CS5014 Machine Learning

Assignment 2: Classification of object colour using optical spectroscopy

University of
St Andrews

Student ID: 170009479

Contents

1 Introduction

This assignment aims at predicting colours based on the given dataset through using classification model. To implement this, firstly, the classification model is produced by training data using proper supervised learning algorithms. Secondly, those built models are used to predict colours. In the whole processes, it also involves some steps, such as, data loading, data cleaning, feature selection and algorithms selection and avoiding underfitting and overfitting. Considering colour classification is a problem of predicting classes, therefore proper supervised learning algorithms includes logistic regression, neural networks and support vector machine.

In this report. Firstly, some preparation work, loading data and dividing data into training dataset and validation are given. Secondly, the steps of cleaning data and the principles of select features are described. Thirdly, data normalisation is done before training a model. Then, the reason of choosing logistic regression, neural networks and support vector machine are illustrated. Last but not least, the prediction results and the qualification of models are evaluated by two parts, one is for binary classifiers, and another is for multi-class classifiers. Note that appendix 1 and 2 shows the prediction results for XtoClass.cvs.

2 Load data

In both binary.zip file and multiclass.zip file, X.csv file contains different samples that are the corresponding optical reflectance intensity of various wavelengths. Concrete wavelengths are referred in Wavelength.csv, and they are regard as features or headings of the dataset in X.csv. Furthermore, since the numbers in y.csv represent colours of different samples, they are used to label the samples. Data in XtoClass.csv is unlabelled, which is test dataset. Therefore, X.csv, y.csv and XtoClass.cvs are loaded as inputs, outputs and test set respectively to the python file through using pandas library.

3 Split data into training dataset and test dataset

Data should be split randomly, which ensures training the data without bias. Theoretically, the larger we make our training set is, the better model we are able to learn. Generally, in supervised learning, a labelled dataset is divided into trained data (70%), validation (30%). Trained dataset is used to train a model that fits the data of the training set. Validation is used to check if the trained model can perform well when predicting the result of validation data. If the empirical loss is low but the generalised loss is high (namely, the model fits training data well but fits the validation data badly), it means the model is overfitting. Conversely, the model is underfitting. Whatever the model is overfitting or underfitting, it is unsatisfied. Then, the dataset need to be randomly split again, train a new model, and evaluate model by using validation data. These steps are repeated until getting a satisfied model (low empirical loss and generalised loss). In this assignment, trained data is split based on the choice of the user, to test if different volumes of dataset can influence the performance of the model. The user can choose 30%, 70% or 90% dataset as training set. It is predicted that 30% dataset as training data to train a model could cause underfitting, 90% could cause overfitting, 70% could be proper. Besides, data in XtoClass.cvs is unlabelled, and its results are required to be predicted, hence, it acts as a test set.

4 Clean the data and select features

It is necessary to clean the training dataset before training a model. Common problems in data sets includes missing data, repetitive data, noise, inconsistent data and high dimensionality. The corresponding treatment is as follows:

(1) Missing data

Check if the training dataset is complete, i.e. no blank in the training set. If a small numbers of data are lost, just simply remove them. If there are many missing data in a column (i.e. feature/attribute), then the data of the attribute can be removed because of having no reference value. Sometimes, the missing data can be made up through the data of other attributes, for example, a postal code can be deduced by an address.

In this assignment, the training dataset is complete, thus there is no sample or the data under a specific attribute that need to be deleted.

(2) Repetitive data

Sometimes duplicated data are useless, which need to be deleted. However, in some cases, for example, counting the word frequency, duplicated data should not be removed.

In this assignment, on the one hand, there is no duplicated data. On the other hand, even if there are a few repeated data, they are still meaningful and no need to be deleted, because they response different colours through different reflective intensity of different wavelength.

(3) Inconsistent data

Inconsistent data includes the inconsistency of different attribute values, inconsistency of different unit of the same attribute, or inconsistency of different data type of the same attribute. For example, when expressing the magnitude of distance, some data use meters as the unit, some use kilometre as the unit. When meeting inconsistency, the data need to be pre-processed before using them.

In this assignment, there is no way of examining whether there is such a problem, hence, we treat them as qualified.

(4) Noise

In this assignment, noise includes unreasonable data or the data that have severe fluctuation within nearby wavelengths. Unreasonable data are those that are beyond the range, i.e. the optical reflectance intensity being lower than zero.

