# Analysis

## In this project, we investigated 8 classifiers on our dataset which contains code snippets of 5 different programming languages. We believe that there are some patterns in test code snippets, and using these patterns or methods will make the test more likely to be unstable.

Overall, Random Forest exhibits excellent performance with the highest F1-score of 0.918 in distinguishing flaky test cases from non-flaky test cases in C++ projects. Logistic Regression delivers a slightly lower F1-score of 0.914 in Go projects. Perceptron also achieves a satisfied F1-score of 0.906 in Go projects. LDA has the lowest results among all the test combinations, and the lowest F1 score of LDA is 0.541 on the all-language combination subset. The accuracy of LDA on this subset is 0.540, which is also the lowest among all the results.

Random Forest and logistic regression have good performance among all the results. Decision Tree and Perceptron have a slightly lower value than them.

For all the sub-datasets, the KNN classifier’s performance is subpar. It achieves a F1-score of 0.814 on Java dataset. However, it gives a F1-score of 0.295 on Go dataset, with an accuracy of 0.333. Due to the curse of dimensionality and imbalanced data, the accuracy of KNN will decrease a lot. In high-dimensional feature spaces, the distance between data point becomes less meaningful, KNN will struggle to find the nearest neighbors. In imbalanced datasets, if one class significantly outnumbers the other, it tends to classify most of the data to be the majority class.

LDA is a linear model, the dataset we used can be regarded as linear. However, LDA assumes that the data in each class follows a multivariate normal distribution with a shared covariance matrix. Our dataset is imbalanced, it may have significantly different data distributions. Another problem is curse of dimensionality. We have 1500 dimensions in our dataset, but we only have 2 classes. This will lead to poor separability between classes.

Comparing the dataset containing only one programming language (PL) and the datasets containing several programming languages, we found that as the number of PLs increases, the results of the classifiers also show a downward trend. For instance, the F1-score of Random Forest drops from 0.835 to 0.798 as the PL changes from Java to Java and JavaScript. Then it drops to 0.771 on Java, C++ and Python dataset. Datasets comprising a singular PL exhibit linearity, thus favoring the performance of linear classifiers (logistic regression, perceptron…) over non-linear ones on them. In combined datasets, the linear relationships between data elements will be weaker, resulting in a decrease in the linear classifiers overall performance.