Classification – Land Use

Introduction

Land use classification is an important task to understand our changing environment. One approach to this involves the use of data from aerial sensors that captures different spectral reflectance properties of the ground below. From this data, the land type can be classified, which has significant effect on environment protection.

Dataset Description

The dataset supplied contains 27 spectral properties and an overall classification of land type (`Class`), which can be one of the following:

- *s*: 'Sugi' forest;
- *h*: 'Hinoki' forest;
- *d*: 'Mixed deciduous' forest;
- *o*: 'Other' non-forest land

Data Preparation

There are two datasets provided for training `Q2/training.csv` and testing `Q2/testing.csv`. Both datasets 28 attributes, with the classification attribute of Class. Training dataset contains 198 rows and testing dataset contains 325 rows.

Class Imbalance

Upon exploring the datasets, there is a significant class imbalance in both training and testing datasets. Moreover, the classification distributions in the two datasets are not comparable. This issue needs to be addressed when we use the testing dataset.

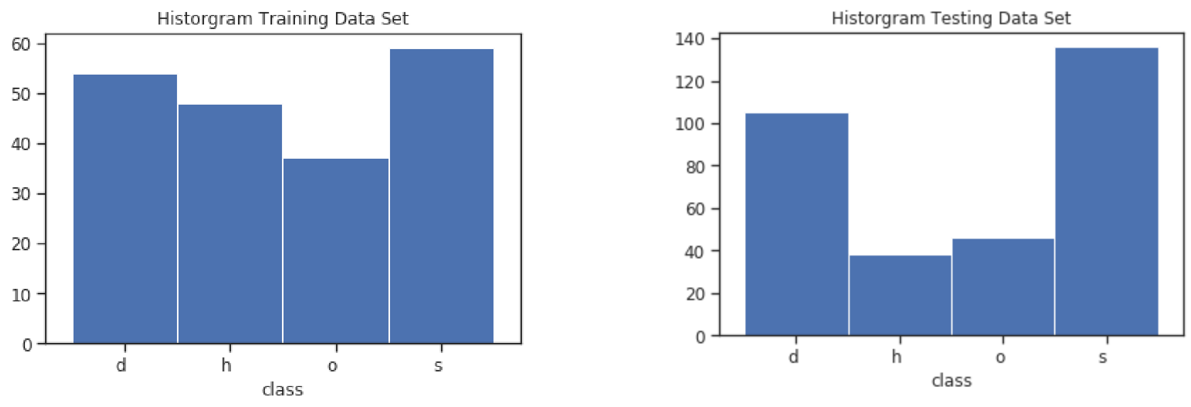


Figure 2.1: Distribution of classification in datasets

Missing Values

There are no missing values in both the datasets.

Splitting Data

The number of rows in the supplied testing set is higher than the training set. Hence, we decided to split the testing dataset into validation set (40%) and testing set (60%). This should give us enough data to train, validate and test our models.

