MSc Final Project Final Report:
Machine Learning Aided Cancer Diagnosis Based on Fourier Transform Infrared Spectral Data

Student name: Wei, Chi

Student ID: 20195059

Supervisor: Dr Sendy Phang

Abstract

Fourier Transform Infrared (FTIR) has been successfully used for analysing cell biology and offering label-free extraction of biochemical information. Cancer diagnosis based on FTIR spectral data has the potential to revolutionise current cancer diagnosis methods because of its time-saving, low cost and non-destructive results. However, despite the advantages FTIR offers, there are many challenges that stopped the implementation of this technology in clinical application. One major difficulty has been reaching a consensus about the choice of parameters of pre-processing and classification models. In this paper a series of experiments is described, which aim was to find the most suitable pre-processing methods and classifiers that avoid underfitting and overfitting. This paper obtained high sensitivity and specificity and low misclassification rates.

Contents

1 Introduction **5**

2 Literature Review **6**

2.1 Pre-processing 7

2.1.1 Cutting Range 7

2.1.2 Spectra Derivatives 8

2.1.3 Baseline Correction 9

2.1.4 Normalisation 9

2.1.5 Principal Component Analysis 10

2.2 Validation Method 12

2.2.1 K-fold Cross Validation 13

2.2.2 Hold Out Validation 14

2.3 Classifier 15

2.3.1 LDA 15

2.3.2 KNN 16

2.3.3 SVM 17

2.3.4 Neural Network 17

2.3.5 Logistic Regression 17

2.4 Model evaluation 18

2.4.1 Evaluation Index 19

2.4.2 Overfitting and Underfitting 20

2.5 Conclusion for Literature Review 21

3 Methodology **22**

3.1 Dataset and Software 22

3.2 Pre-processing 23

3.3 Validation Method 24

3.4 Classifier 24

3.5 Model Evaluation 24

3.6 Conclusion 24

4 Experiments and Results **25**

4.1 All Class Validation 26

4.2 Find the Best Pre-processing Combination 29

4.3 Find the Good Fitting Range 39

5 Discussion **44**

6 Conclusion **46**

7 Acknowledgement **47**

Nomenclature

Amide I is the most intense absorption band in proteins. Amide I at around 1650 cm^{-1} and it is the most intense absorption band.

Amide II is one of the major bands of the protein infrared spectrum. Amide II at around 1550 cm^{-1} and it is the second most intense absorption band.

Fourier Transform Infrared Fourier-transform infrared spectroscopy (FTIR) is a technique used to obtain an infrared spectrum of absorption or emission of a solid, liquid or gas.

k-nearest neighbours (kNN) is a non-parametric method proposed by Thomas Cover used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space.

Linear discriminant analysis (LDA) is a linear classifier used in machine learning, pattern recognition and statistics.

Sensitivity measures the proportion of positives that are correctly identified (e.g., the percentage of sick people who are correctly identified as having some illness).

Specificity measures the proportion of negatives that are correctly identified (e.g., the %age of healthy people who are correctly identified as not having some illness).

Support vector machine (SVM) are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis.

1 Introduction

Cancer is the major public health problem worldwide [1]. In the United States, 2020, [1] predicted 1,806,590 new cancer cases and 606,520 of cancer deaths. Detection of cancer punctually can help cancer death rate decline [1]. However, detection of some cancer can be difficult because some tissue is located deep within the body like pancreas or dangerous and sensitive like brain [2]. Moreover, current detection methods are not good enough for cancer diagnosis. For example, Magnetic resonance imaging (MRI) is time-consuming and expensive for patients [3]. It is necessary to develop a new way which can offer the ability to diagnose cancer better than current methods.

The molecular composition of living organisms can show their physiological states well [4]. Fourier Transform Infrared (FTIR) spectroscopy can extract biochemical information and images non-perturbatively to diagnose and assessment of cell functionality [5]. Figure 1 is a typical biological spectrum showing biomolecular peak assignments from FTIR spectral data [5]. Many studies have shown that FTIR can be used to diagnose cancer [6, 7, 8, 9]. Compare to other methods of cancer diagnose, FTIR has many advantages such as time-saving, easy to use and [5]. It can show much more information than other methods do because it relies on the detection of unique sample absorbance due to different amount of chemical bond and chemical composition in the cells and tissues. Figure 1 shows the highest peak is Amide I at around 1650 cm^{-1} and second highest peak Amide II at around 1550 cm^{-1}.

However, FTIR is not enough for doctors to diagnose cancer. One major difficulty in this field has been finding a proper way to analyse and interpret spectral data[5]. FTIR bring high demands of data analytic techniques because it contains too much information. Even water in tissues can influence the result [4]. Only with appropriate pre-processing and data analytic method, we can unravel the presence of malignant cell or tissue within the FTIR data [10, 5].

Meanwhile, machine learning has been applied to many places and get many impressive achievements. To create a new and efficient way to diagnose cancer, this project will combine FTIR and machine learning methods. This project uses FTIR to get full information from normal and cancer cells and tissues, then use machine learning methods to analyse FTIR spectral data.

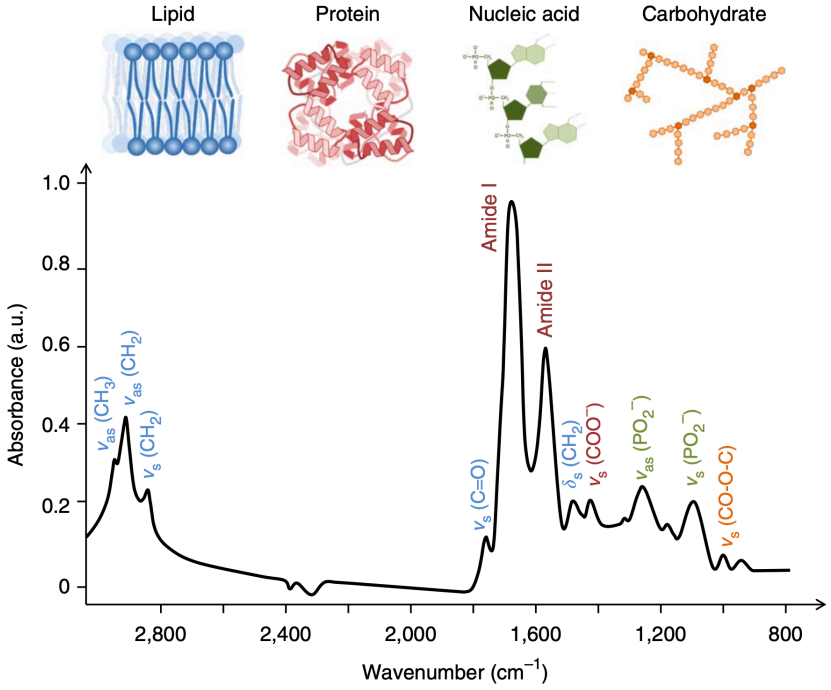

Figure 1: Typical biological spectrum showing biomolecular peak assignments[4]

There are four main stages in this project which include pre-processing, validation, classifier and model evaluation. Figure 2 schematically shows the workflow of FTIR spectral data analysis. Detail of each stage will be described in Chapter 2. After getting results from classifiers, then evaluate the model to see is the results have good generalization so it is reliable for future use.

This dissertation is structured in the following fashion. Chapter 2 will describe the literature review of this project. Chapter 3 will describe in detail the methodology of this project use. In Chapter 4, the details of the experiments will be described and their results will be listed. Chapter 5 will explain outcomes in the context of work and engineering application. Chapter 6 will include a summary of key results and the recommendation for further work. The last chapter will be the acknowledgement of this project.

2 Literature Review

It is necessary to read and summarize relevant literature in this field because of the advance of this project. Since there are 4 main stages in this project, this Chapter will describe these four

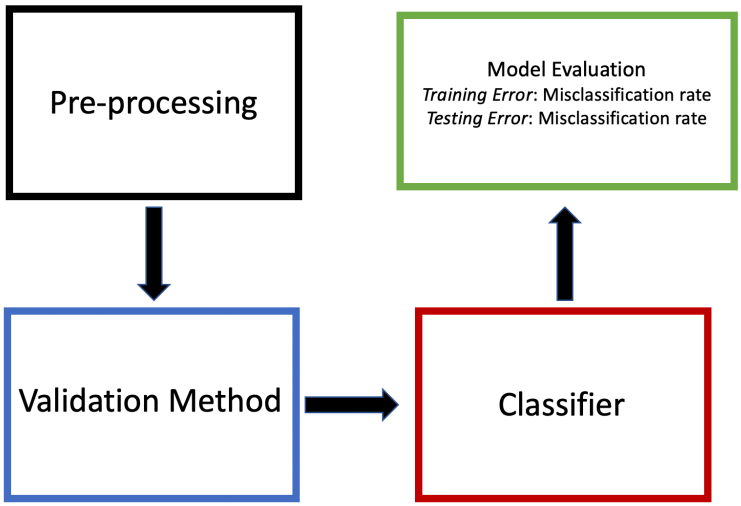

Figure 2: Classification Flow Diagram

stages in order. At last, there will have a conclusion of this Chapter.

2.1 Pre-processing

The purpose of applying pre-processing is to improve the robustness and accuracy of later analyses and predictions and to help human being have a deeper understanding of cancer [5]. Pre-processing can be divided into several parts, cutting range, differentiation baseline correction and normalisation [5]. Figure 3 is a typical mathematical operation used in the pre-processing of FTIR spectra [5]. Figure 3 shows four different path of pre-processing commonly used in the biomedical community [5]. The first paths and the second path both include Rubber band baseline correction first. Then the first path includes normalisation to Amide I peak while the second path includes normalisation to Amide II peak. The third path includes first differentiation first while the forth path includes second differentiation. After that, both the third and the forth includes vector normalisation. In the following subsections, each of those pre-processing will be described in detail.

2.1.1 Cutting Range

In biological applications, the first step is usually cut spectral data to keep certain region [11]. In some studies, they cutting spectral data from 1800 to 900 cm^{-1} because this range is called the fingerprint region [5, 12, 13]. Fingerprint region includes important absorptions of lipids

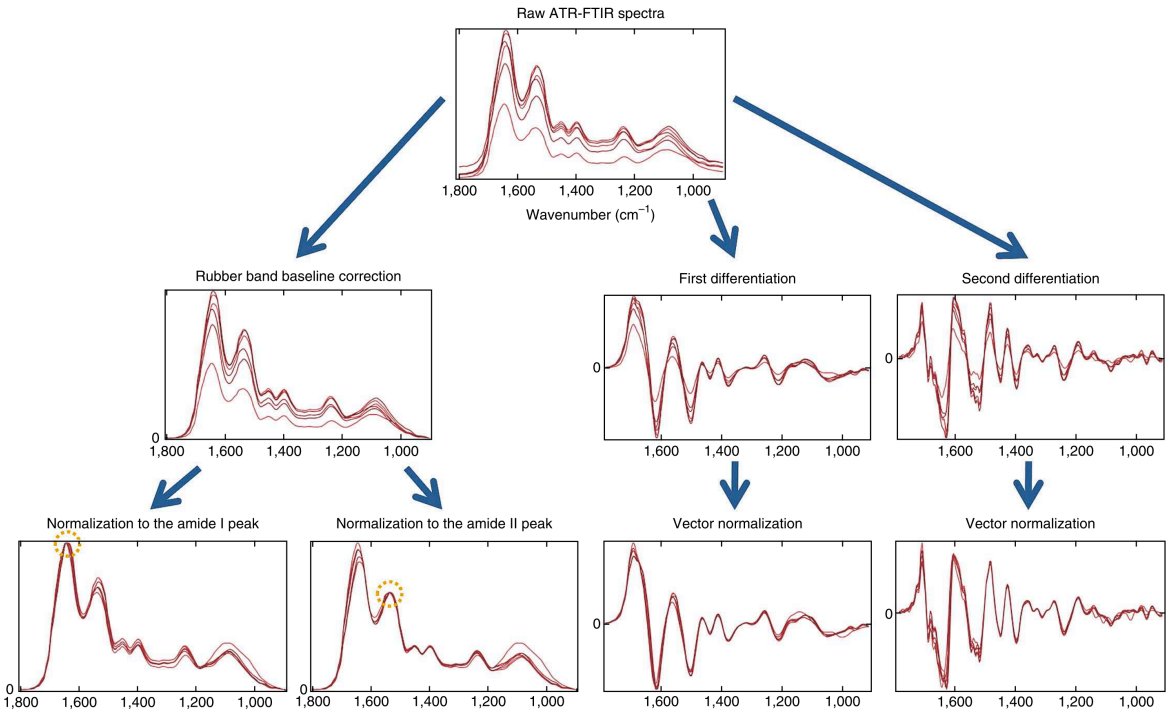

Figure 3: Typical mathematical operation used in the Pre-processing of FTIR spectra [5]

[11]. In this region, there are C=O symmetric stretching at 1750 cm^{-1}, CH_2 bending at 1470 cm^{-1}, the highest peak Amide I at around 1650 cm^{-1}, Amide II at around 1550 cm^{-1}, Amide III at around 1260 cm^{-1}, etc [11].Cutting from 1800 to 900 cm^{-1} also removes spectral artefacts like water and other unrelated absorptions [11].

