oriented open source projects examined with the aid of machine learning tools.

It is seen that WEKA machine learning software has the correct classification with 90% accuracy. Having two projects that are successful and misclassified is considered an acceptable result as false positives. Having small number of true negatives does not hinder our judgement about the project. If general accuracy is high, common procedure of reviewing the negatives would only result in diminishing risks rather than the waste of time.

On the other hand, false positives might affect the projects at hand by misguiding the review process thus, increasing risk. This misclassification should remain minimal with increasing number of projects.

As a result, with 90% accuracy score of classification, it is observed that the usage of CBO, WMC, RFC, NOC, NIM and Ratio C/C metrics in the measurement of maintainability characteristic of ISO 9126 Quality Standard gives consistent results.

As a future work, we are planning to increase the number of projects and a more comprehensive machine learning study can be carried out and it can be discussed whether or not to add or remove existing metrics.

References

[1] National Institue of Standards and Technology, 'The Economic impacts of inadequate infrastructure for software testing', 2002. [Online]. Available: https://www.nist.gov/sites/default/files/documents/director/planning/report02-3.pdf. [Accessed: 10- Oct- 2018].

[2] Tricentis.com, 'software fail watch 5th edition', 2017. [Online]. Available: https://www.tricentis.com/wp-content/uploads/2018/01/20180119_Software-Fails-Watch_Small_Web.pdf. [Accessed: 10- Oct- 2018].

[3] Jung H., Kim S., and Chung C., "Measuring software product quality: a survey of ISO/IEC 9126", September/October 2004, IEEE Software, p:88-92.

[4] Y. Fitrisia and B. Hendradjaya, "Implementation of ISO 9126-1 quality model for asset inventory information system by utilizing object oriented metrics," 2014 International Conference on Electrical Engineering and Computer Science (ICEECS), Kuta, 2014, pp. 229-234.

[5] U. L. Kulkarni, Y. R. Kalshetty and V. G. Arde, "Validation of CK metrics for object oriented design measurement," 2010 3rd International Conference on Emerging Trends in Engineering and Technology, Goa, 2010, pp. 646-651.

[6] Antony P., "Predicting reliability of software using thresholds of CK metrics", 2013 International Journal of Advanced Networking, pp. 1778–1785.