

CS5014 - Machine Learning

P2 - Classification of object colour using optical spectroscopy

Report

140014952

April 21, 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 */multi-classML/* 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.

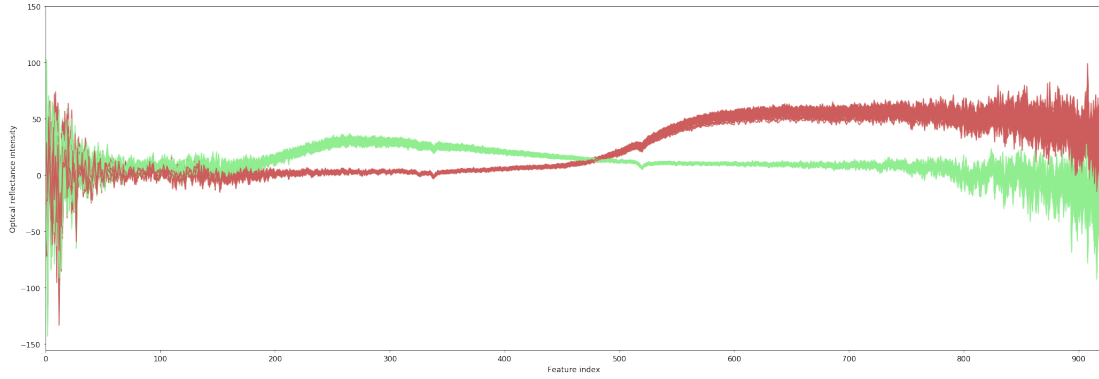


Figure 1: Input feature visualization for binary classification task. Red and green predicting features are indicated by colour.

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 sep-

arately 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.

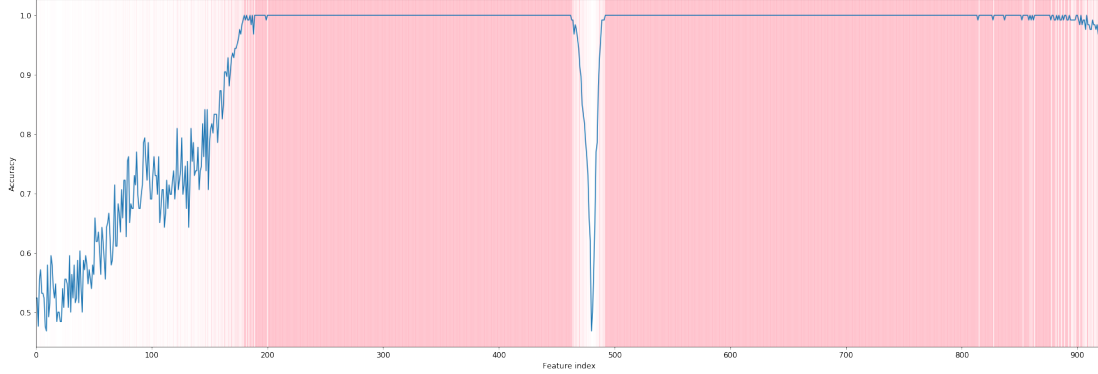


Figure 2: One feature accuracy scores for binary classification.

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 necessary.

3.2 Data Analysis and Visualization

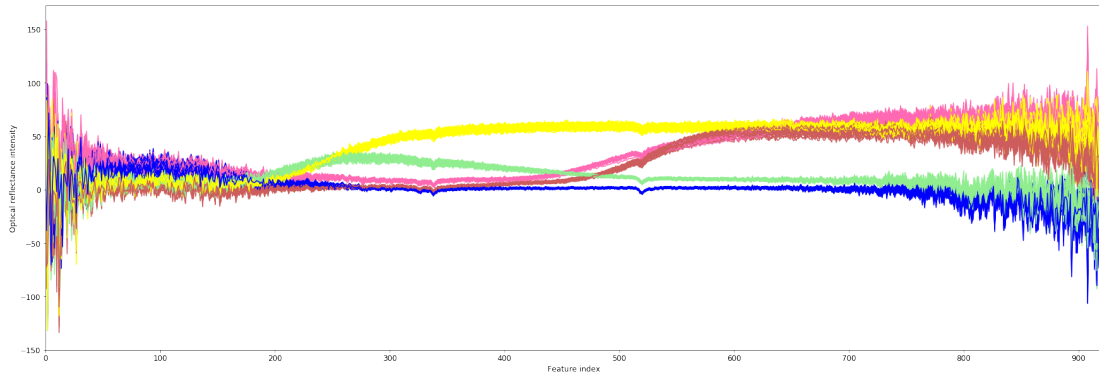


Figure 3: Input feature visualization for multi-class task. Five colour predicting features are indicated by corresponding colour.

Fig.3 shows different colour reflectance intensities for five different colours. Green and red just like in binary task are distinct and the same patterns can be recognized in multi-class task. However, due to larger number of colour classes some seem to overlap slightly. From fig.3 it can be seen that red and pink colour intensities over different wavelengths are very similar. Yellow on the other hand has very distinct reflectance measures. Just from looking at the graph first two hundred features do not distinct different colours.