2.1.2 Spectra Derivatives

First or second differentiation is usually used to spectral data to correct light scattering and eliminate noise[5, 11]. These can also help finding impotent features complex samples[11]. First and second differentiation can be applied by Savitzky Golay (SG) filter.

SG filter was proposed by Savitzky and Golay in 1964 and it is widely used as a denoising method [14, 15].It can be considered as a weighted moving average filter with the weight defined by polynomial order [16]. There are three important parameters in SG filter, polynomial order, window length and derivative order [16]. If the polynomial order is too high, it may generate new noise by yield redundant data [16]. If the polynomial order is too low, it may yield over smoothing and signal distortion [16]. If the window length is too long, it may cause some loss of valid signals [16]. If the window length is too short, it can not denoise well [16]. Derivative

order contains order 1 and order 2 [17].

2.1.3 Baseline Correction

Spectra often need Baseline correction before data analysis. Baseline correction can remove background absorption interferences [11]. Common baseline correction including Rubber band baseline correction, polynomial correction,etc [11].

Figure 4 compares the original data with data after Rubber band baseline correction. Spectra on the top are original data and spectra at the bottom is the original data after Rubber band baseline correction.

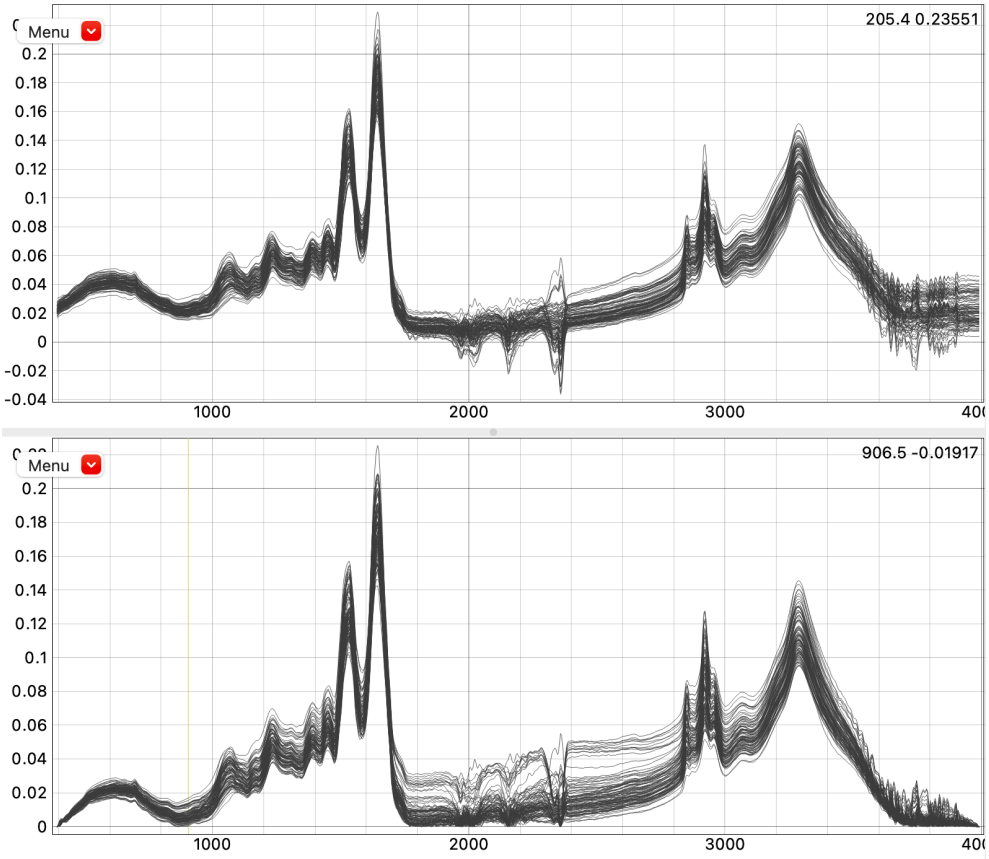

Figure 4: Effect of Rubber band baseline correction

2.1.4 Normalisation

Common normalisation methods include Amide I normalisation, Amide II normalisation and Vector normalisation [5, 11]. Amide I/II normalisation forces all spectra to have the same

absorbance intensity at the Amide I/II peak [5]. Amide I is the most intense absorption band at around 1650 cm^{-1}. Amide II is the second most intense absorption band at around 1550 cm^{-1}. Vector normalisation is used when a different sample thickness is needed while information is unknown [11].

Figure 5 compares the original data with data after Vector normalisation. Spectra on the top are original data and spectra at the bottom is the original data after Vector normalisation.

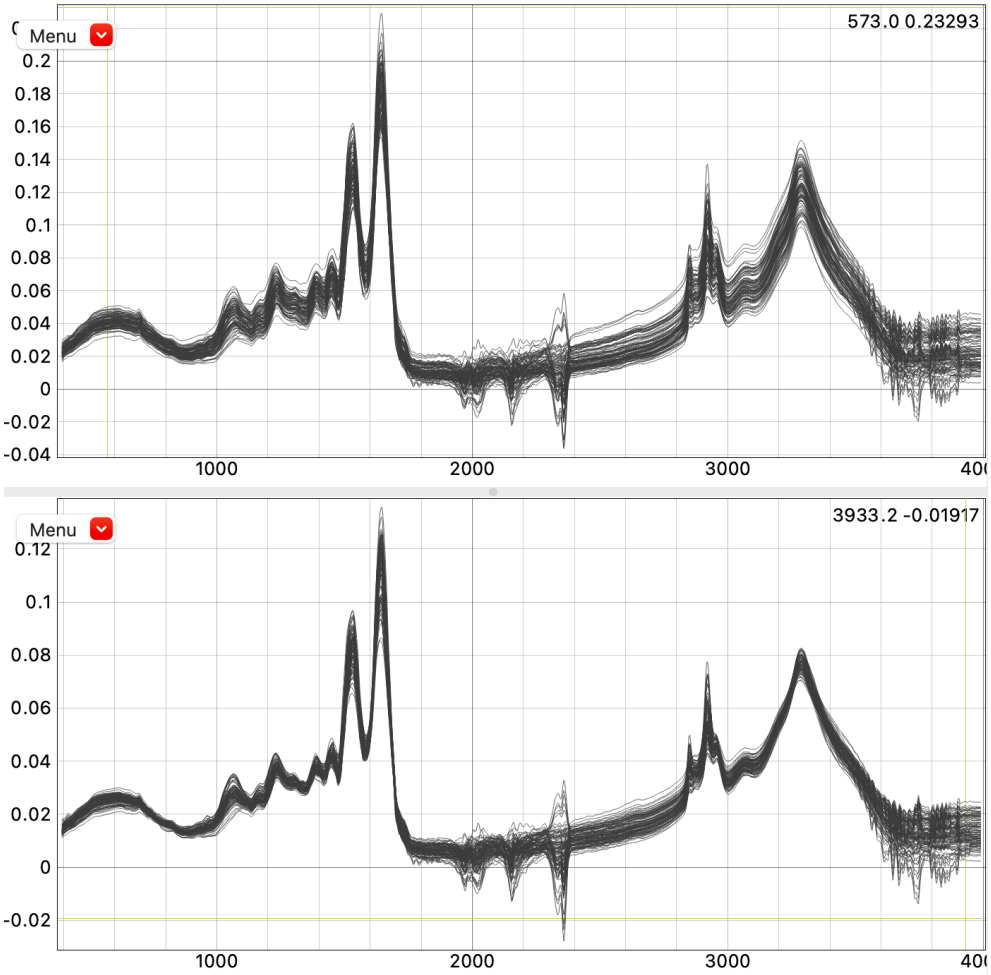

Figure 5: Effect of Vector Normalisation

2.1.5 Principal Component Analysis

To eliminate high-frequency noise, apart from using SG filter, another good method is Principal Component Analysis (PCA). PCA can extract important information from original data and represent it as a set of new orthogonal variables called principal components [18].

PCA is a popular unsupervised learning method to reduce the dimensionality of data [5]. It can also help prevent overfitting [19]. PCA is identified by Pearson in 1901 [20]. PCA broke the spectral data into some Principal Components (PCs) which explains most of the variance within the original dataset [11]. The PCs are orthogonal to each other and are generated in the order of variance decrease[11]. This means, the first PC explains the most of variance, the second explains the second most of variance and so on. The decomposition of PCA is in the form shows in the following equation [11].

$$\mathbf{X} = \mathbf{TP}^{\mathrm{T}} + \mathbf{E} \tag{1}$$

where X represents the preprocessed spectral data, T represents the scores of PCA and E represents the residuals.

For instance, in a dataset shows in the left panel of Figure 6. The process of PCA is to find the first PCs which can explain the most variance. Then find the PC2. The result of PCA for this dataset is shown in the right panel of Figure 6.

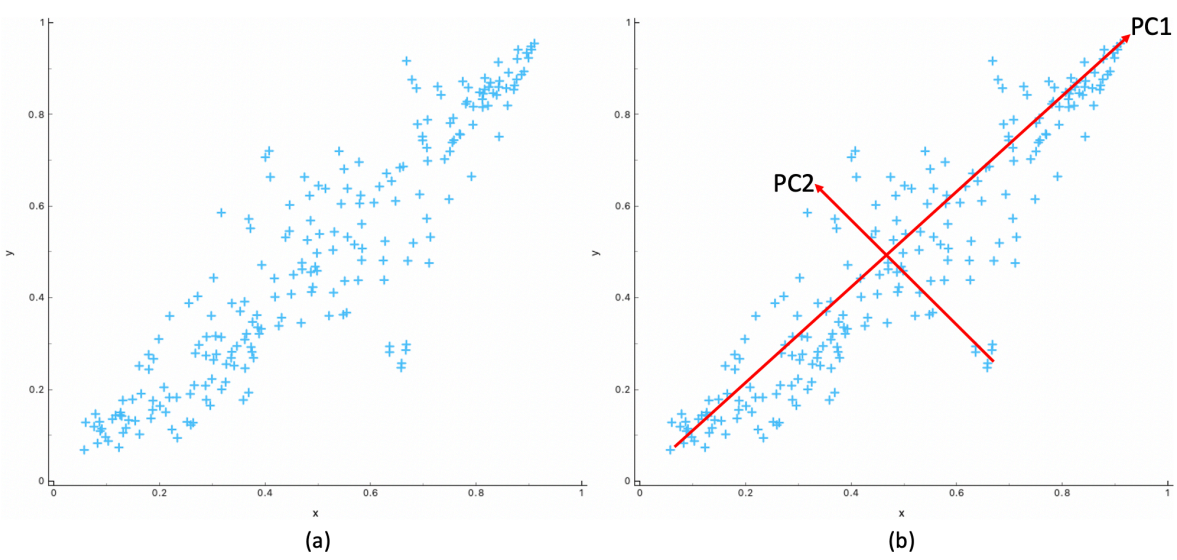

Figure 6: Example of PCA

With the more number of PCs, it can explain more variance in the original dataset. However, it is not necessary to over explain variance. Usually, reachers use the number of PCs can explain 95% of variance [21]. Figure 7 shows the relationship between the number of Principal Components and percentage explained variance. For example, two Principal Components

can explain 54.75% of the variance. To explain 95% of the variance, it needs 10 Principal Components. Figure 7 is the scatter plot matrix of PCA. The y-axis from top to bottom is PC1 to PC10. The x-axis from left to right is PC1 to PC10. In this PCA operation is based on brain dataset which includes two classes, normal brain and brain cancer. The blue represents the normal brain and orange represents brain cancer.