The whole dataset of X.csv is randomly split into training set, and different training sets present similar properties. Figure 1 shows the data distribution of a training set of binary classes, it is noticed that many reflective intensities of wavelengths (about smaller than 550 and bigger than 690) are negative, therefore the features (i.e. wavelength) carrying many negative data need to be removed because of out of range. Moreover, those data also meet the second conditions, namely, having severe fluctuation within nearby wavelengths, thus the corresponding features should be ignored.

Same criteria are used to check if there is noise in the training data of multiple classes. As is shown in figure 2, noise are mainly distributed in the wavelengths than are smaller than 550 and bigger than 690. Therefore, wavelengths in that range are removed. Besides, note that although for blue colour, there are some reflectance intensity of wavelength between 550 and 690 are lower than 0, I decided to remain them and their feature. This is because there are some typical, characteristic, representative and regular data that may be helpful to train a model with only using a small part of features. For example, the corresponding reflectance intensity of the wavelength about 610.

After the process of handling noise, some features are removed, which can also be considered as doing the preliminary feature selection.

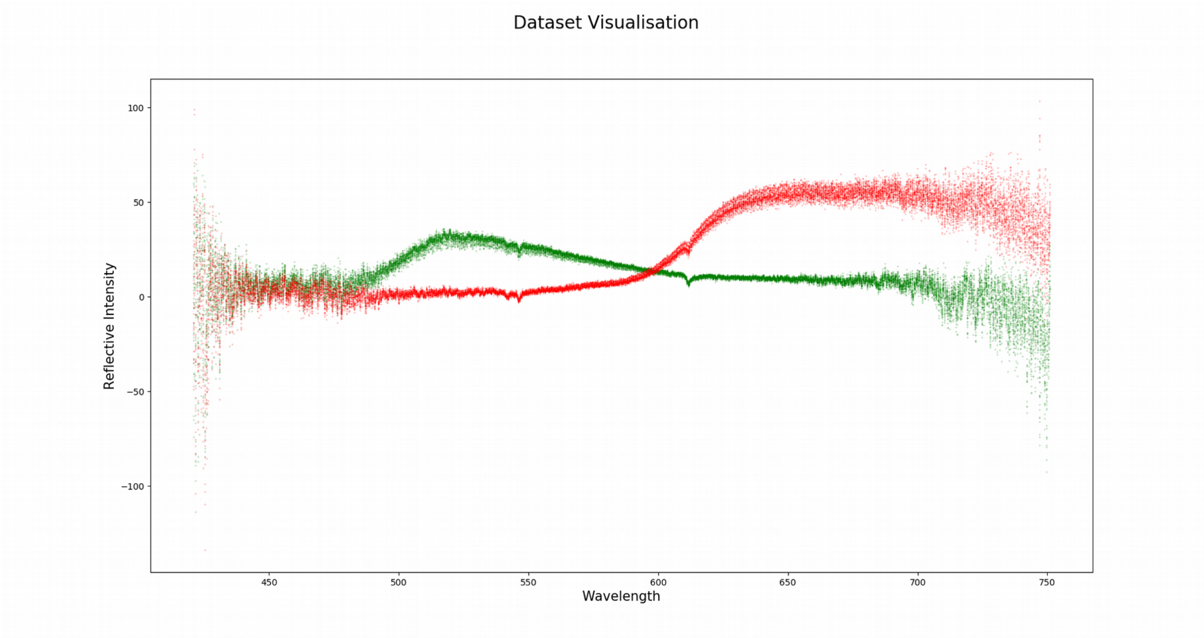

Figure 1

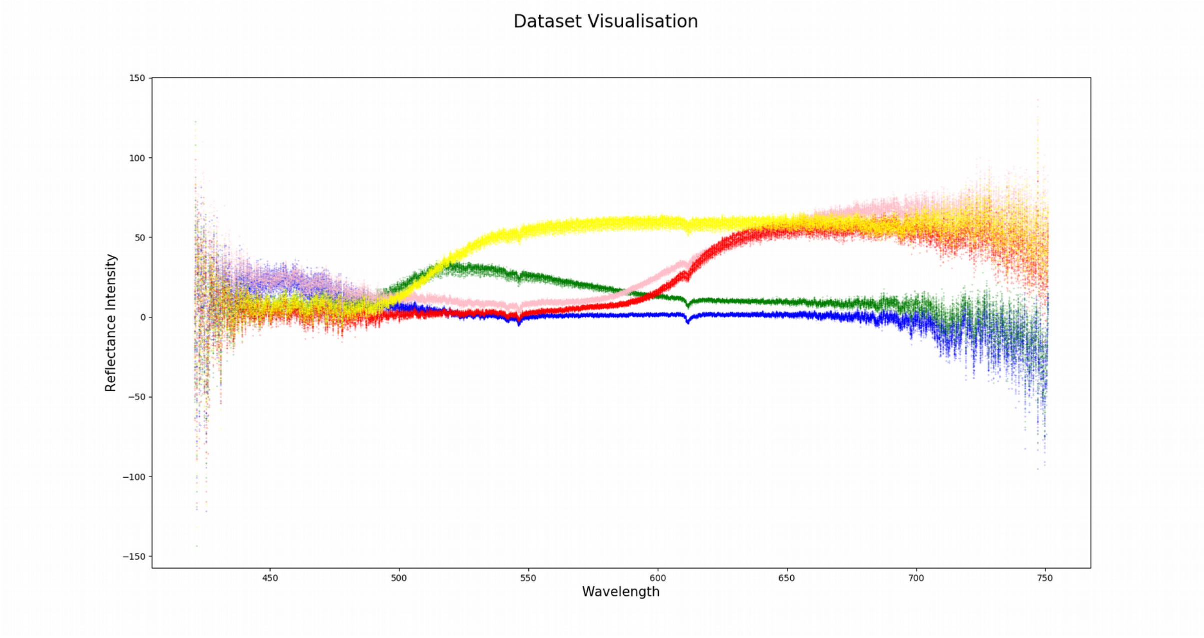

Figure 2

(5) High dimensionality

Resolving the problem of high dimensionality also involves data selection. In this assignment, originally, there are 921 different wavelengths/features, which is much more than the numbers of samples. Intuitively, there are too many features. After removing less than half part of features through avoiding noise, the number of features is still substantial. Since technically, reducing the number of features helps to improve the efficiency of building a model, it is necessary to try to make features concise.

According to figure 1, in the wavelengths less than 500, there is a great deal of noise at the beginning and then some cross area (i.e. reflectance Intensity for two colours being close), therefore, these features are useless. In the wavelengths higher than 690, the fluctuation of the reflectance intensity for both green and red is visible, thus, theoretically, these wavelengths are regard as noise and should be removed. However, the interesting thing is reflectance intensity of different colours fluctuated in different range, so I suppose using wavelengths higher than 690 as features can train a satisfied model. Hence, I will use these features to do an experiment later and test if my prediction is right. Besides, undoubtedly, a good model can be trained using the wavelengths between 500 to 690, because the differences of the value of reflectance intensity in this interval for different colours are distinct, and the tendencies are opposite. Thus, I also use these features to train a model.

