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 */multiclassML/* directories.

As later sections suggest, the number 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 was 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 was implemented to ensure that the samples are fully prepared.

2.2 Data analysis and Visualization

To visualize training data X, each of the features was plotted in fig.1. 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 class. x axis is the index of the feature - each feature is for different wavelength.

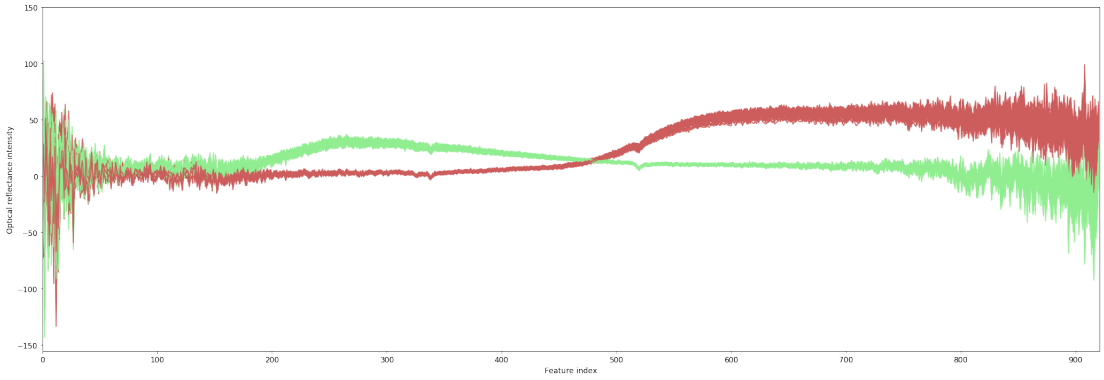

Figure 1: Input feature visualization for binary classification task. Red and green predicting samples are indicated by colour.

Fig.1 shows that there is a clear distinction between red and green colours in terms of sample values provided in the training set. From the same figure it can be said that feature values between 0 and 100 are more or less shared between both, red and green colours. Around 470 both colour feature values 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 300 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 in features 200-to-420 and 520-to-800. Therefore, 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.

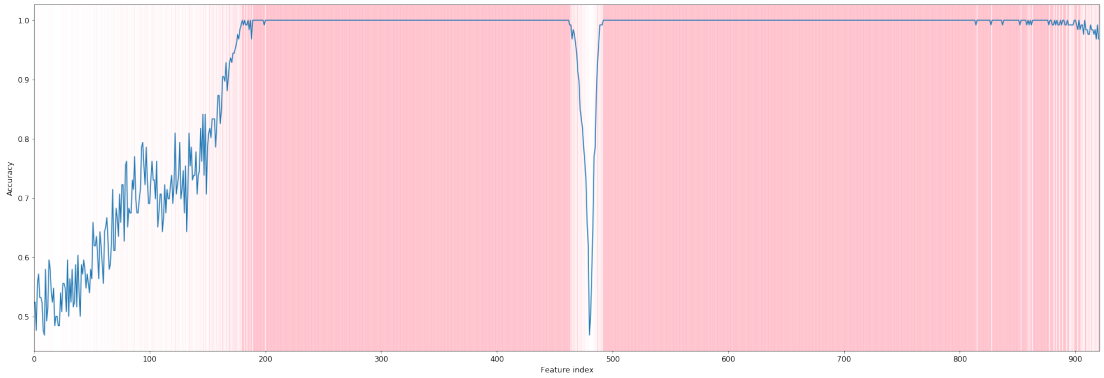

Figure 2: One feature accuracy scores for binary classification task.

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. The results graph very well fits the original feature graph in fig.1 and shows that most feature values are far away for both classes apart from beginning and middle. 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 when testing with training data only. Similar observations are made by using multiple other features as shown before. This was achieved in the experiment described before. The 1.0 accuracy also means that precision and recall also 1.0 - all classes were predicted correctly.

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.

Since a simple linear logistic regression models seemed to work well no other models were investigated.

2.5 Evaluating model performance

2.5.1 Linear logistic regression

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 during four different occasions, each with different feature: 300, 400, 600 or 700 - shown in table 1.

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. The results suggest that both object colours can be distinguished accurately by using a single feature on certain wavelength.

3 Multi-class task

3.1 Cleaning data and Feature extraction

Similarly to binary task, no cleaning or feature extraction was necessary. Null value function was applied on this data set too however no samples were matched. Mean, minimum and maximum values seem to be reasonable considering the features.