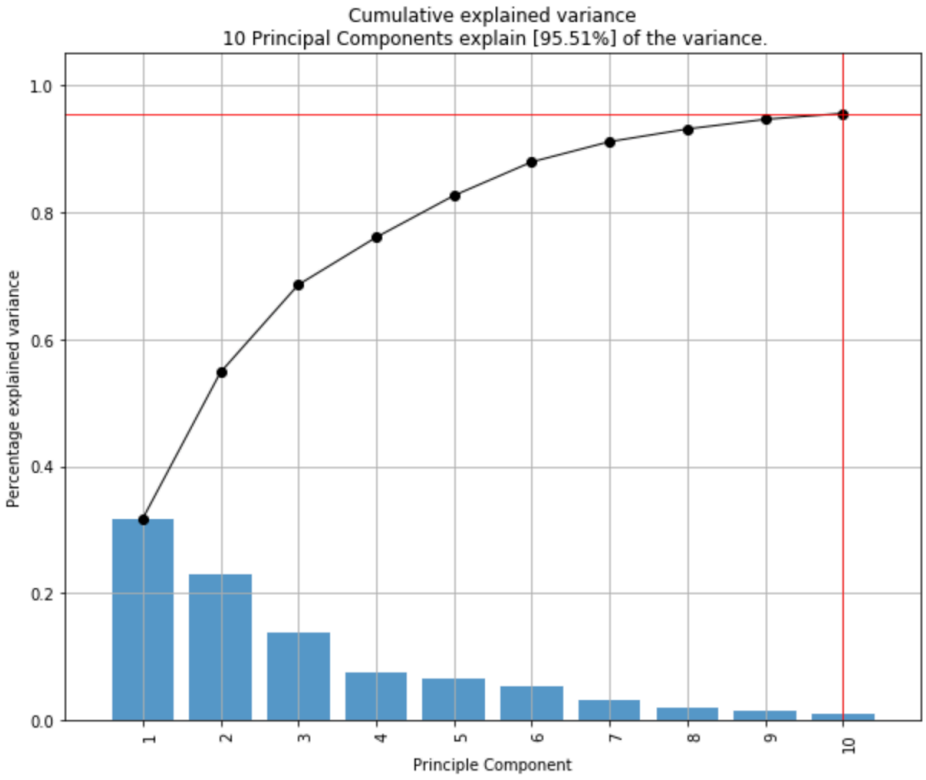

Figure 7: The relationship between number of Principal Components and %age explained variance

2.2 Validation Method

Most researchers in the field of FTIR only use cross validation which including two examples, K-fold and leave-one-out validation [11, 5, 13, 12].

Meanwhile, researchers in the field of data analyse use hold-out validation because cross validation performance does not reflect the model predictive performance toward unknown samples [19, 22, 11]. In hold-out validation, the dataset is first divided into training data and testing data [19, 22]. 70% training and 30% testing or 80% training and 20% testing are the

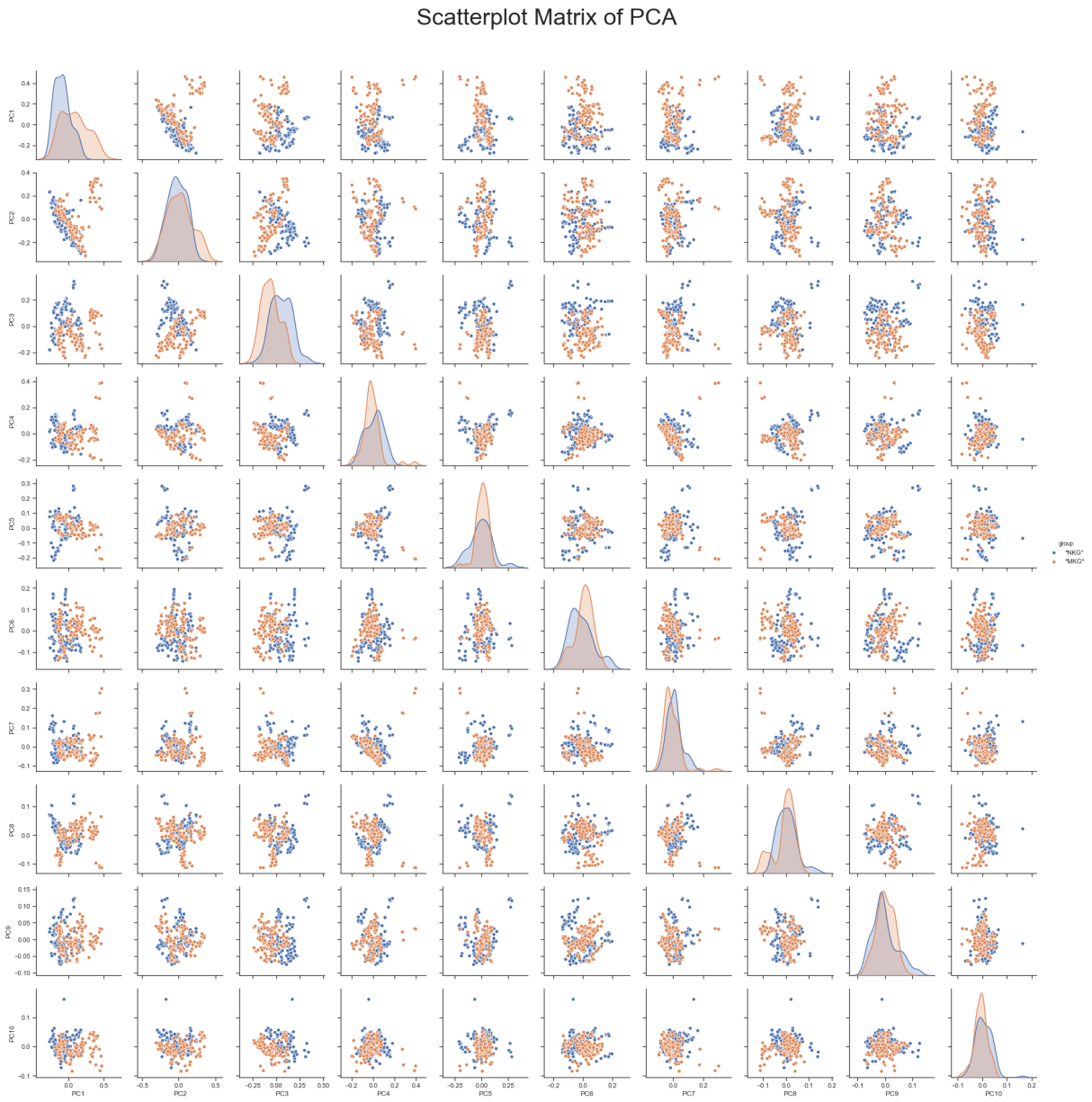

Figure 8: Scatter Plot Matrix of PCA

most common hold-out validation methods. Then on the training data, they use k-fold validation [19]. The testing data is never seen by the model so it can evaluate the quality of the mode.

2.2.1 K-fold Cross Validation

The process of K-fold cross validation is:

1. Divide the dataset into K parts randomly

2. Use (K-1) parts to train the model then use the last part to test the model

3. Repeat the second step but pick another part to test the model till all parts of data was been used to test. This step should repeat (K-1) times.

4. The final result is the mean of every result each time.

2.2.2 Hold Out Validation

The process of hold out validation is:

1. Divide the dataset into two parts, training data and testing data. The key point at this step is to make sure the testing data is new compare to training data. The proportion of training data is normally 70% or 80 %.

2. Train the model on training data.

3. Test the model on testing data.

4. Get the two results from training data and testing data.

Figure 12 schematically shows the 10-fold cross validation and hold out validation. The validation method on the top was 10-fold cross validation. The second and the third was hold out validation. In hold out validation, use 80% or 70% of data to train the model and use the rest of data to test the model.

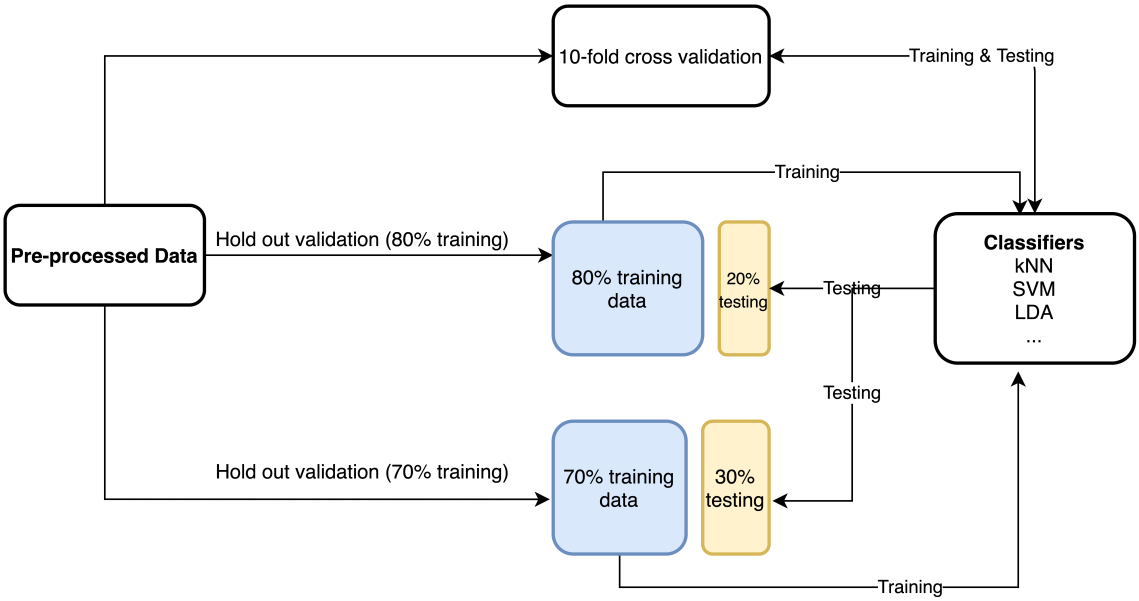

Figure 9: Validation methods used in this project

2.3 Classifier

In other studies, they used quadratic Support Vector Machine (SVM), K-nearest neighbours (kNN), Logistic Regression and Linear Discriminant Analysis (LDA) [19, 11, 5]. Some literature says neural network and deep learning methods are not suitable for this project because these models are too complicated so overfitting can easily occur [11].

2.3.1 LDA

Linear Discriminant Analysis (LDA) is a very popular data dimension technique used in many fields [23].The goal of LDA is to transfer the original data matrix into a lower dimensional space [23]. There are three steps in the process of LDA [23]:

1. Calculate between-class variance (S_B) which is the separability between different classes. Assume the original data matrix is $X = \{x_1, x_2, \ldots, x_N\}$ and N is the total number of samples. Each sample contains M features. Assume this dataset has $c = 3$ classes. x_i represents the i^{th} sample. The total mean is μ and the mean of i^{th} class is μ_i. m_i represents the projection of the mean of the i^{th} class and m represents the projection of the total mean of all classes. W represents the transformation matrix of LDA. Thus, the between-class variance which is $(m_i - m)$ can be calculated as follow:

$$(m_i - m)^2 = \left(W^T\mu_i - W^T\mu\right)^2 = W^T(\mu_i - \mu)(\mu_i - \mu)^T W \tag{2}$$

   Since m_i can be calculated by $m_i = W^T\mu_i$ and m can be calculated by $m = W^T\mu$,μ_i can be calculated by follow:

$$\mu_j = \frac{1}{n_j}\sum_{x_i \in \omega_j} x_i \tag{3}$$

   and μ can be calculated by:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i = \sum_{i=1}^{c}\frac{n_i}{N}\mu_i \tag{4}$$

   In this case, equation (2) can be simplified as:

$$(m_i - m)^2 = W^T S_{Bi} W \tag{5}$$

2. Calculate within-class variance (S_W) which is the distance between the mean and each samples. The within-class variance is defined by the difference between projected

samples of each class $\left(W^T x_i\right)$ and the projected mean (m_i) and can be calculated by follow:

$$\begin{aligned} &\sum_{x_i \in \omega_j, j=1,\ldots,c} \left(W^T x_i - m_j\right)^2 \\ &= \sum_{x_i \in \omega_j, j=1,\ldots,c} \left(W^T x_{ij} - W^T \mu_j\right)^2 \\ &= \sum_{x_i \in \omega_j, j=1,\ldots,c} W^T \left(x_{ij} - \mu_j\right)^2 W \\ &= \sum_{x_i \in \omega_j, j=1,\ldots,c} W^T \left(x_{ij} - \mu_j\right)\left(x_{ij} - \mu_j\right)^T W \\ &= \sum_{x_i \in \omega_j, j=1,\ldots,c} W^T S_{W_j} W \end{aligned} \tag{6}$$

3. Construct the lower dimensional space which maximize the between-class variance and minimize the within-class variance. With the results of between-class variance (S_B) and within-class variance (S_W), transformation matrix (W) of LDA can be calculated by follow:

$$\arg\max_W \frac{W^T S_B W}{W^T S_W W} \tag{7}$$

2.3.2 KNN

KNN [24] is a classification method based on the distance between samples and the number of nearest surrounding neighbour [11]. There are different ways to calculate the distance between samples, Euclidean, Manhattan, Minkowski and Mahalanobis. The most common one is Euclidean distance expressed in Equation 8 [25]:

$$d = \sqrt{\sum_{k=1}^{n} (X_{1k} - X_{2k})^2} \tag{8}$$

where k is the number of values in the sample vector X_1, X_2 are input samples. The parameter in KNN that can change the complexity of the model is the number of neighbours [26].

