# 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.

We create 8 sub-datasets containing different programming languages: Java, Python, C++, Go, Java&JS, Java&Python&C++, No Java and All. Then we run the classifiers on these datasets and analyze the results we obtained.

Overall, all classifiers show a good performance in distinguishing flaky cases from non-flaky test cases except LDA. KNN model has the best performance among all the datasets. We get 4 F1-scores and accuracies which equal to. KNN classifier achieves 1 on Python dataset. Random Forest, Decision Tree and Logistic Regression Classifier get 1 on Go dataset. We checked the confusion matrixes of them. The reason could be that there are very few samples in Python and Go datasets. After train-test split, the test dataset of each contains few samples. For instance, we have 2 positive samples (Flaky) and 25 negative samples (NonFlaky) in the test dataset of Go. It is easy for classifiers to deliver a F1-Score and accuracy equal to 1.

LDA has the lowest results among all the datasets, and the lowest F1 score of LDA is 0.672 on the C++ dataset. The accuracy of LDA on this dataset is 0.625, which is also the lowest among all the results. 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.

On Java dataset, Naive Bayes performs the best, with a F1-score and accuracy both of 0.965. The F1-score of the KNN classifier is slightly lower, which is 0.963. LDA delivers a F1-score of 0.769. Results of other classifiers are similar, the F1-score ranges from 0.936 to 0.959.

On Python dataset, KNN classifier has the highest F1-score and accuracy. It delivers a value of 1 both for F1-score and accuracy. Naïve Bayes classifier ranks second, and its F1-score is 0.968. LDA achieves an F1-score of 0.789, and accuracy of 0.794.

On C++ dataset, Random Forest, SMO, KNN and Logistic Regression classifiers have the same performance. They have a F1-score of 0.956 and an accuracy of 0.958. However, the F1-score of Naïve Bayes model drops to 0.852 on this dataset.

On Go dataset, SMO model achieves a F1-score of 0.899. The other 3 models (Naïve Bayes, KNN and perceptron) share a common value of 0.957 regarding F1-score. LDA has a slightly lower F1-score than SMO, which is 0.882.

On Java&JS dataset, KNN slightly outperforms Random Forest, the F1-score of KNN is 0.967. The performance of other classifiers is similar, and range from 0.93 to 0.96, except LDA. LDA has a F1-score of 0.773.

On Java&py&C++ dataset, KNN has the highest F1-score (0.967), followed by Random Forest, with 0.962.

On no\_java dataset, KNN also delivers the highest F1-score, which is 0.940, followed by Logistic Regression, with 0.933. Comparing with Java&py&C++ dataset, Random Forest drops 0.05, from 0.962 to 0.910. Other classifier, such as Decision Tree, Naïve Bayes and SMO, the F1-scores decrease around 0.07. However, the F1-score of LDA rises from 0.730 to 0.808.

On all language dataset, the highest F1-score is achieved by KNN, which is 0.974, followed by Random Forest, with 0.950.

Comparing the dataset containing only one programming language (PL) and the datasets containing several programming languages, the F1-score and accuracy do not show an obvious trend of change. But for precision and recall, when different languages ​​are added, the occurrence of extreme values ​​is reduced.