According to figure 2, in the wavelength between 500 to 550, there are coincident areas (green and yellow, blue and red), and the values of the reflectance intensity of various colours are close (all five colours in the beginning and blue and red in the end), which means it is relatively difficult to distinguish colours based on these features. Hence, these features are excluded. The Similar problem occurs in the wavelength range between 630 to 690, the values of the reflectance intensity of yellow, pink and red colours are overlapped. Thus, these features are deleted. The left features (550-630) can reflect different colours well, because there are relatively big differences between the values of the reflectance intensity of different colours in this interval. Moreover, these values for different colours display distinctly different tendency with the increase of the value of the wavelengths. Hence, the wavelengths between 550 and 630 will become the features used to training the model.

5 Data normalisation

Data normalisation is to scale the data so that they fall into a small specific range, for example from 0 to 1. There are two main advantages of doing this step. Firstly, it can increase the speed of the convergence of a model. Secondly, it enhances the comparability of value between different features, which can greatly improve the accuracy of the classifier. Therefore, usually, data need to be normalised before training a model.

According figure 1 and figure 2, the ranges of the reflectance intensity of different wavelengths (i.e. features) are different, hence data normalisation is necessary. Here, I just use a commonly used type of normalisation, which maps the data between 0 and 1.

6 Select 3 classification models

Machine learning problems mainly include two categories, namely, value prediction and classification prediction. According to the requirement of this assignment, concretely, the target is to predict colours, which is categorised as classification prediction problem. To solve this kind of problems, both supervised learning algorithms and unsupervised learning algorithms can be used to train a model. The biggest distinguish between these two types of algorithms is whether using a labelled dataset or not. The former is based on the labelled data, and the latter learns from unlabelled data and find some structure from the data. Since, the offered files include labelled data, I choose supervised learning algorithms to train a model. Commonly used supervised classification algorithms include logistic regression, neural networks and support vector machine, which will be also used in latter experiments.