Fine KNN takes one neighbour to distinguish sample data which means all points in each neighbour are weighted equally while weighted KNN uses a distance weight shown in Equation 9 [27]:

$$d = \sqrt{\sum_{k=1}^{n} w_i (x_{1k} - x_{2k})^2} \tag{9}$$

This makes closer neighbours of a query point have a greater influence than the neighbour further away.

2.3.3 SVM

Support Vector Machine (SVM) is a modern computational learning method [28, 29]. When LDA maximises distance between means of two classes divided by variance, SVMs maximise distance between nearest examples of each class through margin. Assume W is normal to the dividing plane. Assume x_- is a point on one side of and x^+ is the nearest point on the other side of decision boundary, so the Vector between two points is $(\mathbf{x}_+ - \mathbf{x}_-) = \lambda\mathbf{w}$. Thus, margin M is defined by Equation 10:

$$M = \|\mathbf{x}_+ - \mathbf{x}_-\| = \lambda\|\mathbf{w}\| \tag{10}$$

Assume the decision boundary is $\mathbf{w}^\top\mathbf{x} = 0$, plus margin is $\mathbf{w}^\top\mathbf{x} = +\gamma$ and minus margin is $\mathbf{w}^\top\mathbf{x} = -\gamma$. For any W, γ is fixed by data points. Consider scaling $\mathbf{w}^* = \frac{\mathbf{w}}{\gamma}$. Therefore for each possible W, M can be calculated as follow:

$$M = \frac{2\gamma}{\|\mathbf{w}\|} = \frac{2\gamma}{\|\gamma\mathbf{w}^*\|} = \frac{2}{\|\mathbf{w}^*\|} \tag{11}$$

To find the maximum margin, it needs to across all W.

The polynomial kernel of SVM is defined by:

$$(gx \cdot y + c)^d \tag{12}$$

where d decides the complexity of the model. When d is 1, then it is Linear SVM. When d is 2, it is Quadratic SVM. When d is 3, it is Cubic SVM. The bigger d is, the model is more complex.

2.3.4 Neural Network

Neural Networks are flexible computing frameworks and widely used for many machine learning applications including computer vision, Natural Language Processing (NLP) and speech recognition [30]. Neural Networks offers the highest accuracy at the cost of high model complexity [30].

Connection to neuron was shown in Figure 10 where $x_i, w_i, f_()$ and b are activations, weights, nonlinear function, and bias [31, 30].

2.3.5 Logistic Regression

Logistic regression proved its value in the field of statistics and machine learning [32]. On the contrary to what SVMs' approach, risk minimise, logistic regression uses a maximum

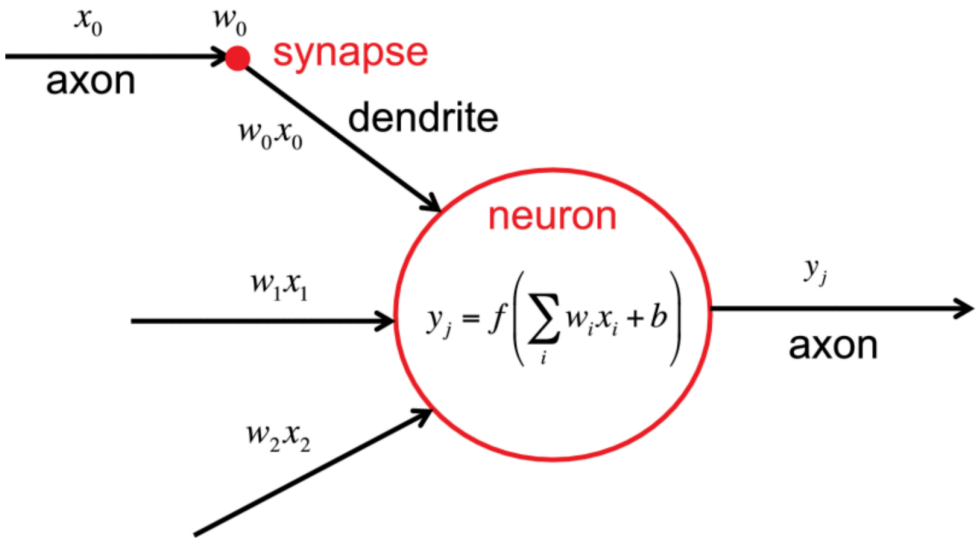

Figure 10: Connections to a Neuron [31]

likelihood argument to get probabilistic results [32]. The output of logistic regression can be transferred to probability based on the Equation 13:

$$\log\left(\frac{P(y=1 \mid x)}{1-P(y=1 \mid x)}\right) = \beta_0 + \beta^T x \tag{13}$$

where β_0 represents the intercept and β denotes a vector of weights. At last, the predicted probability for x can be calculated by Equation 15:

$$P(y=1 \mid x) = \frac{\exp\left(\beta_0 + \beta^T x\right)}{1+\exp\left(\beta_0 + \beta^T x\right)} \tag{14}$$

2.4 Model evaluation

After the results coming out, it is necessary to evaluate the results. Sometimes accuracy is not enough to evaluate a classifier [19]. Overfitting and underfitting are what we should we avoid in our model because those will lower models' performance. In underfitting, important potential interrelationships between the data features may be ignored [19]. In overfitting, although models can have a very good performance on training data, it performs badly on future new and previously unseen data [19]. What we want is the model have high accuracy and good generalization ability at the same time. Moreover, sensitivity and specificity is another thing we need to take into consideration.

2.4.1 Evaluation Index

The performance of the classification model needs to validated by calculating some parameters[11]. Table 1 shows the equations and meanings of these parameters. For models that have more than two classes, these parameters need to be calculated individually per class. For models that have only two classes, these parameters are only calculated once[11].

Table 1: Quality parameters to evaluate the classification model performance[11]

Parameters	**Equation**	**Meaning**
Accuracy	TP+TN/(Total)	The %age of correct classified samples
Sensitivity	TP/(TP+FN)	Proportion of positive samples that are correctly classified (True Positive Rate)
Specificity	TN/(TN+FP)	Proportion of negative samples that are correctly classified
Negative likelihood ratio	SPEC/(1-SENC)	Ratio between the probability of predicting a sample as negative when it is actually positive and the probability of predicting a sample as negative when it is truly negative.

FN = false negatives; FP = false positives; TN = true negatives; TP = true positives; Total = (TP+FP+TN+FN); SENC = sensitivity; SPEC = specificity.

To know more details about the results, the confusion matrix and ROC curve are needed. Confusion matrix visualises the performance of the classification model. Table 2 shows the structure of the confusion matrix. Confusion matrix shows the number of true positives, true negatives, false positives and false negatives. This can allow us to do more analysis in detail than accuracy.

Table 2: Confusion Matrix

		Actual Class	
		P	**N**
Predicted Class	**P**	TP	FP
	N	FN	TN

P = Positive; N = Negative; TP = True Positive; FP = False Positive; TN = True Negative; FN = False Negative.

ROC curve, or Receiver Operating Characteristic Curve, plots true positive rate, or Sensitivity, against the false positive rate, or (1-Specificity) [33].

2.4.2 Overfitting and Underfitting

Another potential problem of classification models is overfitting and underfitting. Overfitting and underfitting are caused by inappropriate model complexity [19]. If the model is too simple so that its estimation has low variance but high bias, it is underfitting [19]. If the model is too complex so that its estimation has high variance but low bias, it is overfitting [19].

First, divide the whole dataset to training dataset and testing dataset. For the same model, when model complexity increases, both error on testing data and error on training data decrease [19]. When model complexity increase continuously, error on test starts to increase while error on training data still decreases [19]. At the range before error on testing data increase, is the ideal range where the model is good fit [19]. Figure 11 shows how to find the good fit range based on different behave of errors on testing data and error on training data [19, 22]. When model complexity increase, error on training data decrease continuously. However, error on testing data decreases at first then increase again. Before the error on testing data increase is the range where the model is not overfitting nor underfitting.

There are many ways apart from accuracy to the estimated performance of a model. Logloss takes into account the uncertainty of predictions based on how much it varies from actual labels [34]. Logloss can give us a deeper view into the performance of the model[34, 35]. Sensitivity shows us how many false-negative predictions in the model. Sensitivity is important because it will be much more serious that cancer tissues been diagnosed to normal than normal tissues

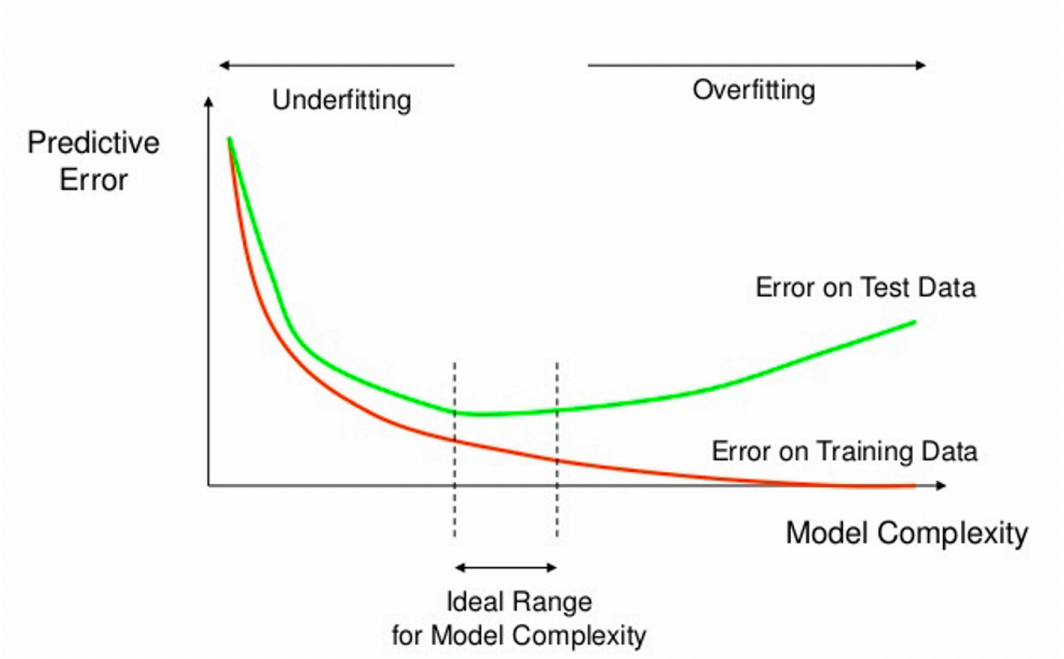

Figure 11: Relationship between underfitting, overfitting and model complexity[19]

been diagnosed to cancer. When the estimated performance of models, it is not enough to only use one metrics.

The most direct and useful to prevent overfitting is to use more data to train the model[22]. However, there are also other ways to prevent overfitting and underfitting as well[22, 19]. To prevent overfitting and underfitting, optimization is needed in the stage of pre-processing, validation and model evaluation. In the pre-processing stage, PCA can prevent overfitting [19]. In the stage of validation, divide the dataset into training data and testing data can help to prevent overfitting and underfitting [19]. Train the model only on training data and keep the testing data unseen for models. On training data, use k-fold cross validation [19]. By doing this, in the stage of model evaluation, it is possible that create a diagram similar to Figure 4 so the good fitting range can be found.

2.5 Conclusion for Literature Review

In pre-processing stage, most literature use cutting range from 1800 to 900 cm^{-1} first. Then use S-G filter to apply first and second differentiation and polynomial order 3, 5 and 7. After that, some researchers use baseline correction including amid I normalisation and Amid II normalisation while others use vector normalisation. At last, some literature shows the use

of PCA can improve the quality of prediction.

In the validation stage, most researchers in the field of FTIR only use k-fold validation. However, in the field of data analyse, many pieces of literatures show that divide the dataset into training data and testing data is necessary to avoid overfitting and underfitting.

As for classifiers, the most common classifiers are LDA, KNN, and SVM.

3 Methodology

After discussing all methods used by other researchers in each stage of spectral data analysis, this section will list all methods used in each stage of the project.

3.1 Dataset and Software

The dataset this project used is brain spectroscopy dataset. This dataset contains six different classes, Glioblastoma Multiforme, Glioma grade 3, Low grade glioma, Metastasis, Meningioma and Normal Brain. There are 1040 spectral data in total includes 140 normal brain spectral data, 100 Glioma grade 3 spectral data, 100 Glioblastoma Multiforme spectral data, 100 Low grade glioma spectral data, 300 Meningioma spectral data and 300 Metastasis spectra data. In most cases, this project only used two classes, Meningioma and normal brain. The 300 Meningioma spectral data was from 15 different patients and 140 normal brain spectral data was form 7 different patients.

The software this project used includes MATLAB (2019b), Orange data mining, and Python. This project used Matlab to transfer the raw spectra dataset format into CSV format. This process made processing the dataset easier because CSV format is commonly used in the data analysis field. The pre-processing, classification and validation process is done in Orange data mining. Some figures were generated by Python. Figure 12 shows a typical workflow in Orange Data Mining. Firstly, import CSV files and select columns. Then, operate pre-processing methods and PCA. Then send all data into a widget called 'Test and Scores' along with classification models. Results shows in the "Test and Score". More results like confusion matrix and ROC curve can be shown in widgets as well.

