AU 311 Introduction to Pattern Recognition

By: Harry Lording (717030990012) Andrew Cain (717030990013)

HW#: 5

December 14, 2019

General Results

```
Logistic Regression Accuracy Mean 0.3302, Std 0.1021
SVM Accuracy Mean 0.4016, Std 0.0827
Random Forest Accuracy Mean 0.7865, Std 0.0402
Extra Tree Accuracy Mean 0.7907, Std 0.0448
Neural Network Accuracy Mean 0.5704, Std 0.0449

The best model is:  Extra Tree
Written to csv!
```

The best results obtained was between 0.79 and 0.80 depending on how Extra Tree performed on a given run.

Preprocessing

We preprocessed the data in two ways. First the dimensionality of the data was reduced. If the dimensionality of the data wasn't reduced, the model would not converge after training on the sampples provided. To reduce the dimensionality of the features, the hot encoded soil type features were condensed to a single feature. After the dimension of the data was reduced, the data was normalized, which is important so that the data has a common scale.

Random Forest and Extra Tree Depth

Random Forest and Extra Tree Depth performed the best among the algorithms we tested. Random Forest works well with a mixture of numerical and categorical features such as the features that were used for this classification problem. When features are on the various scales, it is also fine. Roughly speaking, with Random Forest you can use data as they are.

Random Forest performs well with a mixture of numerical and categorical features because of the structure of the model. Random Forest is composed of a "forest" of uncorrelated decision trees. A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

According to articles both random forest and extra tree depth provide similar results like we sure in our findings. The algorithms are separated in how they split the data, extra tree depth splitting the data stochastically while random forest splits data deterministically. Also extra tree depth also performs better when more noisy features are present. Given that we didn't remove any feature inputs from the data it is likely that a few had no influence on the cover type of an area of forest. This could explain why Extra Tree Depth performed better than Random tree

Tuning the parameters of Random Tree and Extra Tree Depth:

Changing the number of trees (from 500 to 1000) improved the performance of random forest (Accuracy 500: Mean 0.7765, Std 0.0377, Accuracy 1000: Mean 0.7783, Std 0.0368) however performance worsen for

extra tree (Accuracy 500: Mean 0.7915, Std 0.0381, Accuracy 1000: Mean 0.7913, Std 0.0385). Interestingly, under most circumstances adding more trees to random forest or extra tree should only improve the performance of the algorithms. This could have occurred if the added trees were correlated to existing trees. In this case it could have distorted the voting process of the trees in the Extra Depth Tree algorithm.

Logistic Regression

Logistic regression works best when the features of the model are well known and the features are closely correlated to the outcome. It is less effective when some of the input variables are not known, or when there are complex relationships between the input variables.

Given the performance of logistic regression it would appear that the data contains a number of noisy features that do not correlated closely to the output of the model, as is suggested with the strong performance of random forest.

Tuning the parameters of Logistic Regression:

The only parameter to tune for Logistic Regression was the iterations performed. At roughly 50 max iter Logistic Regression reach above 0.3319 accuracy which was close to its convergence of 0.3390.

Logistic Regression 10 iterations: Accuracy Mean 0.2610, Std 0.0828
Logistic Regression 25 iterations: Accuracy Mean 0.3092, Std 0.0778
Logistic Regression 50 iterations: Accuracy Mean 0.3319, Std 0.0940
Logistic Regression 1000 iterations: Accuracy Mean 0.3390, Std 0.0875

SVM

Random Forest is intrinsically suited for multiclass problems, while SVM is intrinsically two-class. For multiclass problem you will need to reduce it into multiple binary classification problems i.e. a one vs all problem. As a result SVM would not perform as well as algorithms which are designed to perform well with classification problems that have more than 2 classes, such as this classification problem which had 7 classes.

Additionally, SVM maximizes the "margin" and thus relies on the concept of "distance" between different points. It is up to you to decide if "distance" is meaningful. One-hot encoding for categorical features is a must-do. It would seem then that after reducing the dimensionality by turning the one-hot encoded soil type feature into a single numeric could have worsen the performance of SVM. However after testing it seemed that this was not the case,with SVM performing exactly the same with data structure both ways.

Tuning the parameters of SVM:

The kernels for the SVM needed to be tuned inorder to optimise the performance of SVM. The following kernels gave the results below:

Linear: SVM Accuracy Mean 0.3464, Std 0.0774
Poly: SVM Accuracy Mean 0.4497, Std 0.0348
Gaussian: SVM Accuracy Mean 0.4024, Std 0.0432
Sigmoid: SVM Accuracy Mean 0.1667, Std 0.0184

Neural Network

From tuning the hyperparameters the highest accuracy that could be achieved with a neural network was 0.5520. Likely that the training set is not large enough for the neural network to be as effective as other models for this problem.

