AI in Cyber Security: Assignment 2

Three algorithms were evaluated as potential candidates for the Malware detector; decision tree (DT), logistic regression (LR), and k-Nearest Neighbour (k-NN). In addition, the output of these models was used to train a fourth detector, hence forth referred to as the Ensemble Classifier. DT, LR, and k-NN models were also evaluated as potential candidates for this detector.

1. Feature Engineering.

**DT:** Due to the nature of this model, all 77(the name of the sample was not used) features were used since uninformative features would automatically go unused as they would not be picked as suitable candidates for node splitting criteria.

**LR:** Again, due to the nature of this model, all 77 features were used. This time it was because L2 regularization was used, meaning that the weights of features would be penalized such that only the features which were truly predictive of the target value would have a learnt rate not equal to 0.

**k-NN:** In this case, the features used by the DT after it was trained (detailed in the next section) were used to train the model.

**Ensemble Classifier:** Here, the predicted label from each of the above three classifiers was used as 3 features for the classifier.

2. Model Design, Evaluation, and Debugging:

For evaluating the single models, the dataset was split into two sets with equal class distributions called X\_train and X\_test, which made up 90% and 10% of the total data set respectively. Each model was then trained using grid search with 5-fold cross validation to find the best parameters for the model. In the case where the best param lied on the edge of the search space, the search was repeated with the search space extended.

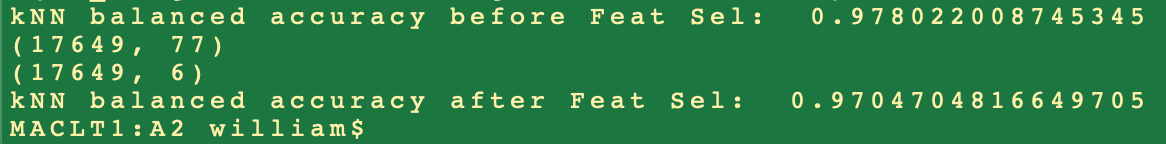
In the case of the Ensemble Classifier, the DT, LR, and K-NN algorithms were also all tested using grid search with 5-fold cross validation except in this case, two separate schemes were tried. In the first, the Ensemble Model was trained on the 3 features discussed previously, and these features were obtained by inputting X\_train into the first three models. Then, the Ensemble Classifier was evaluated on X\_Test using the predicted labels obtained by inputting X\_test into the first three models. In the second scheme, X\_test was split again into X2\_train and X2\_test, both with equal class distributions, and making up 66% and 33% of X\_test respectively. Then, the ensemble was trained on the 3 previously discussed features which were obtained by inputting X2\_train into the first three models. The ensemble Classifier was then evaluated on X2\_Test using the predicted labels obtained by inputting X2\_test into the first three models.

In summary, the first scheme trains the Ensemble Classifier on the predicted lables obtained from the first three models, were the predicted label were the output of the training sets of the first three models but in the second scheme the labels were the output of samples unseen by the first three models during training.

For both the first three models and the ensemble classifier the algorithms tested had the following grid search criteria. k-NN was tested with different values for the number of neighbours and two weight schemes for the neighbours, one being *uniform* where all neighbours have the same weight and the other being *distance* were closers neighbours have a higher weight. LR was tested with different values for the tolerance of the stopping criteria and different values for *C* which controls the strength of the regularization. DT was tested with both the *gini* and *entropy* splitting criterions and with different values for the minimum number of samples per leaf. Minimum number of samples per leaf was chosen because it is a more direct control of overfitting then tree depth. With tree depth limitations, there is a possibility that a leaf with many samples, and thus a leaf that isn’t a “memorization” of the training data and doesn’t contribute to over fitting, is not allowed due to depth limitations. The opposite is also true, where a shallow leaf may be allowed despite having few training samples in it that only correspond to noise. With minimum samples per leaf, complex relationships that require a deep tree can still be learnt while simultaneously guaranteeing that none of the leaf’s are memorizations of random noise in the training data.

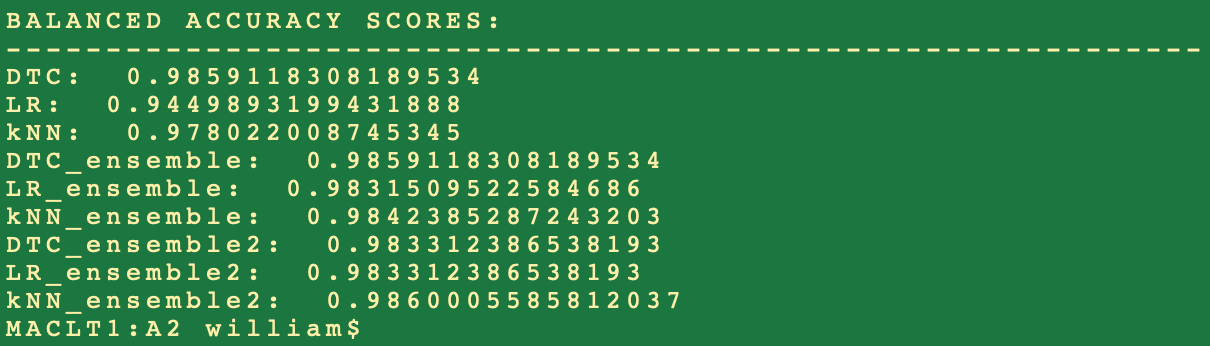
Additionally, for all training runs, the samples were weighted inversely to their classes proportion in the training data. This accounted for class imbalances so the classifier wouldn’t favour the majority class. For all evaluations, balanced accuracy was used since accuracy is misleading when dealing with class imbalances.

Below is a screen shot of the k-NN scores on X\_test before and after feature selection using select by model with the DT.



Because of the poorer performance of the k-NN with feature selection, it was not used for the ensemble and rather k-NN with no feature selection was used.

Below is a screen shot of the scores on X\_test (X2\_test in the case of the ensemble models trained with the second scheme) of the first three models and the ensemble models. The ensemble models trained with the second scheme are indicated with “*ensemble2*”.



3. Novelty Evaluation:

The general rule is that ensemble models generalize better which is most likely the source of the slightly better performance of the ensembles. Since the k-NN ensemble classifier preformed best, I chose to use it to predict the classes of the unlabeled samples. On the next page is the kNN\_ensemle2 predicted labels on the unlabelled samples.