(1) Logistic regression(LR)

In many cases, a classification problem can be regarded as a special regression problem of which the target is to predict a small number of discrete values. Linear regression cannot properly solve the classification problems because a classification problem is not a linear function. However, logistic regression can achieve the effect of differentiating colour types.

For binary classification, i.e. distinguishing green and red, logistic regression gives results directly through comparing the value of h(x) with 0.5. To apply logistic regression to solve multi class classification problems, i.e. distinguishing green, blue, pink, yellow and red, one-vs-all classification idea is used. One-vs-all idea separates multi class classification problems to binary classification problems, and the number of classifiers is equal to the number of the classes.

(2) Neural networks(NN)

Neural networks is also a classification algorithm. In a neural network, there are an inputs layer, hidden layers, biases and an output layer. The neurons of inputs correspond to the different features, and the numbers of neurons of inputs layer and features are equal. The numbers of hidden layers and biases are modified depending on the quality of the trained model. Output layer give the prediction result.

In this assignment, I use logistic function as activation function because it is classification problem. Comparing with logistic regression, neural network is suitable to be applied when the number of features is big. Due to the tens of features will be used, there is no doubt that using neural networks is a rational choice to train a model.

(3) Support vector machine(SVM)

Support vector machine is often used in supervised learning to solve classification problem. It exhibits many unique advantages in solving small sample, nonlinear and high-dimensional pattern recognition, and can be applied to other machine learning problems such as function fitting.

In this assignment, there are hundreds of samples but nearly a thousand of features (i.e. dimensions). Plus, it is a classification problem with labelled data. Hence, SVM is eligible to be used.

7 Demonstrate and evaluated the performance of the model

(1) Binary class classification

There are 3 types of regression models have been applied in this 2 colours classification problem, including logistic regression, neural network and support vector machine. They train models based on 4 different feature sets and 3 different volumes of training data set. All experiments are listed in the figure 3, and their results are shown in figure 4.

Models are evaluated by 3 criteria, comparing the speed of building a model, and comparing the accuracy score and confusion matrix of predicting the result of validation set in different models. Speed of building a model is measured by calculating the time that it takes to build a model, and the unit of time is second. We usually want to spend less time building a model. Since the consuming time can also influenced by the performance of the computer at that time, I run 36 models one by one within a certain continuous period of time and do not add other new affairs in the halfway to avoid effects produced by the differences of performances. Accuracy score signifies the accuracy rate of the prediction (range: 0-1), which is achieved through calculating the number of correctly predicted samples divided by the total number of samples. The higher the accuracy is, the better the model is. Confusion matrix is usually used to show the performance of the model, and the format of the result is '[[true negative, false negative] [false positive, true positive,]]. Green is represented by 0 (i.e. false), and red is represented by 1 (i.e. true). In this scenario, true negative indicates the count that both prediction and actual result are green, false negative indicates the count that the prediction is green but the actual result is red, true positive indicates the count that both prediction and actual result are red, and false positive indicates the count that prediction is red but the actual result is green. Therefore, a good model is expected having 0 false negatives and 0 false positives.

	model	features_selection	training_set/whole_dataset(%)

Experiment1	logistic regression	all wavelength	30
Experiment2	logistic regression	all wavelength	70
Experiment3	logistic regression	all wavelength	90
Experiment4	logistic regression	Wavelength: 420-470	30
Experiment5	logistic regression	Wavelength: 420-470	70
Experiment6	logistic regression	Wavelength: 420-470	90
Experiment7	logistic regression	Wavelength: 576-624	30
Experiment8	logistic regression	Wavelength: 576-624	70
Experiment9	logistic regression	Wavelength: 576-624	90
Experiment10	logistic regression	Wavelength: 705-751	30
Experiment11	logistic regression	Wavelength: 705-751	70
Experiment12	logistic regression	Wavelength: 705-751	90
Experiment13	neural network	all wavelength	30
Experiment14	neural network	all wavelength	70
Experiment15	neural network	all wavelength	90
Experiment16	neural network	Wavelength: 420-470	30
Experiment17	neural network	Wavelength: 420-470	70
Experiment18	neural network	Wavelength: 420-470	90
Experiment19	neural network	Wavelength: 576-624	30
Experiment20	neural network	Wavelength: 576-624	70
Experiment21	neural network	Wavelength: 576-624	90
Experiment22	neural network	Wavelength: 705-751	30
Experiment23	neural network	Wavelength: 705-751	70
Experiment24	neural network	Wavelength: 705-751	90
Experiment25	support vector machine	all wavelength	30
Experiment26	support vector machine	all wavelength	70
Experiment27	support vector machine	all wavelength	90