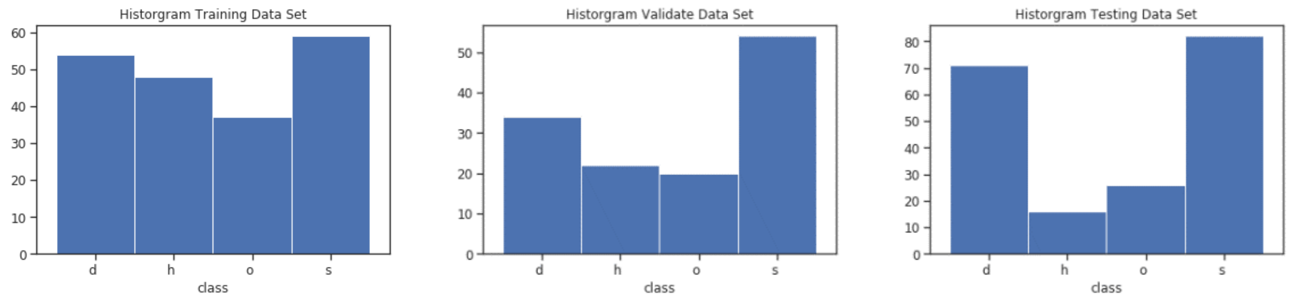


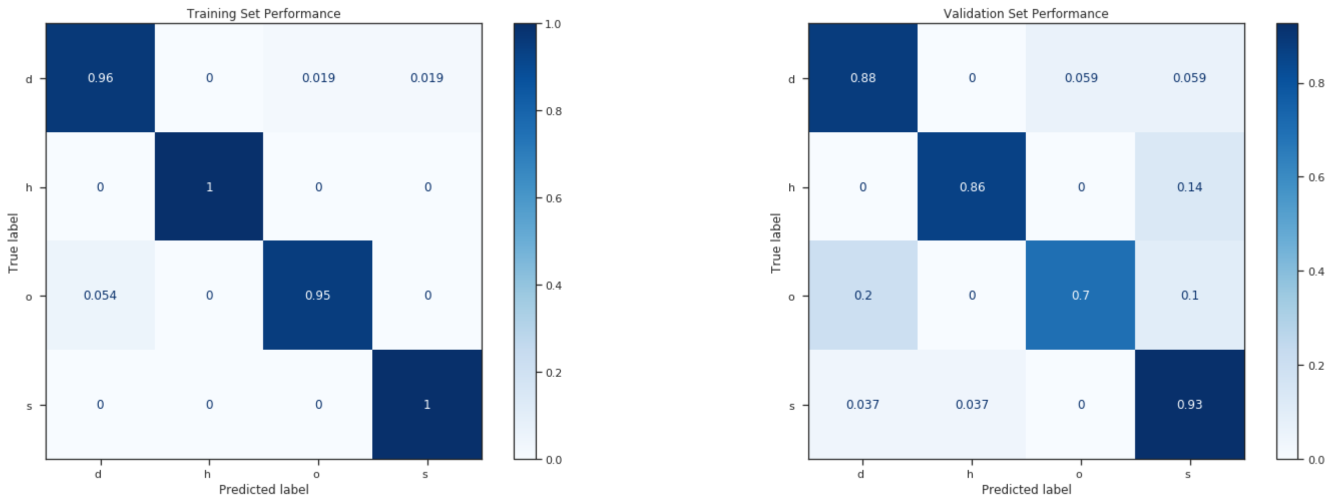
Figure 2.2: Distribution of classification of training, validation and testing datasets

K-Nearest Neighbours Classifier

K-Nearest Neighbours determines classification based on finding similar points. It has one parameter to specify, which is the number of neighbours.

We start with three arbitrary for the number of neighbours: 3, 7, 13.

```
CKNN, with No. Neighbours = 3
Misclassified samples: 17
Accuracy (Validation Set): 0.8692
```



Training Set Performance:

	d	h	o	s
d	0.96	0	0.019	0.019
h	0	1	0	0
o	0.054	0	0.95	0
s	0	0	0	1

Validation Set Performance:

	d	h	o	s
d	0.88	0	0.059	0.059
h	0	0.86	0	0.14
o	0.2	0	0.7	0.1
s	0.037	0.037	0	0.93

Figure 2.3: CKNN Models (No. of Neighbour=3)

```
CKNN, with No. Neighbours = 7
Misclassified samples: 20
Accuracy (Validation Set): 0.8462
```

