CS5014 MACHINE LEARNING

Classification of object colour using optical spectroscopy

APRIL 20, 2018

Lecturer:
David Harris-Birtill
Kasim Terzić

Submitted By:
140011146

1 Introduction

In this practical, two classification tasks were performed on an experimental dataset.

2 Binary Classification

2.1 Preprocessing

Before any preprocessing is done, the data was split into training and test datasets. This is to ensure no bias when looking at the data, even for visualisations. The split is also stratified over the output to ensure the training and test sets have the same proportion of classes that is found in the original data. This again is done to prevent bias and overfitting as much as possible so that the trained model is not overfit to a subset of the data with a different proportion of classes.

Nothing was done about the negative intensities found in the data as this came from issues during data collection where the distance of the spectrometer changed. It is not trivial to account for the errors as the distance moved is not known. Further, scaling to the negative values to keep the intensities going from 0-100% is likely to skew the data in an unknown way. Removing all data points where negative values exist would also hurt as the dataset is already quite small. TODO. Because of these reasons, the data was not cleaned for negative intensities.

2.1.1 Data visualisation

First, the spectroscopy data can be visualised by plotting the intensities and wavelengths to see if there are visual patterns than can be seen in the data. As there are over nine hundred input features, it is likely many of them are either redundant or not well correlated to the input data.
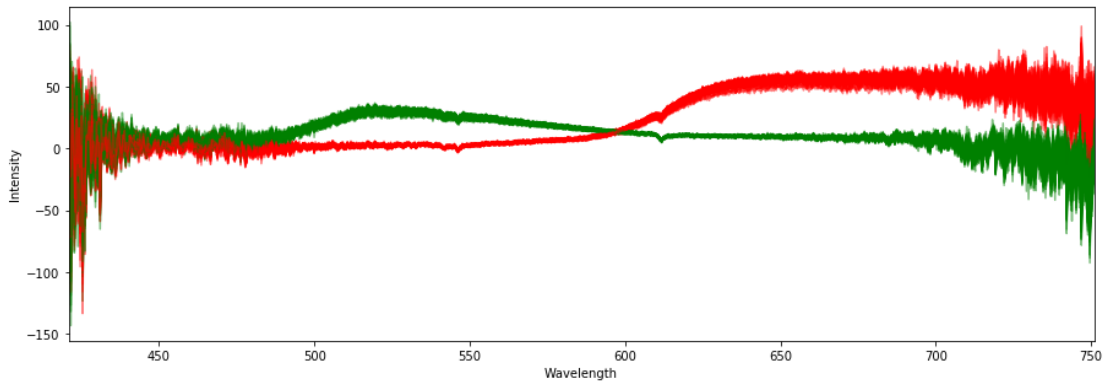

Figure 1: Initial plot of the data, showing the intensity of the wavelengths measured. The plot is colour coded with the output so any differences between the red and green data can be easily seen.

From the visualisation, it is interesting to see that for many wavelengths, there is a very clear distinction between the red and green intensities. This suggests that for those wavelengths, it would be very simple to classify. Further, the visualisation was created from all the training data, so it can be seen there are no large errors in the middle wavelengths where no data points incurred significant differences, which shows the consistency of the data. It can be see towards the start and end of the spectrum, the intensities are more varied, indicating possible errors during measurement. As such, it would make sense when choosing input features to mostly include features from the middle wavelengths where less possible errors occurred.