Experiment28	support vector machine	Wavelength: 420-470	30
Experiment29	support vector machine	Wavelength: 420-470	70
Experiment30	support vector machine	Wavelength: 420-470	90
Experiment31	support vector machine	Wavelength: 576-624	30
Experiment32	support vector machine	Wavelength: 576-624	70
Experiment33	support vector machine	Wavelength: 576-624	90
Experiment34	support vector machine	Wavelength: 705-751	30
Experiment35	support vector machine	Wavelength: 705-751	70
Experiment36	support vector machine	Wavelength: 705-751	90

Figure 3

	Time (secondes)	Accuracy	Confusion matrix
Experiment1	0.00477096	1	[[59 0][0 67]]
Experiment2	0.006492647	1	[[23 0][0 32]]
Experiment3	0.011899111	1	[[6 0][0 12]]
Experiment4	0.001101551	0.873015873	[[62 0][16 48]]
Experiment5	0.001568382	0.927272727	[[27 1][3 24]]
Experiment6	0.002168515	1	[[11 0][0 7]]
Experiment7	0.001153374	1	[[60 0][0 66]]
Experiment8	0.001802823	1	[[29 0][0 26]]
Experiment9	0.002279612	1	[[12 0][0 6]]
Experiment10	0.001227442	1	[[63 0][0 63]]
Experiment11	0.001686473	1	[[28 0][0 27]]
Experiment12	0.002072207	1	[[8 0][0 10]]
Experiment13	0.572988075	1	[[59 0][59 0]
Experiment14	0.269266913	1	[[32 0][0 23]]
Experiment15	0.083130101	0.333333333	[[6 0][12 0]]
Experiment16	0.236812038	0.904761905	[[63 0][12 51]]
Experiment17	0.175157595	0.945454545	[[22 2][1 30]]
Experiment18	0.222436259	0.888888889	[[8 2][0 8]]
Experiment19	0.111332078	1	[[64 0][0 62]]

Experiment20	0.105709108	1	[[27 0][0 28]]
Experiment21	0.133857485	1	[[10 0][0 8]]
Experiment22	0.11692708	1	[[62 0][0 64]]
Experiment23	0.125063348	1	[[21 0][0 34]]
Experiment24	0.142596304	1	[[10 0][0 8]]
Experiment25	0.001081975	1	[[60 0][0 66]]
Experiment26	0.002038799	1	[[25 0][0 30]]
Experiment27	0.002722447	1	[[9 0][0 9]]
Experiment28	0.004139937	0.944444444	[[6 0][1 11]]
Experiment29	0.003059537	0.981818182	[[27 1][0 27]]
Experiment30	0.004166858	1	[[11 0][0 7]]
Experiment31	0.000619753	1	[[63 0][0 63]]
Experiment32	0.00084045	1	[[23 0][0 32]]
Experiment33	0.000943406	1	[[9 0][0 9]]
Experiment34	0.000656387	1	[[58 0][0 68]]
Experiment35	0.000867542	1	[[33 0][0 22]]
Experiment36	0.000933891	1	[[7 0][0 11]]

Figure 4

features_selection	training_set/whole_dataset(%)	LR	NN	SVM
all wavelength	30	2	3	1
Wavelength: 420-470	30	1	3	2
Wavelength: 576-624	30	2	3	1
Wavelength: 705-751	30	2	3	1
all wavelength	70	2	3	1
Wavelength: 420-470	70	1	2	3
Wavelength: 576-624	70	2	3	1
Wavelength: 705-751	70	2	3	1
all wavelength	90	2	3	1
Wavelength: 420-470	90	1	3	2
Wavelength: 576-624	90	2	3	1
Wavelength: 705-751	90	2	3	1

Figure 5

Figure 5 shows the speed of training models. The number below each model demonstrates the ranking, and the smaller the value is, the faster the model being trained. According to figure 5, it is easy to be noticed that in most cases, SVM is most fast to train a model, and NN is lowest.

