# CS5014 - Machine Learning

## P2 - Classification of object colour using optical spectroscopy

## Report

140014952

April 20, 2018

# 1 Overview

The objectives of this practical were to come up with classification model for binary and multi-class classification problems. This submission investigates both binary and multi-class tasks. The solution python scripts can be found in */binaryML/* and */multiclassML/* directories.

As later sections suggest, the amount of features it takes to determine the class for each task is very low. This is hypothesized based on input analysis and later machine learning observations support these hypotheses.

Final predicted class files are recorded in the appendices section of this report and in directories */binaryTask/* and */multiClassTask/*.

# 2 Binary Classification Task

## 2.1 Cleaning data and Feature extraction

As the very first step input data were split into training and testing data with 70-to-30 ratio. The analysis were first done on the training set. Sklearn's *train_test_split* was used to do so. Also, a seed was used to ensure the same samples are used every time python notebook is fully executed.

From observations and further analysis in python, data cleaning was not necessary. Data contains negative and positive values that according to the practical specification make sense.

However, a functionality to remove all records with null values in was implemented to ensure that the samples are fully prepared.

## 2.2 Data analysis and Visualization

To visualize training data X it was plotted with provided wavelength data that contains information about each feature wave length. Additionally, Y training set was used to indicate visually how colours are distributed and can be distinguished. This gave insights as to which features are likely to be good indicators for predicting the colour.

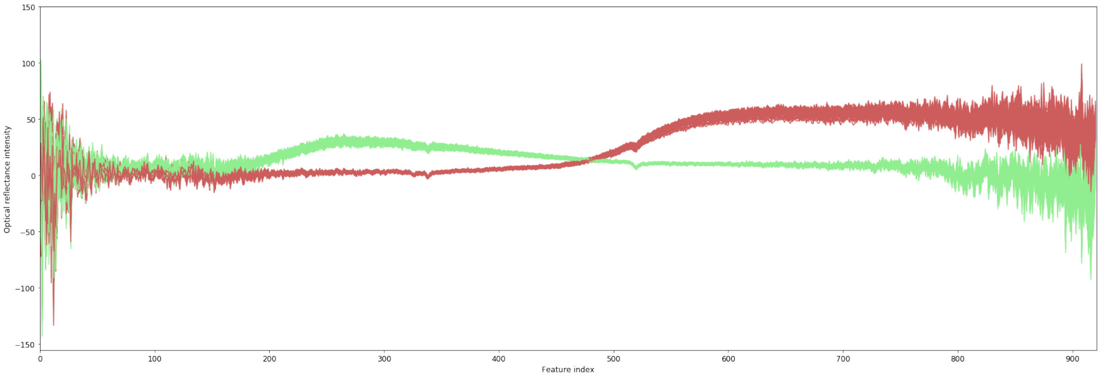

**Figure 1:** Input feature visualization for binary classification task. Red and green predicting features are indicated by colour.

Fig.1 shows that there is a clear distinction between red and green colours in terms of features. From the same figure it can be said that feature values between 420 and 450 wavelengths are more or less shared between both, red and green colours. Around 600 wavelength both colour features overlap. Similarly, towards the final features similar observations can be made. The graph insights suggest that overlapping features are not great for determining the class because a feature value can be shared by both colours therefore reducing possibility of determining the right colour. However, a single feature that is around 530 or 650 should be good enough to determine the class. From fig.1 it is clear that intensity at those wavelengths are distinct to each colour.

The hypothesis therefore is that any feature that distinguishes the two colours at particular wavelength will be good enough to determine the class. According to the fig.1 there are many of these features: at wavelengths 500-to-580 and 610-to-720. Therefore, it one feature should be enough to determine the class.

## 2.3 Preparing inputs and Choosing features

To choose an appropriate feature an experiment was conducted. A single feature could possibly determine a class therefore a training was done using every single feature separately to determine which one is the most accurate. Fig.2 shows the results. Y axis represents accuracy score and X axis represents a single feature 0-to-920.