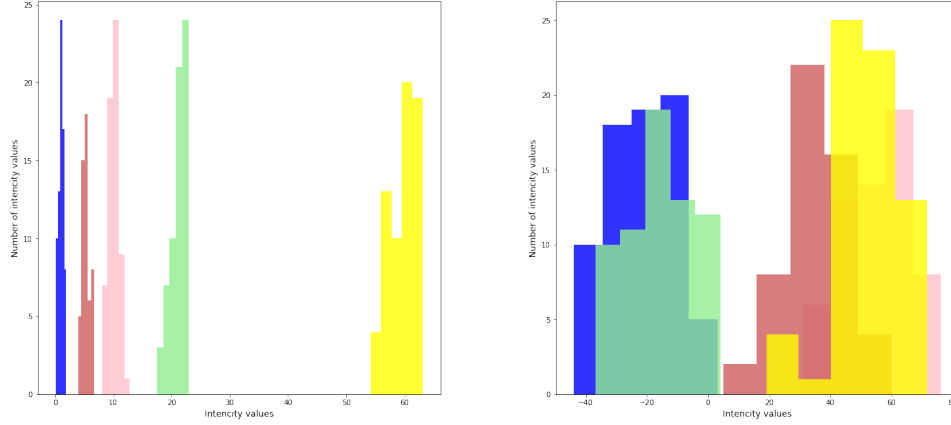


Figure 4: On the left - feature 400, on the right - feature 900.

Blue and green seem to have similar reflectance towards the end. Pink, yellow and red also share similar patterns.

From the same graph it can also be seen that some features distinct colours well. For instance, feature indexed 420 is likely to perform well. Since there are almost no overlapping feature values for any of the colours.

For instance, when feature 400 is selected histogram in fig.?? shows that the samples are clearly distinctly distributed. Whereas, feature 900 in fig.?? shows that green and blue overlap slightly, red pink and yellow follow the pattern. This also suggest that red and green will be most likely to be mixed. Pink, red and yellow mixed together too.

3.3 Preparing Inputs and Choosing Features

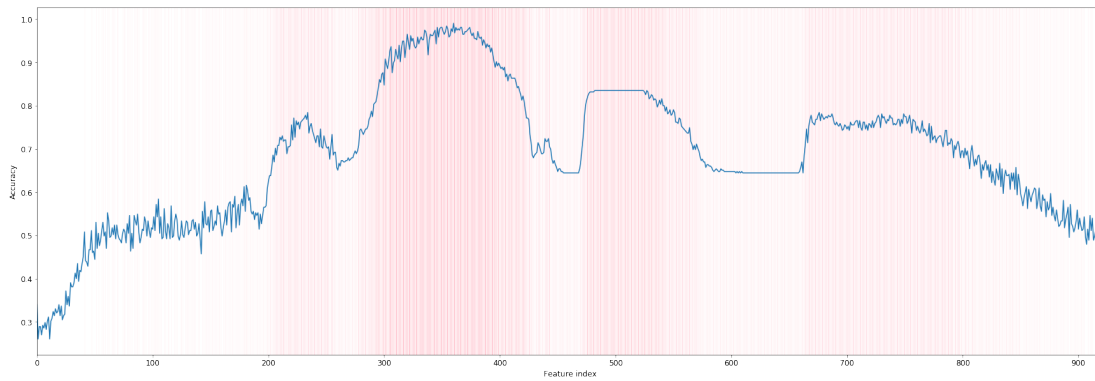


Figure 5: One feature accuracy scores for multi-class classification.

Just like in binary task, in order to investigate single feature performance logistic regression model was trained on every feature separately. Fig.5 demonstrates the results.

From the graph it can be seen that some features perform better than other. As per previous input analysis, best accuracy feature are 300-to-400, providing >85% accuracy on training set however not reaching 100%. Logistic regression model performance is clearly not as great as in binary task. However, still very good. To investigate further and see whether different input combinations provide better accuracy results recursive feature elimination (RFE) algorithm was used from sklearn. The idea behind this selection algorithm is to choose the best features by considering lesser number of them. Starting from the full input set the predictor recursively tries smaller subsets of inputs [1]. The result is mask of true and false values that can be used to choose the best feature.

For RFE experiment 1-to-10 features were chosen and accuracy scores for each selection were recorded in table .2.

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: Accuracy table for different features.

This table shows how many features is the optimal number for maintaining good accuracy score. With feature 421 accuracy score is 0.81 which is already very good. As the number of features is increased the accuracy score also becomes higher. Three features - 421, 429 and 250 are enough to achieve almost 100% accuracy. Later, as the number of features is more than four the accuracy score is stable and is 1.0.

Interestingly, the RFE did not choose 350 feature although as the previous experiment suggests it provides over 0.9 accuracy score on its own. This might be because RFE chooses features by eliminating a subset of features - large subset of features means better accuracy however elimination does not diverge to a single optimal feature.

After analysis, a subset of three features was chosen: 421, 429 and 250. This subset resulted in 0.997 accuracy.

3.4 Selecting and Training Classification Models

3.4.1 Linear logistic regression model

Accuracy 0.997

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	72
1	0.98	1.00	0.99	65
2	1.00	0.98	0.99	60
3	1.00	1.00	1.00	52
4	1.00	1.00	1.00	66
avg / total	1.00	1.00	1.00	315

Table 3: Logistic regression report results.

3.4.2 Linear vector support classifier

Accuracy 1.0.

Class	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	72
1	1.00	1.00	1.00	65
2	1.00	1.00	1.00	60
3	1.00	1.00	1.00	52
4	1.00	1.00	1.00	66
avg / total	1.00	1.00	1.00	315

Table 4: Linear support vector classifier report results.

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 5: Binary task result for $XToClassify$.

A.2 Multi-class task results

References

- [1] Sklearn RFE [Accessed 21/04/2018]
http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html