Using all features to train models is to act as control group. Whatever how big the training data set is, the accuracy of the models trained by LR and SVM is perfect (i.e. 1). However, When the training set is 90% of the whole set, the accuracy of the model trained by NN is very low (0.33). The interesting thing is that the true negative is 6 and the false positive is 12, which means all samples being predicted to green whatever it is. Comparing with the accuracy of the other two models (using same features but having different percentage of training set), the accuracy of this model decreases dramatically, therefore it is believed that the general reason is the model is overfitting. When the wavelengths between 420 to 470 as training set, training a satisfied model (namely the accuracy of the model being 1) is most difficult. According to the confusion matrix of experiment 4 and 5, more errors occur in wrongly justifying green as red. As the number of samples in the training set increases, the accuracy rate increases. However, this may not mean that the model must become better because the volume of validation set reduces, in other words, if there is more validation set, the accuracy rate may decrease. Here, if it is assumed that a higher accuracy score means a better model, then the satisfied model can only be built when the training set reaches to 90%, and the model is based on LR or SVM. Regardless of the volume of a training set, using NN cannot train a perfect model, and the result becomes worse when using 90% data set as training set. From confusion matrix, there are no clear indications that which part has a significant problem.

Other two groups of features, 576-624 and 705-751, got satisfied models whatever the type of algorithms or the number of samples is used. This because both groups of features can reflect the differences between two colours, namely, the colours have different trends and distribution in different wavelengths within 576-624 and 705-751. These phenomena are same as my prediction in the section of high dimensionality, and the analysis of reasons can also refer to there.

Overall, in three different machine learning algorithms, NN performs relatively worst, which consumes more time to train, has more number of lower-than-one accuracy score, and obtains the lowest accuracy score. However, this justification may be rash because the number of layers and the value of learning rate (alpha) are not changed that may affect the performance of the models trained by NN. But there is no doubt that the NN is more complicated, and it is no necessary to use in very simple classification problems.

(2) Multiclass classification

Same 3 types of regression models have been applied in this multi-colours classification problem, including logistic regression, neural network and support vector machine. They train models based on 3 different groups of features and 3 different volumes of training data set. All experiments are listed in the figure 6, and all prediction results are demonstrated in figure 7. Models are evaluated by comparing the speed of building a model, comparing the accuracy of predicting the result of validation set in different models and analysing the classification report in different models.

The first two criteria are same as the criteria in binary classification. Precision, recall, f1-score shown in Classification report are used to evaluate the prediction results. Precision is equal to the count of true positives divided by the sum of true positives and false positives: $\frac{TP}{TP+FP}$, recall is equal to the count of true positives divided by the sum of true positives and false negatives: $\frac{TP}{TP+FN}$, and f1-score is the harmonic mean of precision and recall: $\frac{2TP}{2TP+FP+FN}$.

Precision indicates the ratio between correct true predictions, and all true predictions, while recall means the rate between correct true predictions and the actual total number of samples of that given class. Normally, a bigger training set costs longer time to build a model. However, based on figure 7, when using same algorithm and same features, the training set occupying the whole set being 70% spend the longest time training a model. This is due to losing information when reducing the training size. For example, the distinction between colour patterns may become less clear when only using a few number of samples, resulting in longer time to converge (i.e. a greater number of iterations). Although the larger training size may seem like it should take longer, it appears that the addition of the data contains more information to allow a faster convergence. Average consuming time in different algorithms is also calculated, 0.26 for logistic regression, 0.46 for neural network and 0.617 for support vector machine.

Different with binary classification, using support vector machine to solving multi-class classification problems takes longest time. Unfortunately, the effect of using vector classifiers is also not satisfactory. The lower predication accuracy, 0.51, 0.6 also occurs in support vector classifiers. Similar qualification of classifiers are trained by logistic regression. However, when the features are all wavelength, the accuracy of models reaches to 1. Personally, the logistic regression is too simple to suit to train a multi-class classifier. To improve the accuracy, the methods can be increase the number of features, otherwise the model might be under fitting. From the results presented in figure 7, Using neural network to train data did not perform better than other two algorithms, but when the features are between 580 and 621, the prediction accuracy maintained in 1. Besides, the wavelength from 650 to 690 as features usually show worse accuracy in all algorithms, which is similar to my prediction. All in all, when there are more than two classes need to be detected, feature selection seems more important.

The classification reports for the models that have the accuracy lower than 1 are shown in the figure 8-19. These reports describe more details about the place of making errors. Take figure 8 for example, the accuracy of predicting blue is precise, namely, 100%, but the ability of predicting green is very week. The precision, 0.24, implies that many other colours are wrongly judged as green. Using similar methods, other figures can be understand easily.