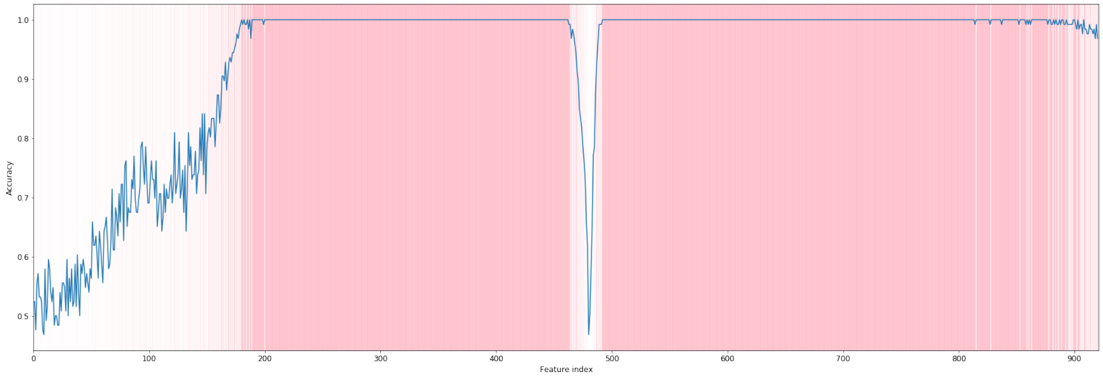

**Figure 2:** One feature accuracy scores for binary classification.

Pink areas cover features that perform very well. It can be seen that features indexed 200-470 and 490 to almost the end of 900 perform with 100% accuracy. Therefore, any feature that falls in between these boundaries will perform very well. From fig.**??**

From the graphs, large number samples also should not be necessary because there are no noticeable outliers.

## 2.4 Selecting and Training classification model

### 2.4.1 Linear logistic regression

For binary classification task a simple linear logistic regression model was chosen from sklearn. As expected, the accuracy score is 1.0 with training data. Similar observations are made by using other features too.

Once might think that this is caused by over-fitting however from the feature analysis the results make sense. Therefore, scaling which is normally incorporated when over-fitting is suspected was not applied in this situation.

## 2.5 Evaluating model performance

### 2.5.1 Linear logistic regression

For evaluating performance

Tbl.1 shows results when running the model on test set using four different features. The accuracy score is 100% in each case as expected. Form the results, it can be said that model works very well on test data when testing on four different occasions, each with different feature: 300, 400, 600 or 700.

| Feature Index | Accuracy |
|---|---|
| 300 | 1.0 |
| 400 | 1.0 |
| 600 | 1.0 |
| 700 | 1.0 |

**Table 1:** Accuracy score when testing linear logistic regression model on testing set.

### 2.6 Result discussion

Results seem to be in according to the observations and hypothesis made during data investigation and visualization steps.

## 3 Multi-Class Task

### 3.1 Cleaning Data and Feature Extraction

Similarly to binary task, no cleaning or feature extraction was required.

### 3.2 Data Analysis and Visualization

Similarly, fig.3 shows different colour reflectance intensities for five different colours. Similar observation can be seen here too. However, due to larger number of colour classes some seem to overlap slightly.

From fig.**??** it can be seen that red and pink colour intensities over different wavelengths are very similar. Green and blue are also similar. Yellow on the other hand is very distinct. Just from looking at the graph it can be seen that there are not many features that

| No | Feature Indexes | Accuracy |
|---|---|---|
| 1 | 421 | 0.810 |
| 2 | 421, 429 | 0.851 |
| 3 | 421, 429, 250 | 0.997 |
| 4 | 421, 429, 250, 251 | 0.997 |
| 5 | 421, 429, 250, 251, 86 | 1.0 |
| 6 | 421, 429, 250, 251, 86, 586 | 1.0 |
| 7 | 421, 429, 250, 251, 86, 586, 88 | 1.0 |
| 8 | 421, 429, 250, 251, 86, 586, 88, 66 | 1.0 |
| 9 | 421, 429, 250, 251, 86, 584, 88, 66, 586 | 1.0 |
| 10 | 421, 429, 250, 251, 86, 584, 88, 66, 586, 914 | 1.0 |