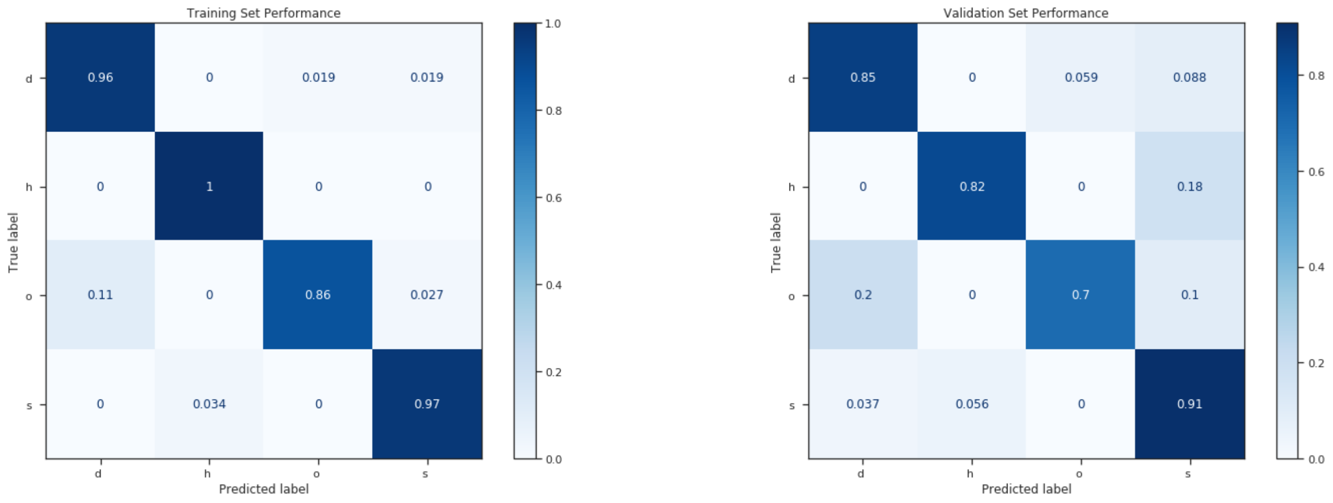


Figure 2.4: CKNN Models (No. of Neighbour=7)

```
CKNN, with No. Neighbours = 13
Misclassified samples: 18
Accuracy (Validation Set): 0.8615
```

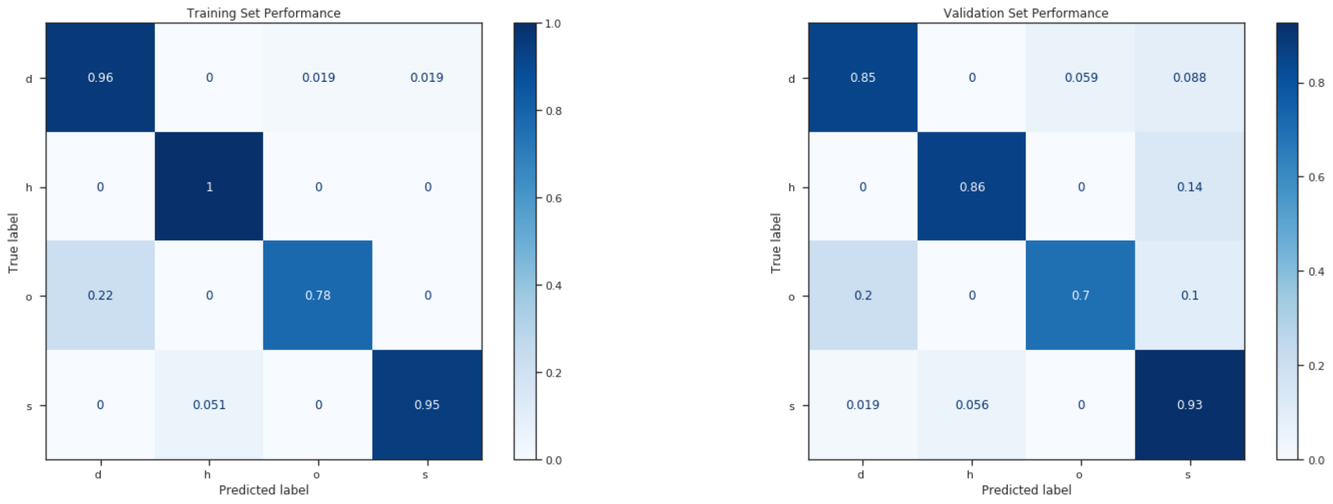


Figure 2.5: CKNN Models (No. of Neighbour=13)

The CKNN accuracy results from validation set were quite comparable, with number of neighbours being the best. From the confusion matrix, it is important to note that prediction performance for class “o” in validation set has been consistently poor. This is due to the class imbalance mentioned previously.

For our final CKNN model, 3 is selected for the parameter value of number of neighbours. With lower the number of neighbours, higher the chance for the model to get rare classes correct. In this case we would like the model to predict rare classes “*o*” more accurately.

Ensemble Classifiers

One vs. One

```
One vs. One
Misclassified samples: 19
Accuracy (Validation Set): 0.8539
```