Overall, comparing the accuracy score between binary classifiers and multi-class classifiers, it proves the relatively complexity of the latter. Facing the complicated problem, choosing proper features and the amount of data as training set are very important. Neural network are more suitable for complex problems because neural network are more easily to occur overfitting in simple classification problems.

	model	features_selection	training_set/whole_dataset(%)
Experiment1	logistic regression	all wavelength	30
Experiment2	logistic regression	all wavelength	70
Experiment3	logistic regression	all wavelength	90
Experiment4	logistic regression	Wavelength: 580-621	30
Experiment5	logistic regression	Wavelength: 580-621	70
Experiment6	logistic regression	Wavelength: 580-621	90
Experiment7	logistic regression	Wavelength: 650-690	30
Experiment8	logistic regression	Wavelength: 650-690	70
Experiment9	logistic regression	Wavelength: 650-690	90
Experiment10	neural network	all wavelength	30
Experiment11	neural network	all wavelength	70
Experiment12	neural network	all wavelength	90

Experiment13	neural network	Wavelength: 580-621	30
Experiment14	neural network	Wavelength: 580-621	70
Experiment15	neural network	Wavelength: 580-621	90
Experiment16	neural network	Wavelength: 650-690	30
Experiment17	neural network	Wavelength: 650-690	70
Experiment18	neural network	Wavelength: 650-690	90
Experiment19	support vector machine	all wavelength	30
Experiment20	support vector machine	all wavelength	70
Experiment21	support vector machine	all wavelength	90
Experiment22	support vector machine	Wavelength: 580-621	30
Experiment23	support vector machine	Wavelength: 580-621	70
Experiment24	support vector machine	Wavelength: 580-621	90
Experiment25	support vector machine	Wavelength: 650-690	30
Experiment26	support vector machine	Wavelength: 650-690	70
Experiment27	support vector machine	Wavelength: 650-690	90

Figure 6

	Time (seconds)	Accuracy
Experiment1	0.057821755	1
Experiment2	0.596354043	1
Experiment3	0.010282472	1
Experiment4	0.0077878	1
Experiment5	0.229337243	0.403174603
Experiment6	0.047030348	0.784126984
Experiment7	0.005749909	1
Experiment8	1.36099925	0.876190476
Experiment9	0.038982261	0.428571429

Experiment10	0.141836858	1
Experiment11	2.872304606	0.614814815
Experiment12	0.140961869	1
Experiment13	0.017104323	1
Experiment14	0.430721015	1
Experiment15	0.099155524	1
Experiment16	0.014427338	0.977777778
Experiment17	0.279205859	0.6
Experiment18	0.145264595	0.777777778
Experiment19	0.188029651	1
Experiment20	3.722560547	0.6
Experiment21	0.238190083	1
Experiment22	0.021246687	1
Experiment23	0.676098308	0.8
Experiment24	0.129840819	1
Experiment25	0.019527107	1
Experiment26	0.336603286	0.511111111
Experiment27	0.226567599	0.733333333

Figure 7

```
A B B
0.229337243 seconds
accuracy: 0.4031746031746032
classification report              precision    recall  f1-score   support

       blue       1.00      1.00      1.00        68
      green       0.24      1.00      0.39        59
       pink       0.00      0.00      0.00        63
        red       0.00      0.00      0.00        61
     yellow       0.00      0.00      0.00        64

avg / total       0.26      0.40      0.29       315
```

Figure 8 Experiment 5

```
A B C
0.047030348 seconds
accuracy: 0.7841269841269841
classification report             precision    recall  f1-score   support

       blue       1.00      1.00      1.00        59
      green       1.00      1.00      1.00        60
       pink       0.00      0.00      0.00        68
        red       0.48      1.00      0.65        64
     yellow       1.00      1.00      1.00        64

avg / total       0.68      0.78      0.71       315
```

Figure 9 Experiment 6

```
A C B
1.360999250 seconds
accuracy: 0.8761904761904762
classification report             precision    recall  f1-score   support

       blue       1.00      0.90      0.95        71
      green       0.89      1.00      0.94        57
       pink       1.00      0.98      0.99        64
        red       0.70      0.85      0.77        61
     yellow       0.80      0.65      0.71        62

avg / total       0.88      0.88      0.88       315
```