2.1.2 Data correlation

Next, the correlation of each input feature wavelength can be calculated and plotted to see how the set of input features is correlated to the output. It is not feasible to plot each input feature's correlation separately due to the large number of features. Therefore all the features are put onto the same graph for comparison.
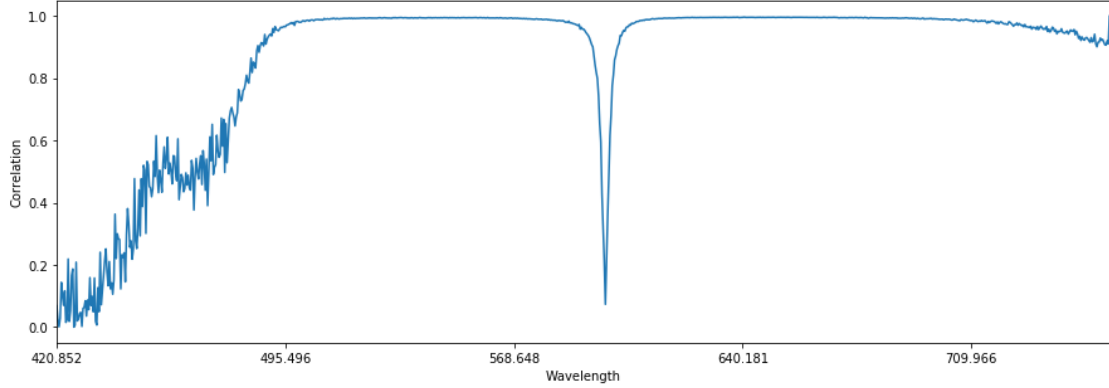

Figure 2: Correlation of each wavelength input feature to the output classification.

The curve of input feature correlation corresponds very closely to figure 1. At the beginning where there seems to be more noise, the correlation is low. At the wavelengths where the red and green curves start to become distinct from each other, the correlation of that wavelength to the output class tends to perfect correlation. At the center where the red and green curves cross over, a dip in the correlation can be seen as it becomes difficult to distinguish the two in the overlap. This is good news and expected, as the clear difference between the red and green curves corresponds to a clear classification.

To show this more clearly, the histograms of a single input feature can be visualised.
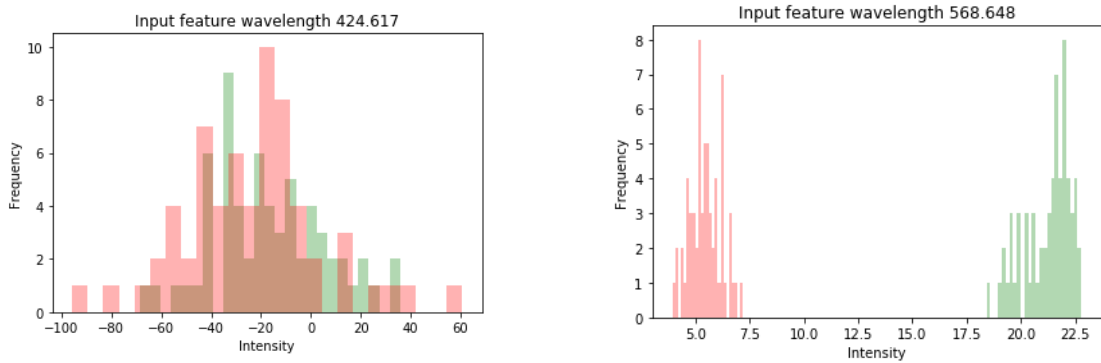
(a) Histogram of a wavelength where there is high overlap and low correlation between the red and green curves.

(b) Histogram of a wavelength where the red and green curves are distinct and there is perfect correlation with the output class.

Figure 3: Comparison of histograms of a single wavelength measurement. One taken at a low correlation wavelength and the other at a high correlation wavelength.

The histograms in figure 3 show the difference between an input feature with low correlation and an input feature with high correlation. The two wavelengths were arbitrarily chosen based on the correlation graph, but an obvious difference can still be seen. With low correlation, there is a lot of

overlap between the two classes, which could also have been seen in figure 1. For the input feature with high correlation, the histogram shows very clearly the separation between the two classes. Figure 4 further supports the fact that all input feature wavelengths where the correlation is high and a clear distinction seen in figure 1 have clear, separate intensities. As the histograms contain all data points from the split training set, it is very likely that a single input feature can lead to a high classification rate.
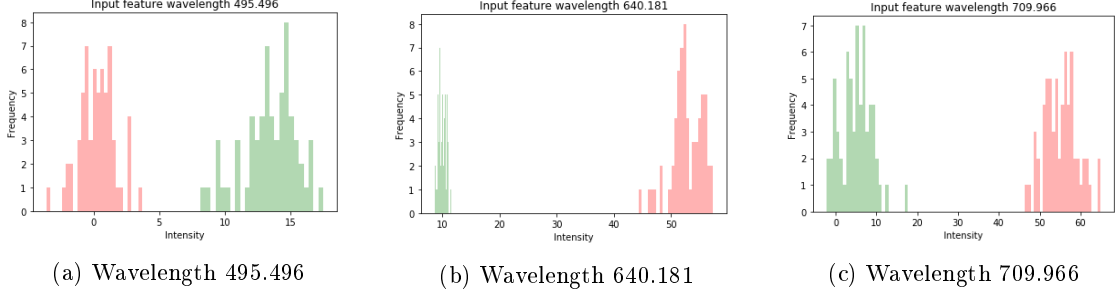