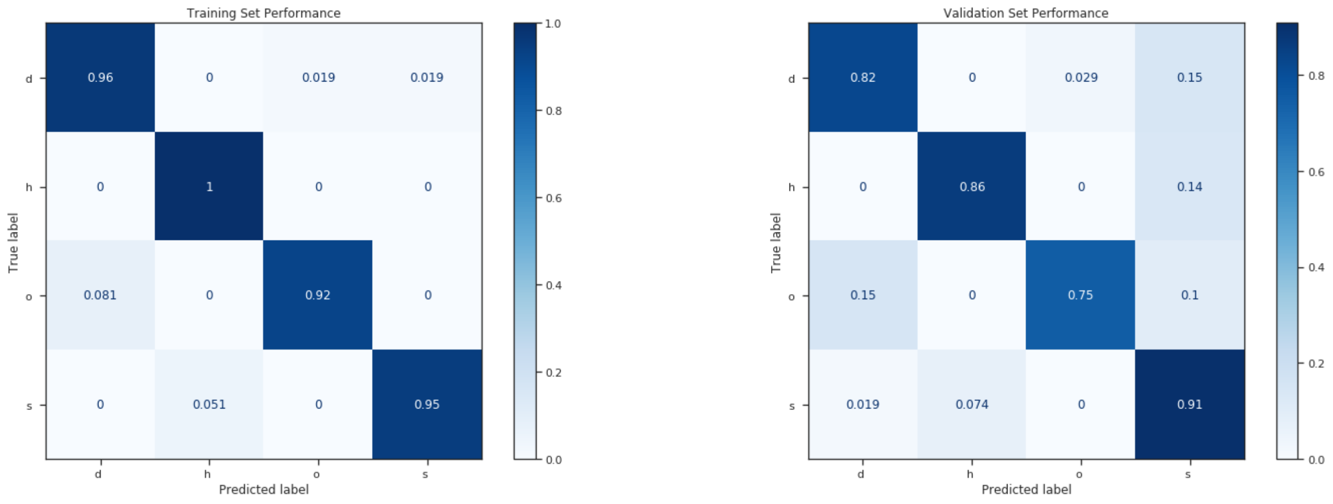


Figure 2.6: One vs. One

One vs. All

```
One vs. All
Misclassified samples: 21
Accuracy (Validation Set): 0.8385
```

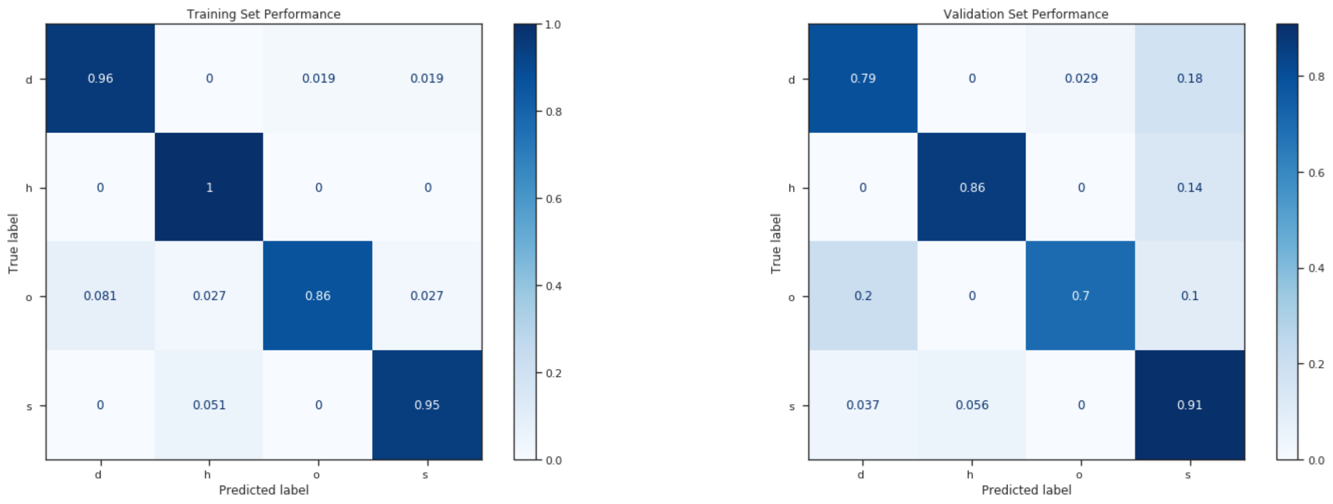


Figure 2.7: One vs. All

We train our model by balancing the weight of under-sampled classes, as there is a class imbalance in the dataset. But the prediction accuracy on validation set are not as good as the CKNN model.