Tuning the parameters of Neural Network:

```
(solver='adam', max_iter=1000, learning_rate_init=0.001, alpha=1e-5, hidden_layer_sizes=(10, 2), random
Neural Network Accuracy Mean 0.1429, Std 0.0000

(solver='adam', max_iter=1000, learning_rate_init=0.005, alpha=1e-5, hidden_layer_sizes=(5, 2), random_
Neural Network Accuracy Mean 0.5435, Std 0.0365

(solver='adam', max_iter=100, learning_rate_init=0.01, alpha=1e-5, hidden_layer_sizes=(5, 2), random_sta
Neural Network Accuracy Mean 0.5313, Std 0.0322

(solver='adam', max_iter=500, learning_rate_init=0.01, alpha=1e-5, hidden_layer_sizes=(5, 2), random_sta
Neural Network Accuracy Mean 0.5520, Std 0.0206

(solver='adam', max_iter=1000, learning_rate_init=0.03, alpha=1e-5, hidden_layer_sizes=(5, 2), random_s
Neural Network Accuracy Mean 0.5444, Std 0.0194


(solver='adam', max_iter=500, learning_rate_init=0.01, alpha=1e-5, hidden_layer_sizes=(4, 2), random_sta
Neural Network Accuracy Mean 0.5299, Std 0.0286

(solver='adam', max_iter=500, learning_rate_init=0.01, alpha=1e-5, hidden_layer_sizes=(6, 2), random_sta
Neural Network Accuracy Mean 0.5482, Std 0.0414

(solver='adam', max_iter=500, learning_rate_init=0.01, alpha=1e-5, hidden_layer_sizes=(5, 2), random_sta
Neural Network Accuracy Mean 0.4561, Std 0.1571
```

K-Folds

Below is the results from varying k-fold values. The best result was 0.7921 with Extra Tree Depth and 4 folds. For both Extra Tree Depth and Random Forest, adding more folds didn't necessarily improve the performance of the models, both reaching optimal performance at 4-5 folds. For Linear Regression and SVM, the more folds in cross validation, the better the accuracy. Interestingly the performance of Neural Networks fluctuated randomly. While we can not explain why this occurred it appeared that the model performed better for an even number of folds.

```
2 folds:

Logistic Regression Accuracy Mean 0.3034, Std 0.0342
SVM Accuracy Mean 0.3685, Std 0.0495
Random Forest Accuracy Mean 0.6985, Std 0.0944
Extra Tree Accuracy Mean 0.7079, Std 0.0853
Neural Network Accuracy Mean 0.3308, Std 0.1879

The best model is:  Extra Tree

3 folds:

Logistic Regression Accuracy Mean 0.3138, Std 0.0628
SVM Accuracy Mean 0.4079, Std 0.0494
Random Forest Accuracy Mean 0.7738, Std 0.0127
Extra Tree Accuracy Mean 0.7744, Std 0.0152
Neural Network Accuracy Mean 0.2958, Std 0.2162

The best model is:  Extra Tree

4 folds:

Logistic Regression Accuracy Mean 0.3302, Std 0.1021
SVM Accuracy Mean 0.4016, Std 0.0827
Random Forest Accuracy Mean 0.7857, Std 0.0403
Extra Tree Accuracy Mean 0.7921, Std 0.0449
Neural Network Accuracy Mean 0.3583, Std 0.2192

The best model is:  Extra Tree

5 folds:

Logistic Regression Accuracy Mean 0.3390, Std 0.0875
SVM Accuracy Mean 0.4154, Std 0.0364
Random Forest Accuracy Mean 0.7749, Std 0.0425
Extra Tree Accuracy Mean 0.7880, Std 0.0409
Neural Network Accuracy Mean 0.2325, Std 0.1792

The best model is:  Extra Tree

6 folds:
```

```
Logistic Regression Accuracy Mean 0.3405, Std 0.1013
SVM Accuracy Mean 0.4164, Std 0.0578
Random Forest Accuracy Mean 0.7779, Std 0.0387
Extra Tree Accuracy Mean 0.7863, Std 0.0403
Neural Network Accuracy Mean 0.4415, Std 0.2114

The best model is:  Extra Tree

7 folds:

Logistic Regression Accuracy Mean 0.3476, Std 0.1220
SVM Accuracy Mean 0.4192, Std 0.0845
Random Forest Accuracy Mean 0.7825, Std 0.0399
Extra Tree Accuracy Mean 0.7892, Std 0.0398
Neural Network Accuracy Mean 0.1429, Std 0.0002

The best model is:  Extra Tree

8 folds:

Logistic Regression Accuracy Mean 0.3455, Std 0.1217
SVM Accuracy Mean 0.4323, Std 0.0872
Random Forest Accuracy Mean 0.7771, Std 0.0523
Extra Tree Accuracy Mean 0.7854, Std 0.0500
Neural Network Accuracy Mean 0.2455, Std 0.1787

The best model is:  Extra Tree
```