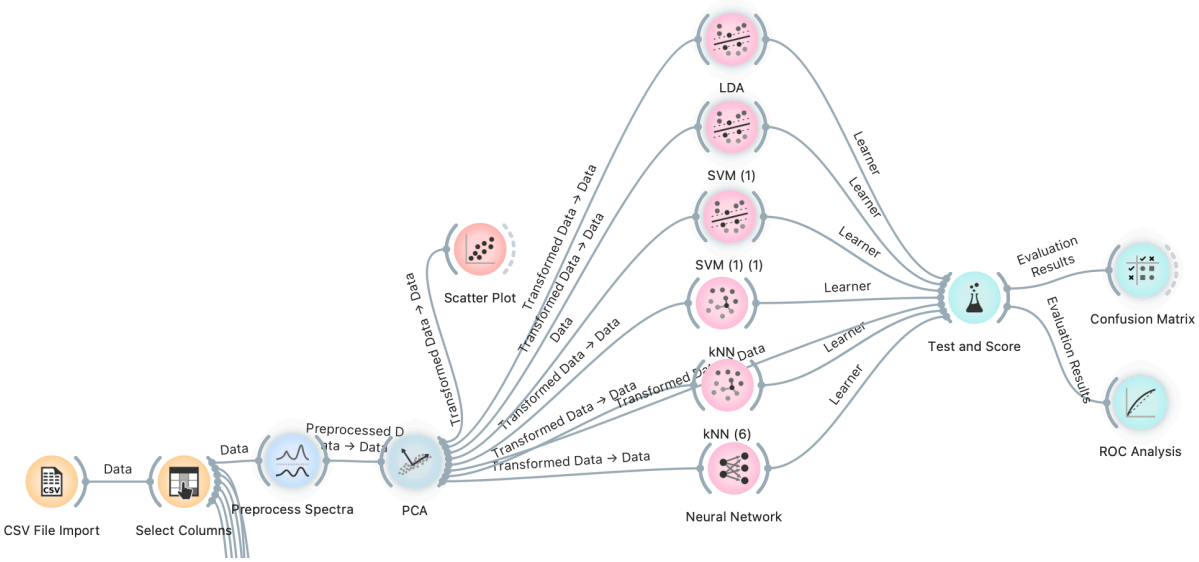

Figure 12: Orange Data Mining Interface

3.2 Pre-processing

In the stage of pre-processing, the first step is to cut the spectral data. Since other researchers used 1800 to 900 cm^{-1}, this project will use this cutting range as well. Meanwhile, since the ultimate goal of this project is to be able to diagnose cases in vivo, and the water band appears at around 1645 cm^{-1} and overpowers the whole spectrum, there needs a way to avoid it. Because the water band doesn't affect the signal from the smaller wavenumber region, this problem can be avoided by cutting the spectral data without Amide I, 1430 to 900 cm^{-1}. This project will test both the cutting range.

The second stage of pre-processing is baseline correction. In this stage, this project uses SG filter and Rubber band baseline correction. In SG filter, polynomial order includes 3, 5 and 7, derivative order includes 1 and 2. Polynomial order will be written in short like P3 and derivative order will be written in short like D1.

The third part of pre-processing is normalisation. In this stage, this project use Vector normalisation and Amide I normalisation. The Vector normalisation is only used after SG filter and Amide I normalisation is only used after Rubber band baseline correction.

The last part of pre-processing is PCA. It helps to reduce the dimensionality of data and prevent overfitting. This project uses the number of principal components that can explain 95% of variance.

3.3 Validation Method

This project uses both validation methods, k-fold validation and hold out validation. For K-fold validation, this project uses 10 folds in the validation. For hold out validation, this project uses two different training proportion, 70% and 80%. Figure 9 schematically shows the validation methods used in this project. After data is pro-processed, use three different methods to validate. The first validation method is 10-fold cross validation. The second and the third is hold out validation. In hold out validation, use 80% or 70% of data to train the model and use the rest of data to test the model.

3.4 Classifier

This project used LDA, KNN, SVM, neural network, logistic regression. LDA is the first choice of classifier in this project because LDA has the best performance according to literature.

3.5 Model Evaluation

This project evaluated the performance of models from three indices, accuracy, sensitivity and specificity.

3.6 Conclusion

Figure 13 schematically shows the flow chart of this project. After CSV file was imported, cut the spectra in either 1800 to 900 cm^{-1} or 1430 to 900 cm^{-1}. Then SG filter with polynomial order 3, 5 and 7, differentiation order one and two or Rubber band baseline correction was applied. After that, this project used Vector normalisation or Amide I normalisation. At the end of pre-processing, used PCA to reduce the dimensionality of the dataset.

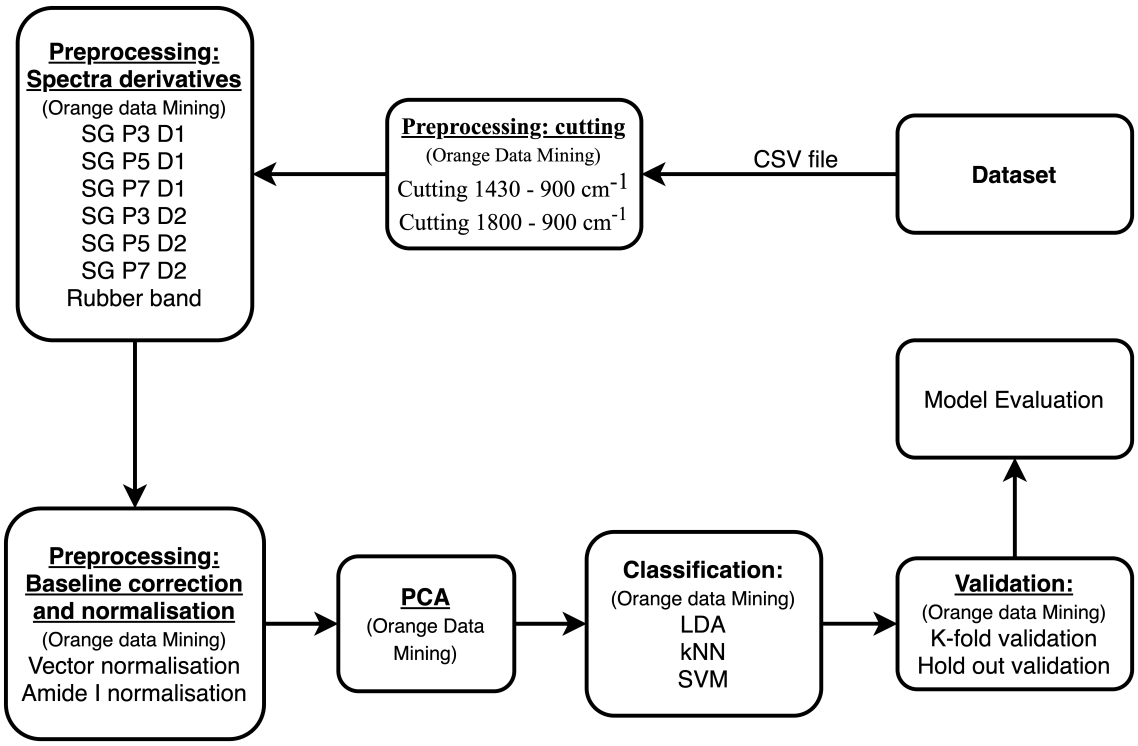

Figure 13: Project Flow Chart

4 Experiments and Results

All experiments were designed to find out the data analysis with both high accuracy and good generalization. Experiments of this project have three stages. The first stage was to test the feasibility of machine learning aided cancer diagnosis based on FTIR spectral data. The second stage of experiments was aimed to find out the best pre-processing method. The last stage of experiments was designed to evaluate the performance of models. All class means to use 6 class of brain dataset together. Table 3 shows the abbreviations used in this section.

Table 3: Abbreviations

Notion	Description	Notion	Description
GG	Glioma grade 3	NB	Normal Brain
MS	Metastasis	MA	Meningioma
GM	Glioblastoma Multiforme	LG	Low grade glioma
P5	Polynomial Order 5	D1	Derivative Order 1

4.1 All Class Validation

In this section, all classes of brain spectral data were used. The main goal of this section is to have a basic understanding of each classifiers' performance after different pre-processing method combination.

Pre-processing methods includes:

- 1800 to 900 cm^{-1}, Rubber band baseline correction, Amide I Normalisation
- 1800 to 900 cm^{-1}, Rubber band baseline correction, Vector Normalisation
- 1800 to 900 cm^{-1}, SG filter P5 D1, Amide I Normalisation
- 1800 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation
- 1430 to 900 cm^{-1}, Rubber band baseline correction, Amide I Normalisation
- 1430 to 900 cm^{-1}, Rubber band baseline correction, Vector Normalisation
- 1430 to 900 cm^{-1}, SG filter P5 D1, Amide I Normalisation
- 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

Figure 14 shows the effect of the pre-processing combination, 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. The spectral data shown on the top of Figure 13 is original brain spectral data with all six classes. The spectral data at the bottom of Figure 13 is the pre-processed data.

Classification model includes KNN, SVM, Logistic Regression and Neural Network.

In this experiment, the validation method was 10-fold cross Validation. Table 4 shows the results of all experiments in this section. The results includes the accuracy, sensitivity and specificity of models.

Table 4: Results of All class Validation

Model	**Accuracy**	**Sensitivity**	**Specificity**
1800 to 900 cm^{-1}, Rubber band baseline correction, Amide I Normalisation			

Model	Accuracy	Sensitivity	Specificity
KNN	0.818	0.818	0.962
SVM	0.812	0.812	0.941
Logistic Regression	0.506	0.506	0.821
Neural Network	0.952	0.952	0.987
1800 to 900 cm^{-1}, Rubber band baseline correction, Vector Normalisation			
KNN	0.836	0.836	0.965
SVM	0.814	0.814	0.947
Logistic Regression	0.432	0.432	0.770
Neural Network	0.954	0.954	0.988
1800 to 900 cm^{-1}, SG filter P5 D1, Amide I Normalisation			
KNN	0.811	0.811	0.958
SVM	0.829	0.829	0.949
Logistic Regression	0.659	0.659	0.893
Neural Network	0.942	0.942	0.985
1800 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation			
KNN	0.923	0.923	0.982
SVM	0.853	0.853	0.948
Logistic Regression	0.443	0.443	0.779
Neural Network	0.987	0.987	0.996
1430 to 900 cm^{-1}, Rubber band baseline correction, Amide I Normalisation			
KNN	0.857	0.857	0.968
SVM	0.669	0.669	0.873
Logistic Regression	0.548	0.548	0.852
Neural Network	0.949	0.949	0.988
1430 to 900 cm^{-1}, Rubber band baseline correction, Vector Normalisation			
KNN	0.858	0.858	0.971
SVM	0.798	0.798	0.936
Logistic Regression	0.412	0.412	0.773
Neural Network	0.944	0.944	0.986
1430 to 900 cm^{-1}, SG filter P5 D1, Amide I Normalisation			

Model	Accuracy	Sensitivity	Specificity
KNN	0.776	0.776	0.943
SVM	0.643	0.643	0.889
Logistic Regression	0.468	0.468	0.808
Neural Network	0.904	0.904	0.975
1430 to 900 cm^{-1},SG filter P5 D1, Vector Normalisation			
KNN	0.799	0.799	0.950
SVM	0.650	0.650	0.892
Logistic Regression	0.412	0.412	0.773
Neural Network	0.944	0.944	0.986

Figure 15 shows the confusion matrix of SVM, after the pre-processing of 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. From the confusion matrix, we can see the class MA was well classified. There are 300 Meningioma samples in total, 294 of which are correctly classified. GM performed the worst. There are 100 GM samples but only 68 of which were correctly classified. On the other hand, there are 300 MA samples but 368 samples were classified as MA. This indicated misclassification of samples to MA easily occurred.

Figure 16 shows the ROC curve of SVM, Neural Network and Logistic Regression, after the pre-processing of 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. The green curve was from Neural Network performed the best. The orange curve was from SVM, which performed the second best. The pink curve was from Logistic Regression, which performed the worst.