**Table 2:** My caption

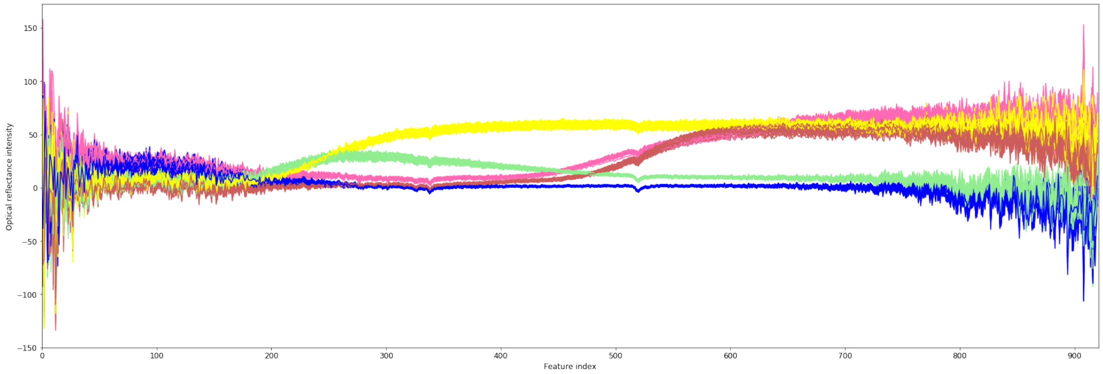

**Figure 3:** Input feature visualization for multi-class task. Five colour predicting features are indicated by corresponding colour.

### 3.3 Preparing Inputs and Choosing Features

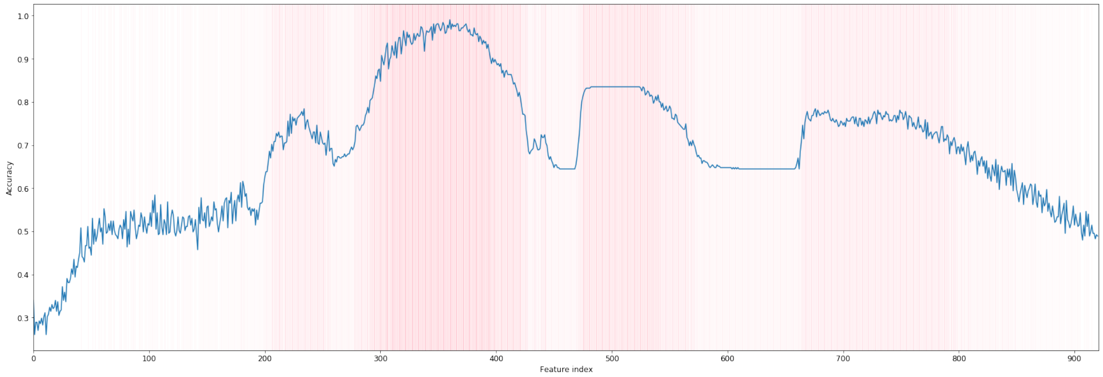

**Figure 4:** One feature accuracy scores for multi-class classification.

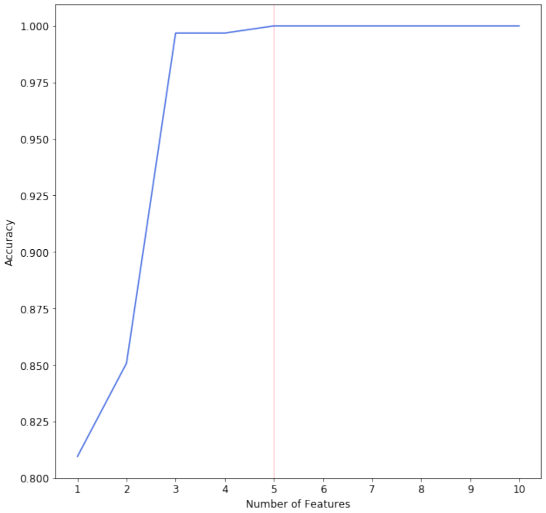

**Figure 5:** Accuracy depending on number of features extracted by REF.

### 3.4 Selecting and Training Classification Models

### 3.5 Evaluating and Comparing Model Performance

### 3.6 Result Discussion

## 4 Conclusion

# A Appendices

## A.1 Binary task results

| No | Class | No | Class |
|---|---|---|---|
| 1 | 1 | 11 | 0 |
| 2 | 1 | 12 | 1 |
| 3 | 0 | 13 | 1 |
| 4 | 0 | 14 | 1 |
| 5 | 1 | 15 | 0 |
| 6 | 0 | 16 | 1 |
| 7 | 0 | 17 | 1 |
| 8 | 0 | 18 | 0 |
| 9 | 1 | 19 | 1 |
| 10 | 0 | 20 | 0 |

**Table 3:** Binary task result for *XToClassify*.

## A.2 Multi-class task results