3.2 Data analysis and Visualization

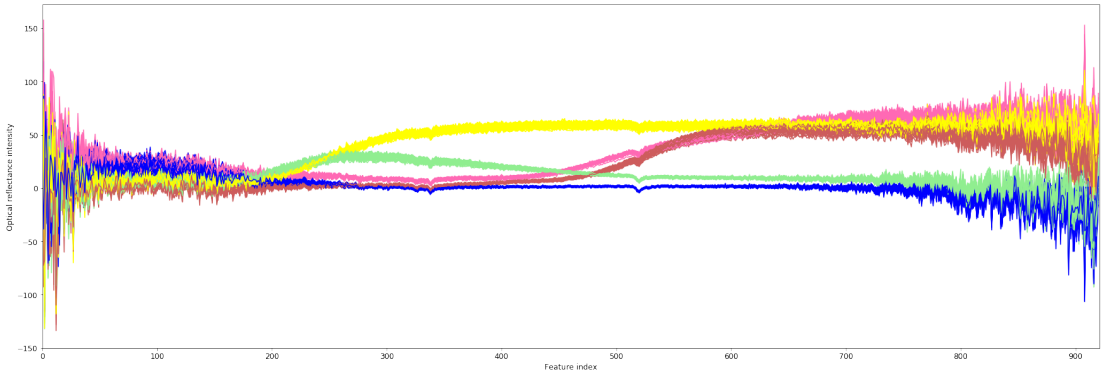

Figure 3: Input feature visualization for multi-class task. Five colour predicting samples are indicated by corresponding colour.

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. 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.4 shows that the sample values are clearly distinctly distributed. Whereas feature 900 in fig.4 shows that green and blue overlap, red pink and yellow follow the pattern. This and the original graph also suggest that blue and green will be most likely to be mixed. Pink, red and yellow mixed together too.

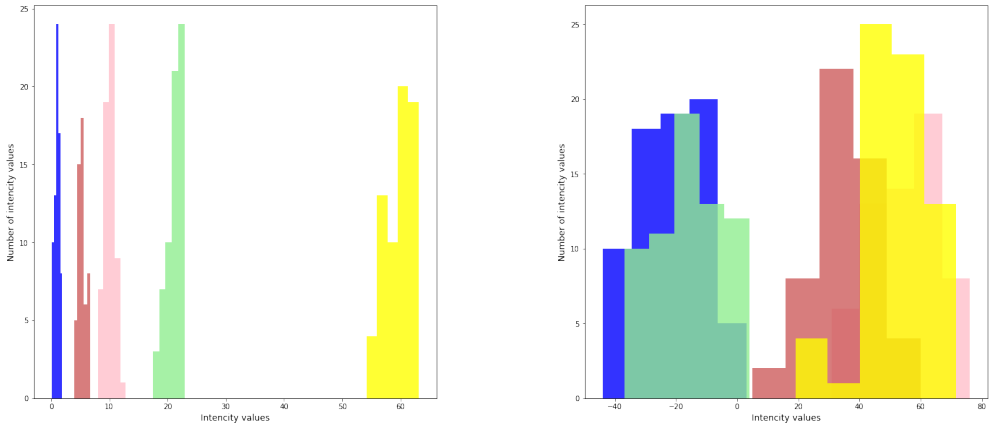

Figure 4: On the left - feature 400, on the right - feature 900.

3.3 Preparing inputs and Choosing features

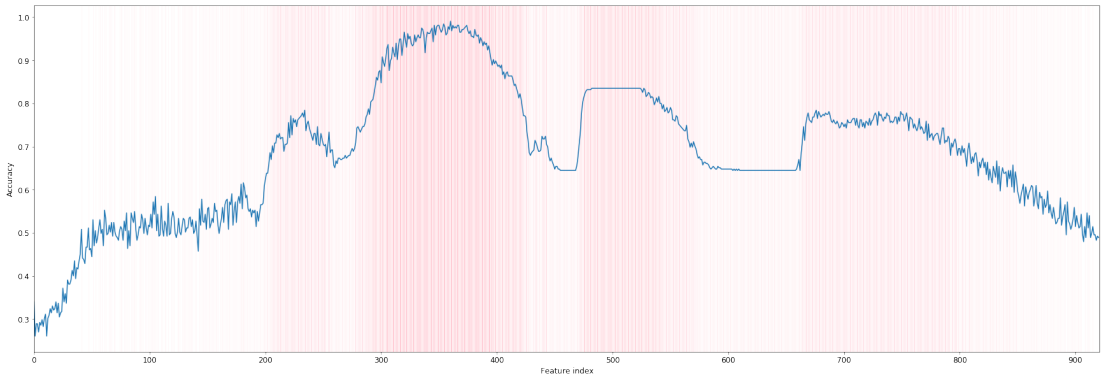

Figure 5: One feature accuracy scores for multi-class classification.