From the results, Neural Network performs the best in all cases while Logistic Regression performs the worst. This experiment verified that machine learning models could be used to help diagnose cancer based on brain FTIR data. In this study, accuracy reach over 70 and in some cases it can even reach over 90 while in clinical applications the minimum accuracy, sensitivity and specificity requirement is 75%. However, these experiments also have the following flaws:

1. It is hard to tell if these results were under good fitting range.

2. It is hard to give a solid conclusion which pre-processing method performs the best because large amount of classes were used in the experiment,

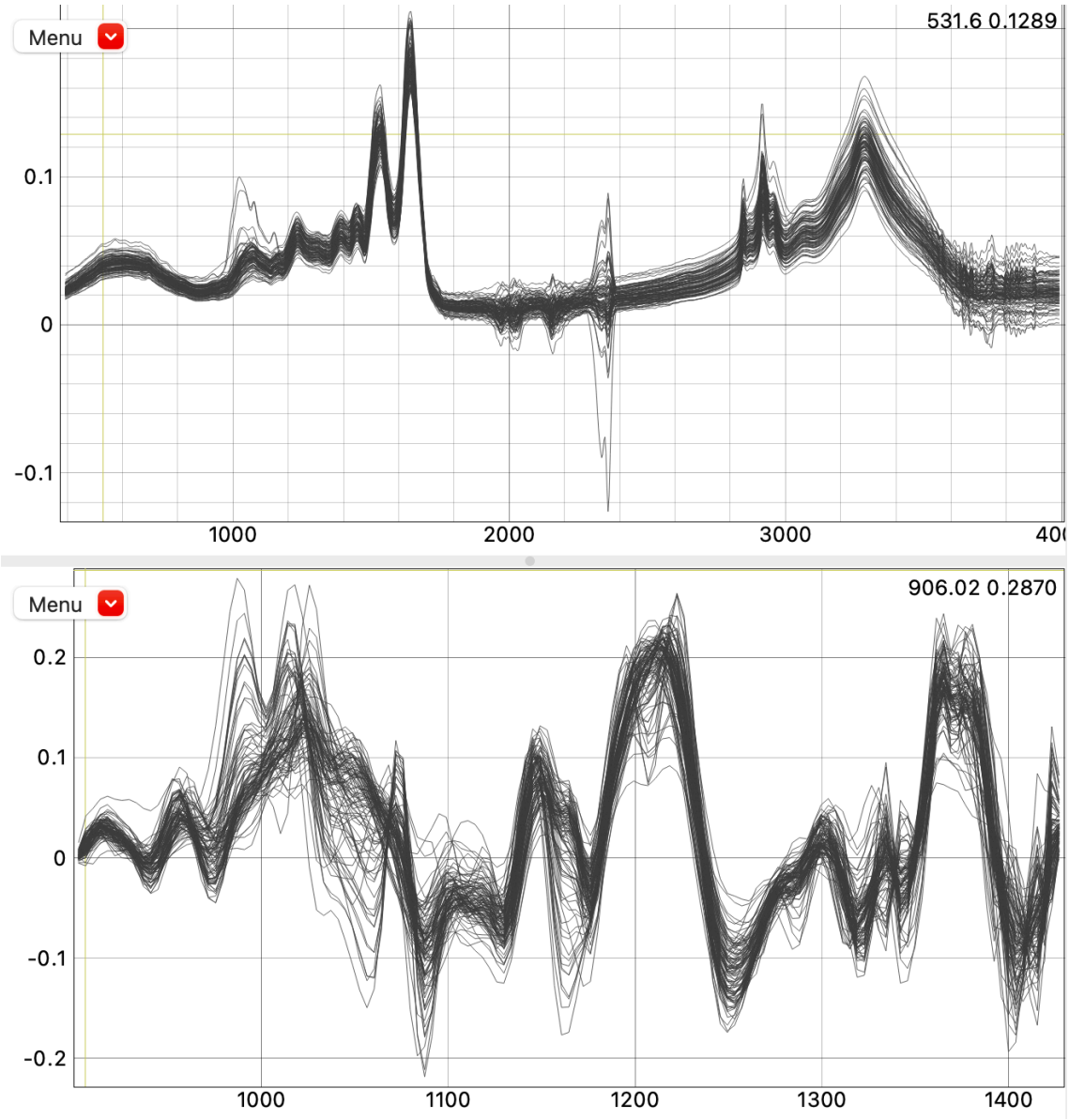

Figure 14: Effect of 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

In the next following sections, experiments were designed to solve these problems.

4.2 Find the Best Pre-processing Combination

The goal of this section is to find the best pre-processing method combination for most classifiers. First, this section only used two classes of brain spectral data, Meningioma and Normal brain data, instead of all six classes. Second, experiments in this section were designed by control variates method, to find the best pre-processing method combination. Third, more pre-processing method and classifiers were included.

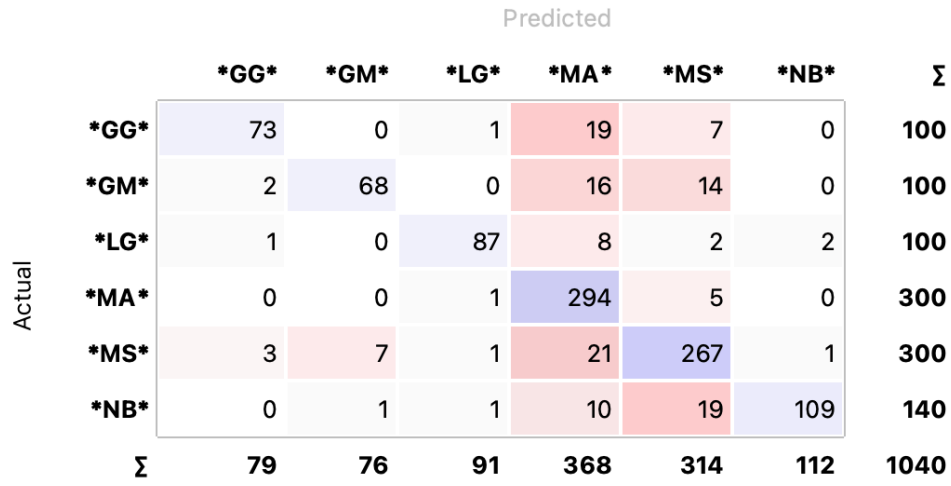

Figure 15: Confusion Matrix of SVM, after 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

The first group of the experiment is designed to assess how the polynomial and derivative order of the SG filter affects the classification rate of normal brain and Meningioma brain data in the range of 1430 to 900 cm^{-1}. Validation method used in this section is 10-fold cross validation. All the pre-processing method combination includes:

1. 1430 to 900 cm^{-1}, SG P3 D1, Vector Normalisation, PCA
2. 1430 to 900 cm^{-1}, SG P3 D2, Vector Normalisation, PCA
3. 1430 to 900 cm^{-1}, SG P5 D1, Vector Normalisation, PCA
4. 1430 to 900 cm^{-1}, SG P5 D2, Vector Normalisation, PCA
5. 1430 to 900 cm^{-1}, SG P7 D1, Vector Normalisation, PCA
6. 1430 to 900 cm^{-1}, SG P7 D2, Vector Normalisation, PCA

In all PCA methods, explained variance is the same, 95%, while the number of components is different.
Classification models includes:

1. LDA
2. Quadratic SVM
3. Cubic SVM

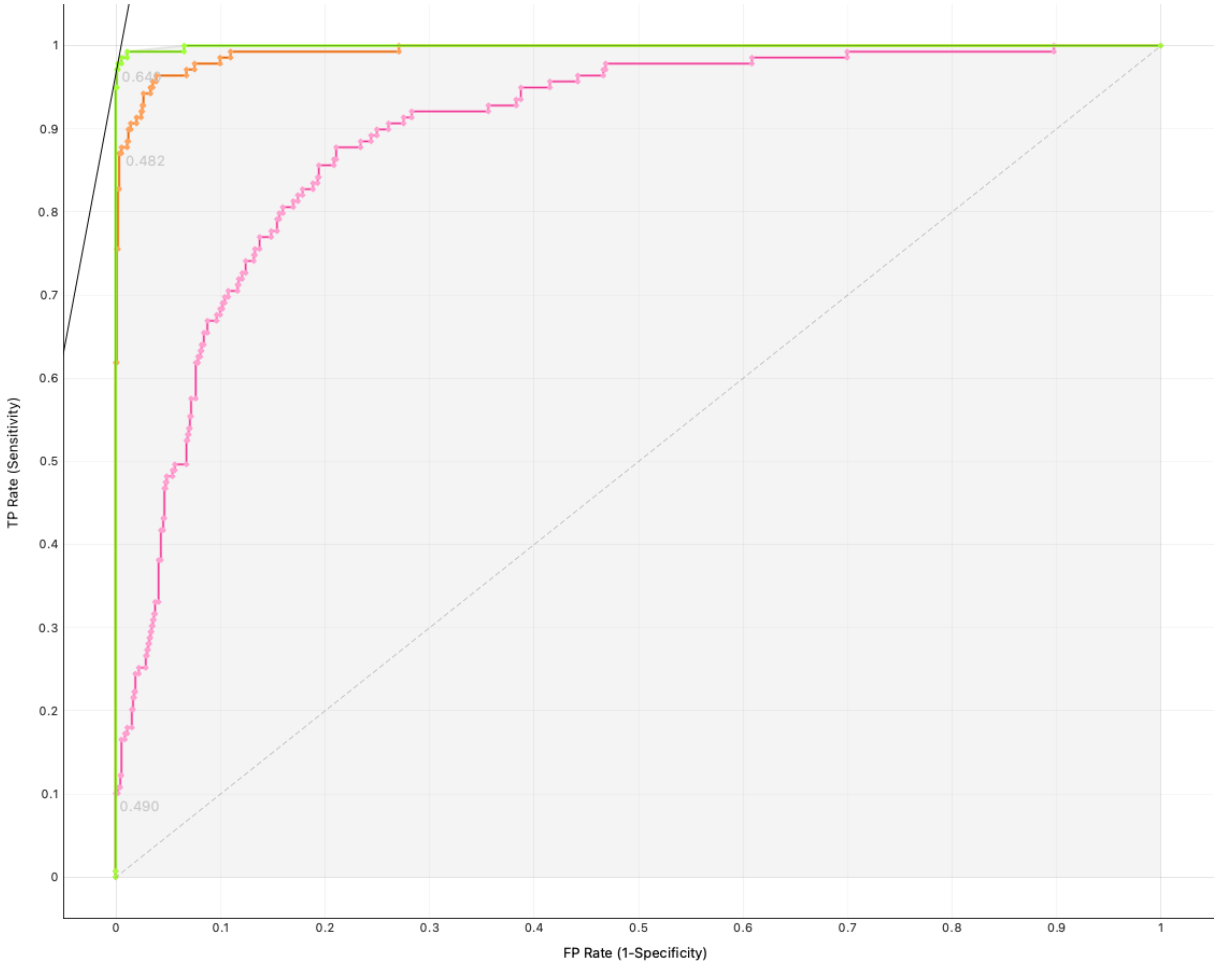

Figure 16: ROC curve of SVM, Neural Network and Logistic Regression, after the pre-processing of 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

4. Fine KNN (Number of Neighbors: 1)
5. Weighted KNN (Number of Neighbors: 10)
6. Neural Network (Three hidden layers with 300, 100 and 20 neurones in)

Presents methods includes:

1. Accuracy
2. Sensitivity
3. Specificity

Table 5 shows the results in this first group of experiment:

Table 5: Validation Results In the Range of 1430 to 900 cm^{-1}

Model	Accuracy	Sensitivity	Specificity
1430 to 900 cm^{-1}, SG P3 D1, Vector Normalisation, PCA			
LDA	0.573	0.573	0.622
Quadratic SVM	0.570	0.570	0.700
Cubic SVM	0.541	0.541	0.767
Fine KNN	0.805	0.805	0.730
Weighted KNN	0.811	0.811	0.634
Neural Network	0.893	0.893	0.84
1430 to 900 cm^{-1}, SG P3 D2, Vector Normalisation, PCA			
LDA	0.916	0.916	0.873
Quadratic SVM	0.973	0.973	0.945
Cubic SVM	0.977	0.977	0.951
Fine KNN	0.991	0.991	0.998
Weighted KNN	0.986	0.986	0.982
Neural Network	0.995	0.995	0.990
1430 to 900 cm^{-1}, SG P5 D1, Vector Normalisation, PCA			
LDA	0.939	0.939	0.922
Quadratic SVM	0.984	0.984	0.966
Cubic SVM	0.977	0.977	0.955
Fine KNN	0.995	0.995	0.994
Weighted KNN	0.986	0.986	0.975
Neural Network	0.993	0.993	0.989
1430 to 900 cm^{-1}, SG P5 D2, Vector Normalisation, PCA			
LDA	0.918	0.918	0.897
Quadratic SVM	0.973	0.973	0.945
Cubic SVM	0.975	0.975	0.946
Fine KNN	0.991	0.991	0.988
Weighted KNN	0.986	0.986	0.982
Neural Network	0.995	0.995	0.990
1430 to 900 cm^{-1}, SG P7 D1, Vector Normalisation, PCA			

Model	Accuracy	Sensitivity	Specificity
LDA	0.927	0.927	0.909
Quadratic SVM	0.984	0.984	0.970
Cubic SVM	0.975	0.975	0.950
Fine KNN	0.998	0.998	0.995
Weighted KNN	0.982	0.982	0.969
Neural Network	0.993	0.993	0.985
1430 to 900 cm^{-1}, SG P7 D2, Vector Normalisation, PCA			
LDA	0.923	0.923	0.880
Quadratic SVM	0.973	0.973	0.945
Cubic SVM	0.975	0.975	0.946
Fine KNN	0.991	0.991	0.988
Weighted KNN	0.986	0.986	0.982
Neural Network	0.995	0.995	0.990

To see the details behind results, confusion matrix is needed. Figure 17 shows the confusion matrix of Neural Network, after the pre-processing of 1800 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. This confusion matrix shows that within 300 MA samples, only one of them is misclassified. Within 140 normal brain samples, only 2 is misclassified.

