The steps being used to train the models in A1 are similar to the steps used in Lab 1. The data preprocessing steps are the same. GridSearchCV is used to tune the hyperparameters in each model. The best estimator given by GridSearchCV is then used to predict accuracy and F1-score on training set and test set.

The accuracy for most trained models on the four datasets are reasonably high. The trained models using decision tree, random forest, svc sigmoid, naive bayes, and svc linear reach around 80% accuracy for all datasets. However, the F1-score for all trained models are not as good. Except for the ones trained with decision tree and random forest, the other models have F1-score ranging from around 13% to 55%.

The discrepancy between accuracy and F1-score implies that the trained models are not actually good at detecting code smells. The high accuracy might be a result of imbalanced datasets. In our dataset, there are about 17% of positive instances and 83% of negative instances, which could make it easy for some models to make many right predictions. When we take into account the F1-score, we see that the models are not powerful classifiers with good balance of precision and recall rates. In order to make the classier models more powerful, we might need to increase the dataset size and do additional feature selections before fitting the data. However, as most of the features in our datasets have been set to zeros, we can infer that some sort of feature selections have been performed already. The datasets have been regularized and reduced in dimension to prevent overfitting. In that case, increasing individual dataset size might be more effective with building stronger classifiers. Or alternatively, we could simply combine four datasets to increase the number of samples. This combined datasets would also be a better representation of real software systems, where various types of code smells co-exist. The resulting classifier would be more relevant in addressing code smell detection issue.

In conclusion, the results from A1 tasks suggest that ML algorithms show different performance in detecting code smells. Higher accuracy and f1-scores are reached when using tree-based algorithms like decision tree and random forest. Arcelli et al. might have reached the conclusion that all ML algorithms have similar performance because of their datasets choice. Their datasets contain metric distributions that are statistically significant, hence easier for ML algorithms to perform predictions on. Moreover, their datasets include many correlated independent variables, which could easily lead to ML algorithms overfitting the data. These factors could have led Arcelli et al. to arrive at a different conclusion than that in A1.