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 others. As per previous input analysis, best accuracy features are 300-to-400, providing >85% accuracy on training set however not reaching 100%. In particular, feature 350 provides over 0.9 accuracy score. 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 features.

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

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 1.0 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 on training set when experimenting.

3.4 Selecting and Training classification models

3.4.1 Linear logistic regression model

When using linear regression model with the three features accuracy of 0.997 is achieved on training set. Table 3 holds classification report results. Each class has a precision, recall, f1-score and support measures.

It can be seen that only green and pink have precision or recall other than 1.0. Precision indicates what part of positive classifications were actually correct whereas recall what part of actual positives were identified correctly. F-1 score is a weighted average of both precision and recall. Support identifies the number of actual classifications for particular class.

From the totals it can be said that the linear logistic regression model seems to be a very good model for this particular problem.

Class	Class No	Precision	Recall	F1-score	Support
Blue	0	1.00	1.00	1.00	72
Green	1	0.98	1.00	0.99	65
Pink	2	1.00	0.98	0.99	60
Red	3	1.00	1.00	1.00	52
Yellow	4	1.00	1.00	1.00	66
	avg / total	1.00	1.00	1.00	315

Table 3: Logistic regression report results on training data.

3.4.2 Linear vector support classifier

To experiment further, a linear vector support classifier was applied on the same three features. Accuracy of 1.0 was achieved. This is slightly better than the logistic regression model results.

From the table 4 it can be observed that the averages are just like in logistic regression model. However, vector support model seems to perform better with colours pink and green achieving 1.0 in precision and recall.

Class	Class No	Precision	Recall	F1-score	Support
Blue	0	1.00	1.00	1.00	72
Green	1	1.00	1.00	1.00	65
Pink	2	1.00	1.00	1.00	60
Red	3	1.00	1.00	1.00	52
Yellow	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 on training data.

The results table tells that there are more blue colour samples in the training set than any other colour. Red has the smallest number of samples. This would normally indicate that the model could possibly perform better when classifying certain classes however in this case this does not apply.

3.5 Evaluating model performance on testing set

From the two models chosen support vector classifier performs slightly better therefore is chosen for testing on test data set. The three same input features are used.

The results of the model performance are recorded in table 5. The same observations can be made as with training data. The chosen futures and the VSC seems to provide 1.0 accuracy. The results tell that all of the test samples were classified correctly. Due to used cross validation and input observations this is not likely to be caused by over-fitting.

Class	Class No	Precision	Recall	F1-score	Support
Blue	0	1.00	1.00	1.00	18
Green	1	1.00	1.00	1.00	25
Pink	2	1.00	1.00	1.00	30
Red	3	1.00	1.00	1.00	38
Yellow	4	1.00	1.00	1.00	24
	avg / total	1.00	1.00	1.00	135

Table 5: Linear support vector classifier report results on test data.

3.6 Result discussion

One might think that the results are too good to be true and perhaps over-fitting has occurred. However, from the input inspection this should not be the case - the data provided seems to be excellent for classifying especially when extracting the right features which was achieved successfully using the RFE approach.

It was first considered to apply scaling techniques but the final models performs very well on the test data too. Therefore, this step was skipped. Also, data is already to some extend scaled the values are in range from -150 to 150. However, I do understand that it is good practice in general to apply scaling and normalization techniques. I believe in this case it is not necessary.

4 Conclusion

This practical submission contains two python notebooks: one for each task. As per practical specification, appendices section of this report contains two tables for each of the task final predictions for *XToClassify*. Directories */binaryTask/* and */multiClassTask/* contain files with the same results.

Data was clean and ready for machine learning. Overall, this submission investigated data sets through different visualization techniques, then it applied multiple machine learning models and observed and reported results using common classification measures in results tables. From binary and multi-class classification results both models perform very well. Due to previously stated facts this is unlikely to be due to over-fitting.

Because of good results I did not explore more models. The two simple logistic and VSC models were well-suited for these two tasks.

From the data observation the models should perform well on a very small training sample too e.g, five samples - one for each of the class for multi-class task and two for binary classification. If had more time, this I believe could be proved too.

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 6: Binary task result for *XToClassify*.

A.2 Multi-class task results

No	Class	No	Class	No	Class	No	Class	No	Class
1	2	11	4	21	4	31	1	41	0
2	0	12	3	22	1	32	1	42	0
3	2	13	3	23	2	33	3	43	1
4	0	14	2	24	1	34	1	44	1
5	0	15	0	25	4	35	2	45	3
6	0	16	4	26	2	36	4	46	0
7	2	17	2	27	3	37	2	47	3
8	0	18	4	28	2	38	3	48	4
9	4	19	3	29	0	39	1	49	3
10	1	20	3	30	1	40	4	50	4

Table 7: Binary task result for *XToClassify*.

References

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