(a) Wavelength 495.496   (b) Wavelength 640.181   (c) Wavelength 709.966

Figure 4: Further histograms of highly correlated wavelengths, showing the separation of the two classes occurs for many different input features.

2.2 Results

From the preprocessing stage, it was shown that there are multiple input features where a clear distinction using just that single input feature could be seen between the two classes. As such, a single input feature is tried first for training to see what classification rates could be obtained.

Input feature	Accuracy
Wavelength 421.228	0.5
Wavelength 495.496	1.0
Wavelength 568.648	1.0
Wavelength 640.181	1.0
Wavelength 709.966	1.0

Table 1: Results of training a logistic regression model on a single input feature.

Despite the indications that the data would be easy to classify, the predictions from the logistic regression model are suspiciously high. All single input features used that had a high correlation gave perfect accuracy during prediction. The first thought it perhaps the model is overfitting, so the same training is done again with k-fold cross validation.

Input feature	5 fold accuracy	10 fold accuracy	25 fold accuracy
Wavelength 421.228	0.5	0.49	0.5
Wavelength 495.496	1.0	1.0	1.0
Wavelength 568.648	1.0	1.0	1.0
Wavelength 640.181	1.0	1.0	1.0
Wavelength 709.966	1.0	1.0	1.0

Table 2: Accuracy of using k-fold cross validation to show the logistic regression model is not overfitting.

Even with using a high k-fold cross validation, the strong input features continued to get 100% accuracy, showing it is unlikely due to overfitting but because the data and those specific input features lends themselves well to a binary classification. It was further confirmed how well defined

the classification was by splitting the training data into an even smaller dataset with only 10% of the training dataset used for training and testing on the rest of the training data.

Input feature	Training accuracy	Testing accuracy
Wavelength 421.228	0.67	0.46
Wavelength 495.496	1.0	0.98
Wavelength 568.648	1.0	0.99
Wavelength 640.181	1.0	1.0
Wavelength 709.966	1.0	1.0

Table 3: Accuracy of training logistic regression on 10% of the training data and testing on the remaining 90%.

Finally, the use of a single feature obtaining such a high classification rate was confirmed with the use of recursive feature elimination with cross validation. This shows how the accuracy score improves as the number of features used increases TODO.
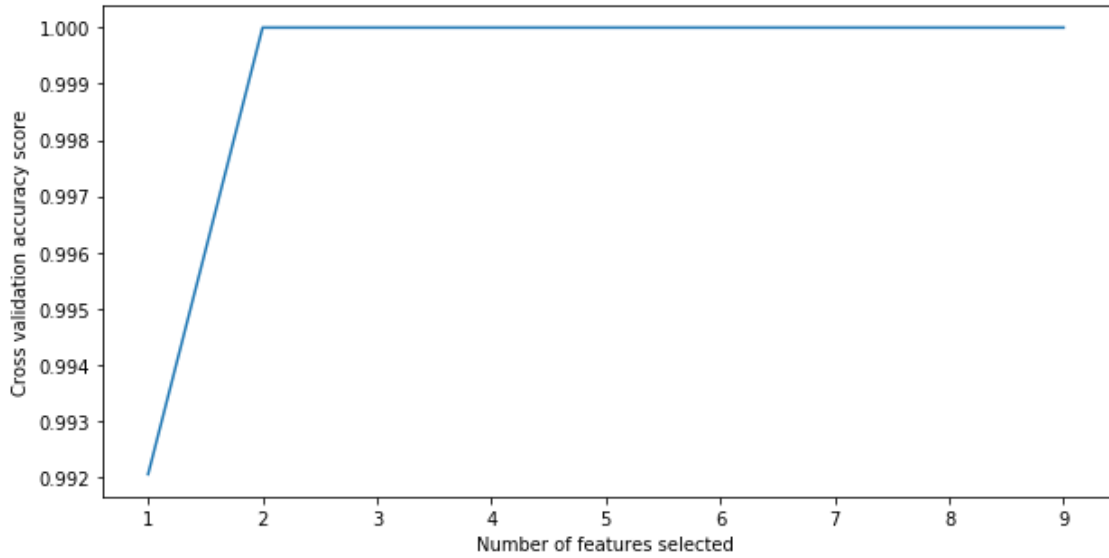