Hyper-parameter Selection

SVM has several hyper parameters which are configurable for optimisation. We perform a series of evaluations by passing different values for these hyperparameters. The strategy here is to search through 3 kernels (linear, RBF and Polynomial), together with a range of hyperparameter values. In this case, we consider different values of C for Linear kernel, Gamma for RBF and Degree for Polynomial. Finally, the final model is selected by considering its prediction accuracy on using validation set.

		Misclassified samples	Accuracy
	Linear SVM, with C = inf	20	0.8462
	Linear SVM, with C = 100	20	0.8462
	Linear SVM, with C = 10	20	0.8462
	Linear SVM, with C = 0.1	20	0.8462
	Linear SVM, with C = 0.01	18	0.8615
	Linear SVM, with C = 0.001	17	0.8692
	Linear SVM, with C = 0.0001	16	0.8769
	Linear SVM, with C = 0.00001	29	0.7769
	RBF SVM, with scale = 0.01	55	0.5769
	RBF SVM, with scale = 0.001	16	0.8769
	RBF SVM, with scale = 0.0001	17	0.8692
	RBF SVM, with scale = scale	18	0.8615
	Polynomial Kernel SVM, with degree = 2	17	0.8692
*	Polynomial Kernel SVM, with degree = 3	15	0.8846
	Polynomial Kernel SVM, with degree = 5	16	0.8769
	Polynomial Kernel SVM, with degree = 10	18	0.8615

Figure 2.8: Hyperparameters Optimisation

As a result, “Polynomial Kernel SVM, with degree=3” is selected as it produces the best prediction accuracy.

Model Evaluation

The performance of the final two models on testing set is summarised below:

```
CKNN, with No. Neighbours = 3
Misclassified samples: 35
Accuracy (Testing Set): 0.8205
```

```
Polynomial Kernel SVM, with degree = 3
Misclassified samples: 42
Accuracy (Testing Set): 0.7846
```

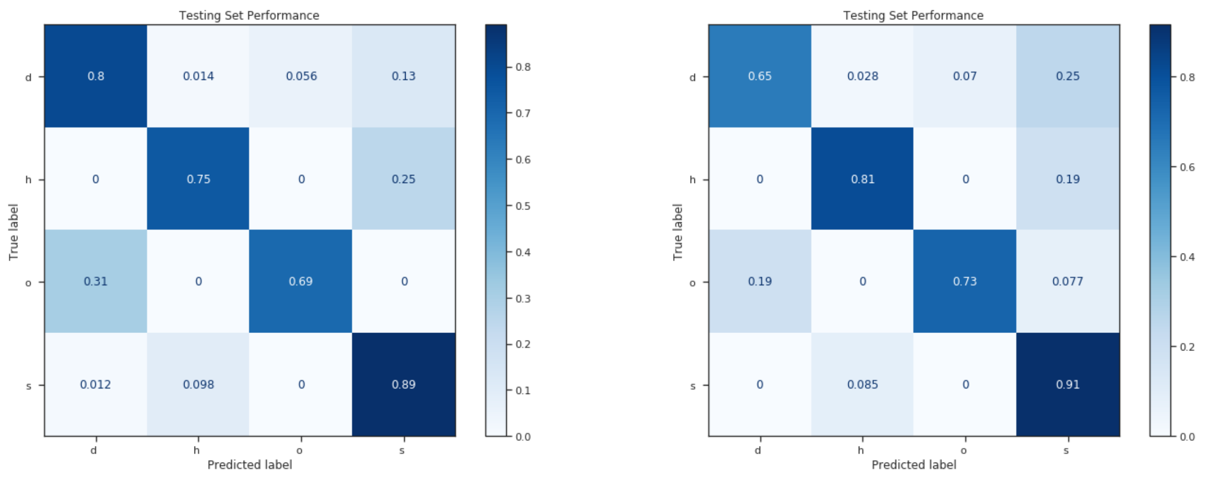


Figure 2.9: Final models comparison

According to our testing, model CKNN performed better in prediction accuracy (>80%), with 7 misclassified sample difference from our Polynomial SVM model. However, in comparison, CKNN has a higher propensity in misclassify class “h” to “s” (0.25) and “o” to “d” to “s” (0.31), which might have been attributed by the class imbalance in both the training and testing datasets.

Another difference to note is that SVM Polynomial Kernel model performs slightly better than CKNN in correctly classifying the rare class “*o*” (0.73 vs 0.69), in which has the lowest proportion of data in both training and testing dataset. In real-world context, as it is the only non-forest classification, it may be important to identify this land type classification correctly.