The Uses of Machine Learning Technique to Improve Software Quality and Testing

SE433 Winter 2021

Group 7

Jenney Chang	jdzchang@gmail.com
Amy Aumpansub	amy.aump@gmail.com
Darrell Lane	darrell.lane25@gmail.com

PROJECT SPECIFICATION

A. CONTEXT

CK-Metrics have been widely used to assess the quality of object-oriented software. The CK-Metrics suite contains Weighted Method per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling Between Objects (CBO), Response for Class (RFC), and Lack of Cohesion (LCOM) to evaluate the quality attributes of object-oriented software development such as classes, inheritance, and encapsulation [1]. During the past few years, several machine learning techniques have been utilized in Software Quality and Testing to automatically learn and improve from experience and data [2].

To efficiently assess and improve the software quality, our team proposes a 3-step method which combines an effective machine learning technique called K-Means Clustering and the well-known CK metrics to efficiently assess and predict software quality and cluster all classes based on their quality. This method is more effective than the traditional method.

To implement our proposed 3-step method, we will utilize CK-Metric values of various releases of Image loaders for Android such as Picasso, Photo View, and Android Universal Image Loader written in Java. The extracted values will be used for the quality assessment and trained as inputs of K-Means Machine Learning model for clustering the classes of each Image Loader based on their predicted quality. The outputs from the machine learning model are the predicted quality of classes grouped into 3 clusters (low, medium, high-quality). Refactoring and testing will be targeted on the classes grouped as 'low-quality' to improve the quality effectively and speedily. The final output of our project is an improved version of various Image Loaders with higher quality. The proposed method comprises 3 steps as follows:

- Software quality assessment using CK metrics
- Quality Prediction and Clustering using K-Means Clustering (Machine Learning)
- Software Quality Improvement on low-quality classes using refactoring and testing

The details of each step will be discussed in Part C: Expected Contribution and Objectives

B. MOTIVATIONS AND PROBLEM STATEMENT

The traditional method to improve software quality may be time-consuming if software developers need to detect the faults in all classes, which software quality may not be improved efficiently and rapidly. Our team aims to utilize CK-Metrics and machine learning to improve software quality. The use of K-mean clustering will group all classes of each software based on measures provided by CK-Metrics which will enable us to focus on low-quality classes during the improvement process. This will increase speed and effectiveness of software quality improvement.

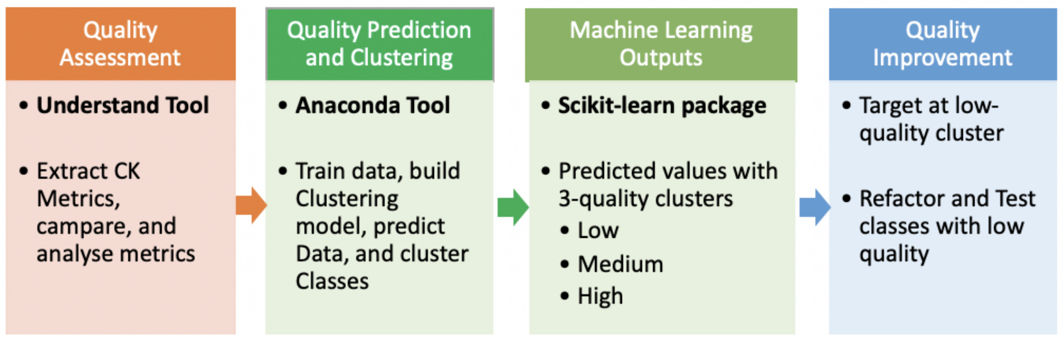
C. EXPECTED CONTRIBUTIONS AND OBJECTIVES

Our 3-step method is aimed to efficiently assess and improve the software quality at the class level in a timely manner. The final output of our project is an improved version of Image Loaders with higher quality. The details of all steps are discussed below:

Step 1: We will utilize the "Understand" tool to extract the metric values from different releases of various Image Loaders for Android such as Picasso, Photo View, and Android Universal Image Loader at class level. The metric values will be compared across releases and image loaders to assess their quality. Additionally, the metric values of each loader will be trained as inputs of the Machine Learning model in step 2.