Figure 5: Cross validation accuracy score of increasing number of features using recursive feature elimination.

The fact that this classification problem could be easily solved with the simplest logistic regression model using only one input feature and a minuscule number of training samples shows TODO.

3 Multiclass Classification

3.1 Preprocessing
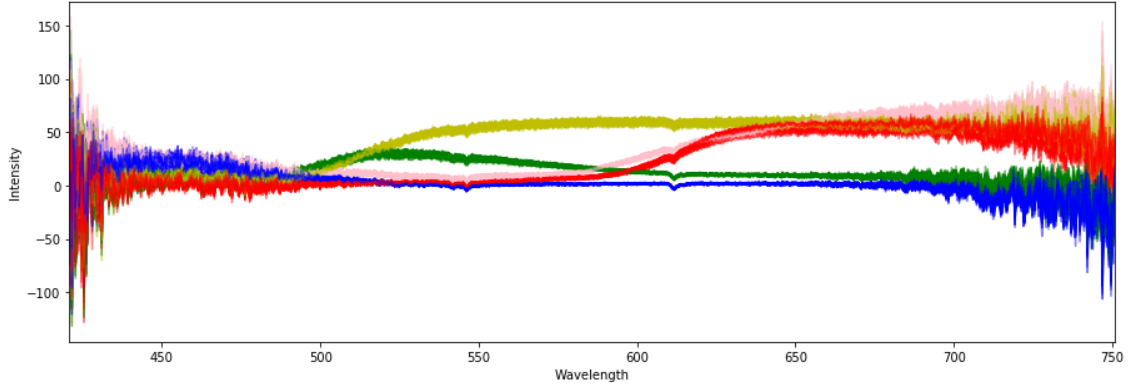
3.1.1 Data visualisation

Figure 6: Plot of the data to show the intensity of wavelengths of the different colour classes.
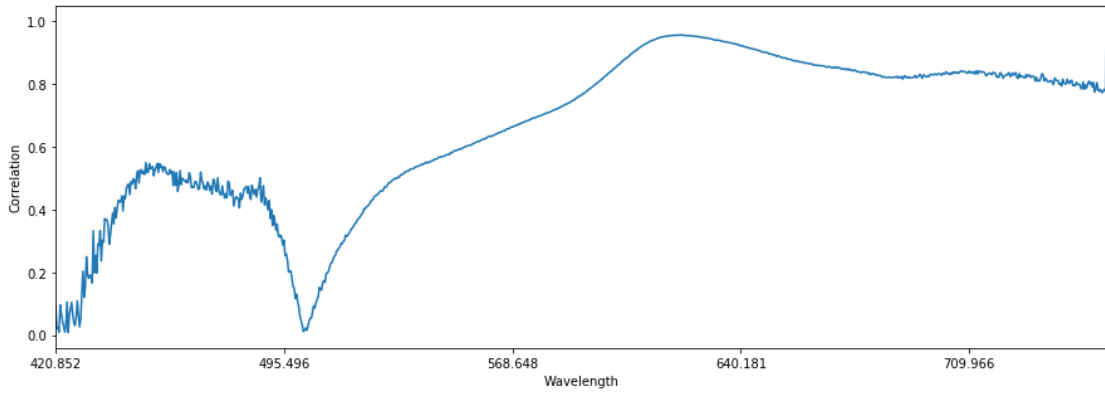
3.1.2 Data correlation

Figure 7: Correlation of each wavelength input feature to the colour classification.

3.2 Results

4 Conclusion and Evaluation