# Machine Learning Coursework Notes

Propose doing Decision Tree compared to Random Forest

## Datasets

### Letter Image Recognition Data

* <http://archive.ics.uci.edu/ml/datasets/Letter+Recognition>
* 16 features describing letters in various fonts and distortions
* 1 target (Letter), 26 classes
* 20 000 examples
* Poor papers
* Clean dataset, balanced
* Default tree gives approx. 60% predictability, papers gives 80%

To Aaron:

This is an optional step. However, if there is a significant class imbalance in your dataset, it's reasonable to apply oversampling/undersampling during cross validation (only on training set, not validation samples). When you compare algorithms, the two models have to be retrained on the entire training set with e.g. SMOTE using the optimal hyper-parameters discovered during model selection. Yes, you can present the best performing one. To enrich your analysis, optionally you could investigate the impact of varying degrees of SMOTE on our final models.

# Dataset Notes on Plots

The scattermatrix and correlation matrix used together that:

* Width and Height are highly correlated and have a linear distribution with xbox and ybox respectively. This is likely to be more than correlation: association.
* P-values correlation are high for yBox and yEgvy.

Removing the yBox is worth a try. Removing xBox less so.

The images and results csv files show that removing yBox and yEgvy slightly reduces the accuracy, without noticeably reducing the overall run time.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **HoldOut** | **Train Prop.** | **Max**  **Split** | **Split Crit.** | **Train**  **Acc.** | **Test**  **Acc.** | **Avg**  **Misclas.** | **Entry**  **Count** | **CPU**  **Time** |
| 20 | 0.7 | 25 | gdi | 0.3686 | 0.3654 | 3056.85 | 4800 | 22.875 |
| 15 | 0.7 | 25 | gdi | 0.3691 | 0.3647 | 3049 | 4800 | 17.7656 |
| 5 | 0.7 | 25 | gdi | 0.3654 | 0.3602 | 3077.6 | 4800 | 5.5938 |
| 15 | 0.8 | 100 | deviance | 0.675 | 0.6656 | 1072.0667 | 3199 | 12.7813 |
| 10 | 0.8 | 100 | deviance | 0.6764 | 0.6645 | 1078 | 3199 | 9.0625 |
| 20 | 0.8 | 100 | deviance | 0.6739 | 0.6631 | 1083.5 | 3199 | 18.5625 |
| 1 | 0.8 | 100 | deviance | 0.6829 | 0.6608 | 1072 | 3199 | 0.8594 |
| 5 | 0.8 | 100 | deviance | 0.6738 | 0.6583 | 1099.6 | 3199 | 4.5 |
| 15 | 0.8 | 75 | deviance | 0.6499 | 0.6391 | 1157.5333 | 3199 | 14.8125 |
| 20 | 0.8 | 75 | deviance | 0.6485 | 0.6389 | 1155.4 | 3199 | 18.0625 |
| 5 | 0.8 | 75 | deviance | 0.6491 | 0.6368 | 1158 | 3199 | 4.4531 |
| 10 | 0.8 | 75 | deviance | 0.6472 | 0.6333 | 1178 | 3199 | 8.9063 |
| 1 | 0.8 | 75 | deviance | 0.6422 | 0.6292 | 1188 | 3199 | 0.9219 |
| 1 | 0.8 | 50 | deviance | 0.5915 | 0.5848 | 1339 | 3199 | 0.875 |
| 5 | 0.8 | 50 | deviance | 0.5882 | 0.5812 | 1349.8 | 3199 | 4.4063 |
| 20 | 0.8 | 50 | deviance | 0.5876 | 0.581 | 1345.85 | 3199 | 19.8125 |
| 15 | 0.8 | 50 | deviance | 0.586 | 0.5809 | 1342.4 | 3199 | 12.6563 |
| 10 | 0.8 | 50 | deviance | 0.5852 | 0.577 | 1354.5 | 3199 | 8.7344 |
| 5 | 0.8 | 25 | deviance | 0.4755 | 0.4724 | 1678.4 | 3199 | 3.9063 |

The results for all features is used to select the hyperparameters to try for the deviance split criterion:

* **Split:** deviance
* **Hold out:** 1 5 10
* **Max split:** 75 100 125 150
* **Train / Test Proportion:** 0.8