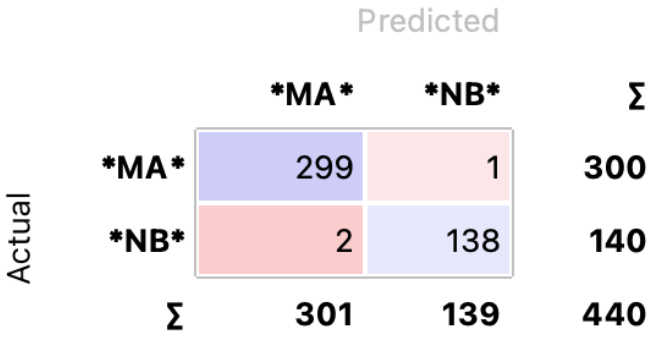

Figure 17: Confusion Matrix of Neural Network, after 1800 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

Figure 18 shows the confusion matrix of Cubic SVM, after the pre-processing combination of 1800 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. This confusion matrix shows all of

300 MA samples are correctly classified. Nine NB class samples are misclassified.

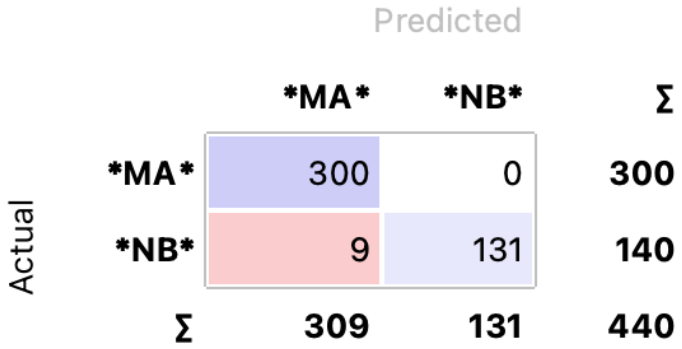

Figure 18: Confusion Matrix of Cubic SVM, after 1800 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

Figure 19 shows the confusion matrix of LDA, after the pre-processing combination of 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. This confusion matrix shows that within 300 MA samples, 19 of them is misclassified. Within 140 normal brain samples, 17 of them were misclassified.

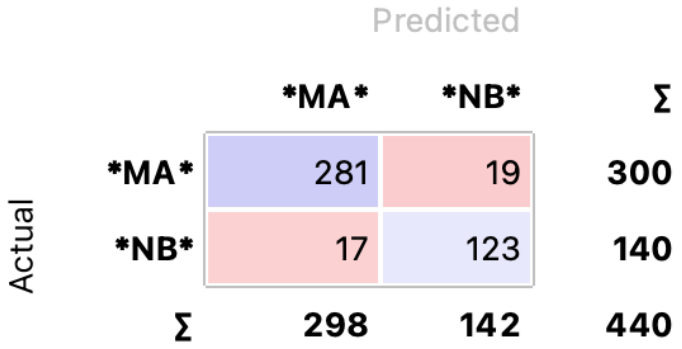

Figure 19: Confusion Matrix of LDA, after 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

Figure 20 shows the ROC curve of all classifiers after the pre-processing combination of 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation. This figure shows that, apart from LDA, all classifiers' performance is really close. LDA performs slightly worse than other classifiers but still good.

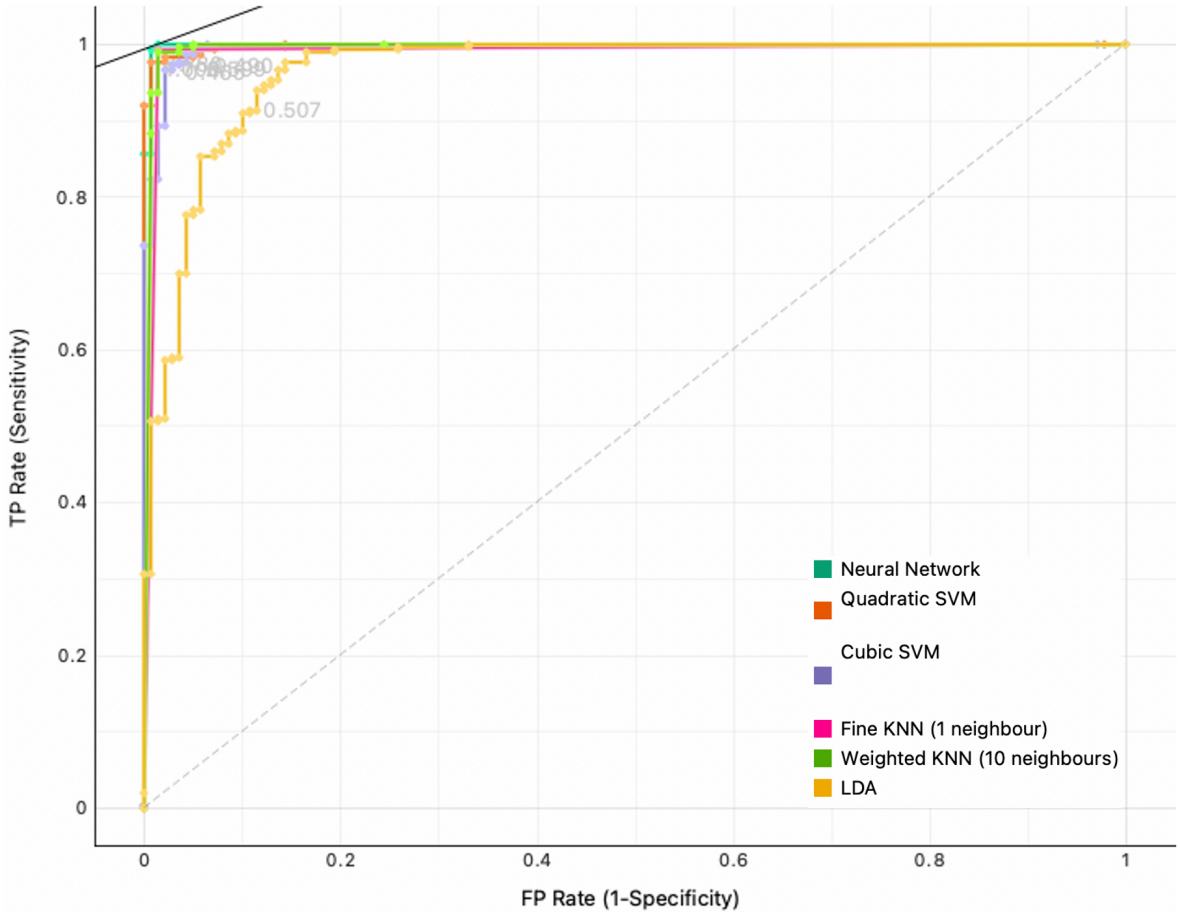

Figure 20: ROC curve of All Classifiers, after 1430 to 900 cm^{-1}, SG filter P5 D1, Vector Normalisation

This result shows that LDA has the worst performance. Neural Network has the best performance while Fine KNN and Weighted KNN has the second best performance. The best pre-processing combination is 1430 to 900 cm^{-1}, SG P5 D1, Vector Normalisation, PCA.

The second group of the experiment is designed to assess how the polynomial and derivative order of the SG filter affects the classification rate of normal brain and Meningioma brain data in the range of 1800 to 900 cm^{-1}. Validation method used in this section is 10-fold cross validation. All the pre-processing method combination includes:

1. 1800 to 900 cm^{-1}, SG P3 D1, Vector Normalisation, PCA
2. 1800 to 900 cm^{-1}, SG P3 D2, Vector Normalisation, PCA
3. 1800 to 900 cm^{-1}, SG P5 D1, Vector Normalisation, PCA

4. 1800 to 900 cm^{-1}, SG P5 D2, Vector Normalisation, PCA
5. 1800 to 900 cm^{-1}, SG P7 D1, Vector Normalisation, PCA
6. 1800 to 900 cm^{-1}, SG P7 D2, Vector Normalisation, PCA

Classification models includes:

1. LDA
2. Quadratic SVM
3. Cubic SVM
4. Fine KNN (Number of Neighbours: 1)
5. Weighted KNN (Number of Neighbours: 10)
6. Neural Network (Three hidden layers with 300, 100 and 20 neurones in)

Presents methods includes:

1. Accuracy
2. Sensitivity
3. Specificity

Table 6 shows the results in this second group of experiment:

Table 6: Validation Results In the Range of 1800 to 900 cm^{-1}

Model	**Accuracy**	**Sensitivity**	**Specificity**
1800 to 900 cm^{-1}, SG P3 D1, Vector Normalisation, PCA			
LDA	0.966	0.966	0.965
Quadratic SVM	0.984	0.984	0.977
Cubic SVM	0.982	0.982	0.961
Fine KNN	0.998	0.998	0.995
Weighted KNN	0.984	0.984	0.966
Neural Network	0.995	0.995	0.990
1800 to 900 cm^{-1}, SG P3 D2, Vector Normalisation, PCA			

Model	Accuracy	Sensitivity	Specificity
LDA	0.9139	0.939	0.922
Quadratic SVM	0.984	0.984	0.966
Cubic SVM	0.977	0.977	0.955
Fine KNN	0.995	0.995	0.994
Weighted KNN	0.986	0.986	0.975
Neural Network	0.993	0.993	0.989
1800 to 900 cm^{-1}, SG P5 D1, Vector Normalisation, PCA			
LDA	0.918	0.918	0.897
Quadratic SVM	0.973	0.973	0.945
Cubic SVM	0.975	0.975	0.946
Fine KNN	0.991	0.991	0.988
Weighted KNN	0.986	0.986	0.982
Neural Network	0.995	0.995	0.990
1800 to 900 cm^{-1}, SG P5 D2, Vector Normalisation, PCA			
LDA	0.927	0.927	0.909
Quadratic SVM	0.984	0.984	0.970
Cubic SVM	0.975	0.975	0.950
Fine KNN	0.998	0.998	0.995
Weighted KNN	0.982	0.982	0.969
Neural Network	0.993	0.993	0.985
1800 to 900 cm^{-1}, SG P7 D1, Vector Normalisation, PCA			
LDA	0.923	0.923	0.880
Quadratic SVM	0.973	0.973	0.945
Cubic SVM	0.975	0.975	0.946
Fine KNN	0.991	0.991	0.988
Weighted KNN	0.986	0.986	0.982
Neural Network	0.995	0.995	0.990
1800 to 900 cm^{-1}, SG P7 D2, Vector Normalisation, PCA			
LDA	0.993	0.993	0.993
Quadratic SVM	0.982	0.982	0.972
Cubic SVM	0.984	0.984	0.966

Model	Accuracy	Sensitivity	Specificity
Fine KNN	0.995	0.995	0.990
Weighted KNN	0.986	0.986	0.971
Neural Network	0.998	0.998	0.995

After all of the experiments are finished, the details are further discussed as the following. To compare the performance of two cutting range, 1800 to 900 cm^{-1} and 1430 to 900 cm^{-1}, the highest accuracy and average accuracy of each pre-processing combination are listed as shown in Table 7. For instance, the highest accuracy that classifiers can get after 1430-900 cm^{-1},SG P3 D1, Vector Normalisation, PCA is 0.893.

Table 7: Comparison of Validation Results in Accuracy Among All Pre-processing Methods

Pre-processing	Highest	Average	Highest	Average
	1430-900 cm^{-1}		**1800-900 cm^{-1}**	
SG P3 D1, Vector, PCA	0.893	0.698	0.995	0.973
SG P3 D2, Vector, PCA	0.995	0.973	0.995	0.979
SG p5 D1, Vector, PCA	0.995	0.979	0.995	0.973
SG P5 D2, Vector, PCA	0.995	0.973	0.998	0.977
SG P7 D1, Vector, PCA	0.998	0.977	0.995	0.974
SG P7 D2, Vector, PCA	0.995	0.974	0.998	0.983

Table 7 shows that the pre-processing method combination can get the highest accuracy is SG P5 D1, Vector Normalisation, PCA.