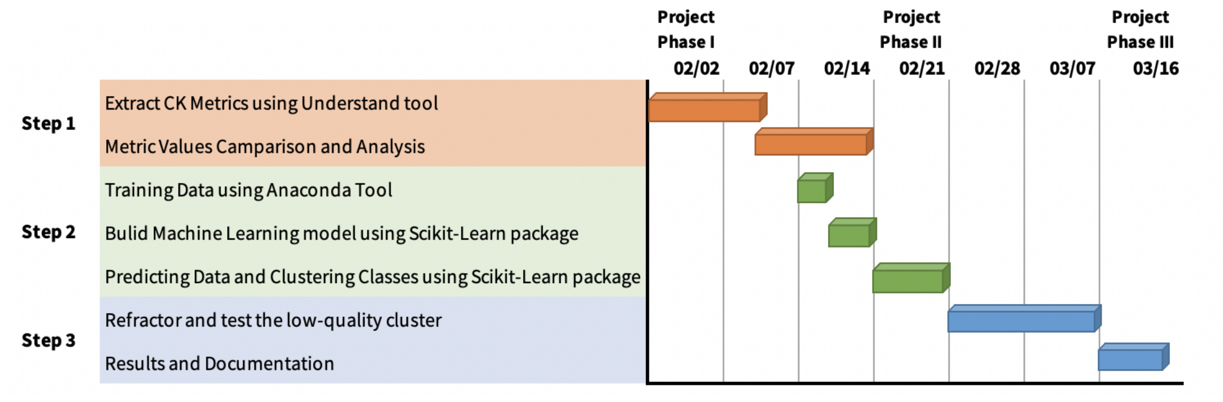
Step 2: The Scikit-learn package in "Anaconda" tool will be utilized to predict the quality of all classes of each image loader and cluster them based on quality. The machine learning technique called "K-means clustering" is designed to group similar data points together and discover underlying patterns shared among data groups [3]. We will use this algorithm to train data and build a strong predictive model to predict quality, discover patterns, and partition all classes of each Image Loader into 3 clusters: low-, medium-, high-quality clusters based on the historical metric values extracted in Step 1. Before fitting the K-mean clustering model, the metric values from Step 1 will be normalized with z-scores and trained. The outputs from the model are the predicted quality of each class grouped into 3 clusters.

Step 3 : We will focus on the low-quality classes (predicted from Step 2) to efficiently refractor and test those classes. We will conduct the in-depth code detection and inspection, then apply refactoring and testing to those classes to improve the software quality. Then, we will write unit tests and compare the metric values of our improved version to the original version to ensure the quality has been improved as expected.


Quality Assessment
• Understand Tool
• Extract CK Metrics, campare, and analyse metrics
Quality Prediction and Clustering
• Anaconda Tool
• Train data, build Clustering model, predict Data, and cluster Classes
Machine Learning Outputs
• Scikit-learn package
• Predicted values with 3-quality clusters
• Low
• Medium
• High
Quality Improvement
• Target at low-quality cluster
• Refactor and Test classes with low quality

D. TASK MANAGEMENT

Our 3-step proposed method contains 7 major tasks as shown by Gantt chart below.



TOOLS

- Understand Tool from https://www.scitools.com
- Anaconda Platform from https://www.anaconda.com/downloads
- Scikit-learn Python Package from https://scikit-learn.org/stable/
- Eclipse from https://www.eclipse.org/downloads/

E. REFERENCES

1. Yeresime Suresh, Lov Kumar, Santanu Ku. Rath, "Statistical and Machine Learning Methods for Software Fault Prediction Using CK Metric Suite: A Comparative Analysis", International Scholarly Research Notices, vol. 2014, Article ID 251083, 15 pages, 2014. https://doi.org/10.1155/2014/251083
2. Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, Mikio Aoyama, Lisa Joeckel, Julien Siebert, Jens Heidrich, "Requirements-Driven Method to Determine Quality Characteristics and Measurements for Machine Learning Software and Its Evaluation", Requirements Engineering Conference (RE) 2020 IEEE 28th International, pp. 260-270, 2020
3. Garbade Michael (2018) "Understanding K-means Clustering in Machine Learning" Retrieved from https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1