Figure 10 Experiment 8

```
A C C
0.038982261 seconds
accuracy: 0.42857142857142855
classification report             precision    recall  f1-score   support

       blue       0.50      1.00      0.66        63
      green       0.00      0.00      0.00        64
       pink       1.00      0.17      0.29        65
        red       0.34      1.00      0.51        61
     yellow       0.00      0.00      0.00        62

avg / total       0.37      0.43      0.29       315
```

Figure 11 Experiment 9

```
B A B
2.872304606 seconds
accuracy: 0.6148148148148148
classification report              precision    recall  f1-score   support

       blue       1.00      0.94      0.97        17
      green       0.00      0.00      0.00        32
       pink       1.00      0.41      0.58        32
        red       0.56      1.00      0.72        24
     yellow       0.48      1.00      0.65        30

avg / total       0.57      0.61      0.53       135
```

Figure 12 Experiment 11

```
B C A
0.014427338 seconds
accuracy: 0.9777777777777777
classification report              precision    recall  f1-score   support

       blue       1.00      1.00      1.00        32
      green       1.00      1.00      1.00        20
       pink       1.00      1.00      1.00        27
        red       1.00      0.89      0.94        28
     yellow       0.90      1.00      0.95        28

avg / total       0.98      0.98      0.98       135
```

Figure 13 Experiment 16

```
B C B
0.279205859 seconds
accuracy: 0.6
classification report              precision    recall  f1-score   support

       blue       1.00      1.00      1.00        37
      green       1.00      1.00      1.00        23
       pink       0.00      0.00      0.00        28
        red       0.00      0.00      0.00        26
     yellow       0.28      1.00      0.44        21

avg / total       0.49      0.60      0.51       135
```

Figure 14 Experiment 17

```
B C C
0.145264595 seconds
accuracy: 0.7777777777777778
classification report             precision    recall  f1-score   support

       blue       1.00      1.00      1.00        27
      green       1.00      1.00      1.00        21
       pink       0.55      1.00      0.71        27
        red       1.00      0.53      0.70        30
     yellow       0.64      0.47      0.54        30

avg / total       0.83      0.78      0.77       135
```

Figure 15 Experiment 18

```
C A B
3.722560547 seconds
accuracy: 0.6
classification report             precision    recall  f1-score   support

       blue       0.83      1.00      0.91        10
      green       0.00      0.00      0.00        12
       pink       0.64      1.00      0.78         7
        red       1.00      0.20      0.33         5
     yellow       0.43      0.82      0.56        11

avg / total       0.50      0.60      0.50        45
```

Figure 16 Experiment 20

```
C B B
0.676098308 seconds
accuracy: 0.8
classification report             precision    recall  f1-score   support

       blue       1.00      1.00      1.00        10
      green       1.00      1.00      1.00        11
       pink       0.47      1.00      0.64         8
        red       0.00      0.00      0.00         9
     yellow       1.00      1.00      1.00         7

avg / total       0.71      0.80      0.74        45
```

Figure 17 Experiment 23

```
C C B
0.336603286 seconds
accuracy: 0.5111111111111111
classification report              precision    recall  f1-score   support

       blue       1.00      1.00      1.00         9
      green       1.00      1.00      1.00        10
       pink       0.00      0.00      0.00         9
        red       0.00      0.00      0.00        13
     yellow       0.15      1.00      0.27         4

avg / total       0.44      0.51      0.45        45
```

Figure 18 Experiment 26

```
C C C
0.226567599 seconds
accuracy: 0.7333333333333333
classification report              precision    recall  f1-score   support

       blue       1.00      1.00      1.00        13
      green       1.00      1.00      1.00         7
       pink       0.70      1.00      0.82         7
        red       1.00      0.10      0.18        10
     yellow       0.36      0.62      0.45         8

avg / total       0.84      0.73      0.69        45
```

Figure 19 Experiment 27

Appendix 1

Binary Task:

- What percentage dataset you prefer to use as training set? **30%**
- How many features would you like to select? **Wavelength: 500-690**
- What kind of regression do you prefer? **Logistic Regression**

Logistic Regression
1
1
0
0
1
0
0
0
1
0
0
1
1
1
0
1
1
0
1
0

Appendix 2

Binary Task:

- What percentage dataset you prefer to use as training set? **70%**
- How many features would you like to select? **Wavelength: 580-621**
- What kind of regression do you prefer? **Support Vector Machine**

Support Vector Machine
2
0
2
0
0
0
2
0
4
1
4
3
3
2
0
4
2
4
3
3
4
1
2
1
4
2
3
2
0
1
1
1
3
1
2
4
2
3
1
4
0
0
1
1
3
0
3
4
3
4