Table 8 shows the average and highest accuracy in two cutting ranges. The highest accuracy in both 1800-900 cm^{-1} and 1430-900 cm^{-1} is the same, which is 0.998. The average of all the highest accuracy in two cutting ranges is also the same, 0.979. However, the highest accuracy in the cutting range of 1800-900 cm^{-1}, 0.996, is higher than the highest accuracy in the cutting range of 1430-900 cm^{-1}, 0.978.

Table 8: Comparison of Validation Results in Accuracy Between Cutting Range of 1800 to 900 cm^{-1} with 1430 to 900 cm^{-1}

Pre-processing	**Highest**	**Average**	**Highest**	**Average**
	1430-900 cm^{-1}		**1800-900 cm^{-1}**	
Average Accuracy	0.978	0.929	0.996	0.976
Highest Accuracy	0.998	0.979	0.998	0.979

This shows that both cutting ranges, 1800-900 cm^{-1} and 1430-900 cm^{-1}, perform very good. The very little performance difference between the two cutting range makes it ignorable. Moreover, for this reason, this project will only use 1430-900 cm^{-1} in the next section. Thus, in the next section, the only pre-processing method combination will use is 1430-900 cm^{-1}, SG P5 D1, Vector Normalisation, PCA.

4.3 Find the Good Fitting Range

As showed in Figure 21, to find the good fitting range, training error and testing error are needed. In this section, the main goal is to find the good fitting range of each classification model. To achieve this goal, this section will use hold out validation.

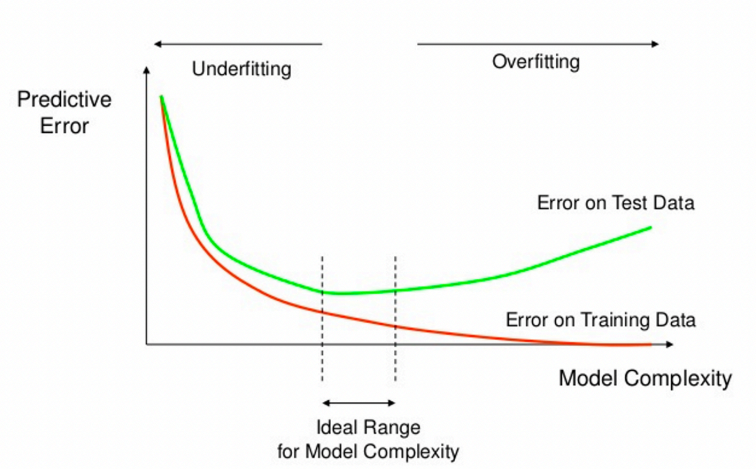

Figure 21: Overfitting and Underfitting [19]

The samples used in this section includes:

1. Meningioma (300 samples from 15 patients)

2. Normal Brain (140 samples from 7 patients)

Figure 22 schematically shows the workflow of pre-processing in this section. The pre-processing method included cutting the spectra from 1430 cm^{-1} to 1430 cm^{-1}, SG filter with Polynomial 5 and Derivative order 1, Vector Normalisation and PCA.

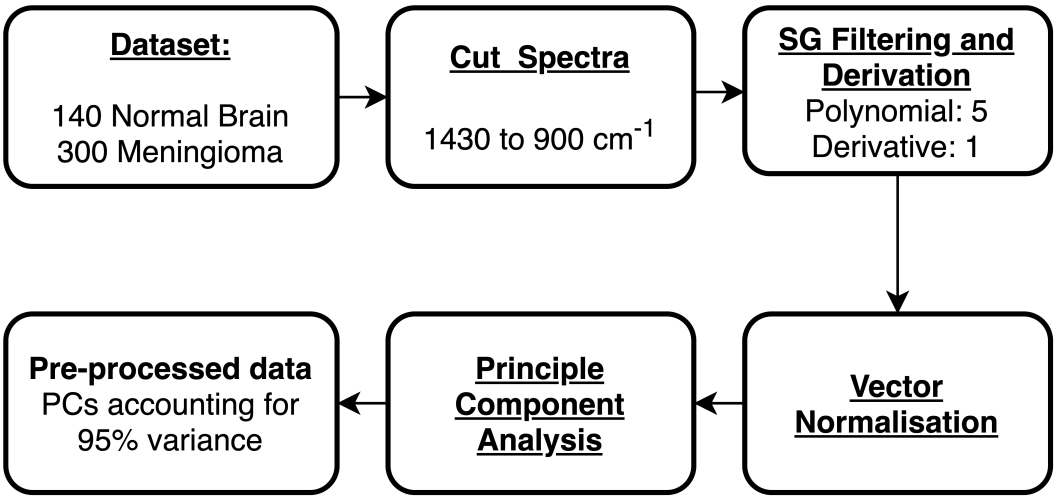

Figure 22: Pre-processing Workflow

Since 20 spectral data were collected from each patient, it was inappropriate to divide all the data into training data and testing data at random. The reason was that the two parts of data should be completely independent, otherwise the performance would be affected. To balance the classifiers' performance on both data class, this section used Meningioma and normal brain from the same amount of patients. As the dataset includes 7 patients' normal brain spectra and 15 patients' Meningioma spectra, it was reasonable to use 5 patients each to train the model and 2 patients each to test or 6 patients to train and 1 patient each to test. Figure 22 shows two validation methods used in this section. As Figure 23 (a) shown on the top, the first hold out validation method use 5 Meningioma patients and 5 normal brain patients' spectra as training data to train the models, use 2 Meningioma and 2 normal brain spectra as testing data (5+2 Validation). As Figure 23 (b) shown at the bottom, the second hold out validation method use 6 Meningioma and 6 normal brain as training data and 1 Meningioma and 1 normal brain as testing data (6+1 Validation). After getting the results, this project created the diagram to show the relationship between model complexity and classification error.

The classification models used in this section includes:

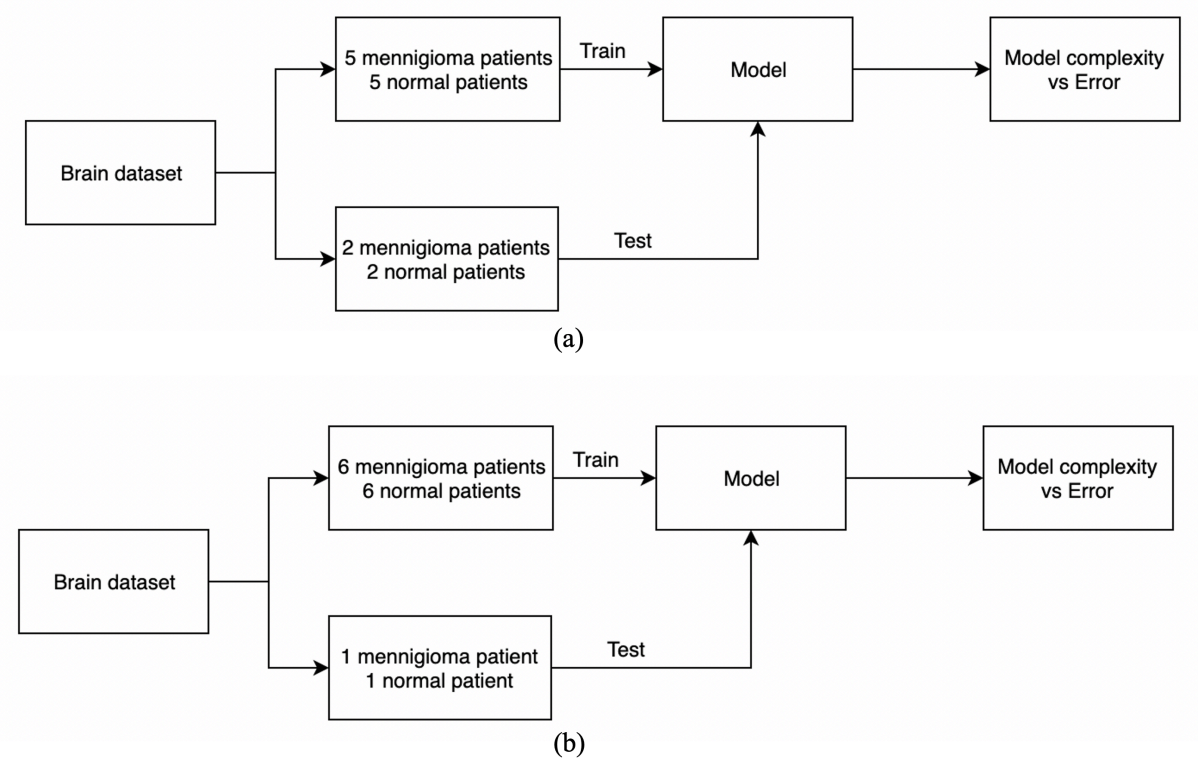

Figure 23: Validation Method to Find Goof Fitting Range

1. Logistic Regression
2. SVM (d = 1-10)
3. Fine KNN (Number of Neighbours: 1-30, 40, 50, 60, 80, 100)

The number of d decides the complexity of SVM as shown in Equation 10. The number of d is an integer from 1 to 10. Thus, to find the good fitting range of SVM, this project used the d from 1 to 10. The number of neighbours decides the complexity of KNN. This project used number of neighbours from 1 to 30 continuously, plus 40, 50, 60, 80 and 100.

Figure 24 shows the the relationship between model complexity and error of SVM from the 5+2 validation. The error is defined by $(1 - Accuracy)$. The green line is SVM's testing error and the blue line is SVM's training error. The grey line is Logistic Regression's training error and the red line is Logistic Regression's testing error. When number of d increase on x-axis from left to right, the model complexity increase. Figure 23 shows SVM's training error increase continuously and testing error go ups and downs when model complexity increase. This result is not match the ideal case shown in Figure 21.

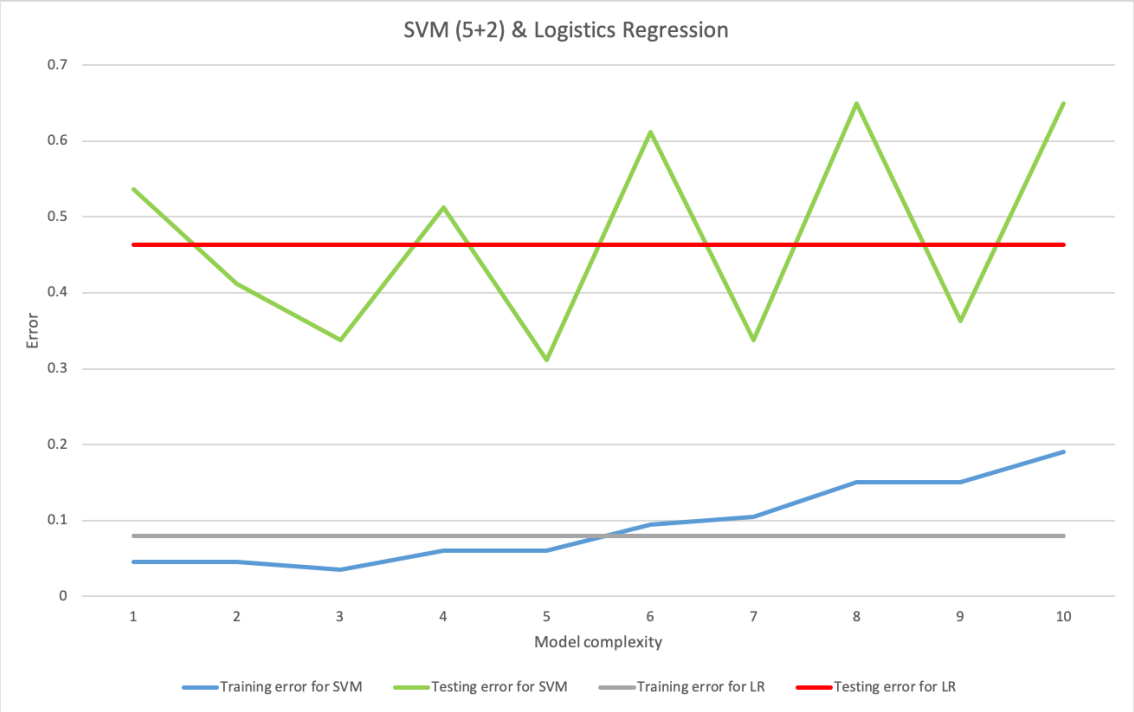

Figure 24: Relationship between Model Complexity and Classification Error from SVM (5+2 Validation)

Figure 25 shows the SVM's ROC curve on testing Meningioma data. The green ROC curve is from SVM with $d = 1$, orange curve is from SVM with $d = 2$ and purple curve is from SVM with $d = 3$. The pink curve is form Logistic Regression. From these ROC curves, it can tell four classifiers have similar performance.

Figure 26 shows the relationship between model complexity and error rate of Fine KNN from 5+2 validation. The error is defined by $(1 - Accuracy)$. The orange curve is Fine KNN's testing error and the blue curve is Fine KNN's training error. The yellow curve is Logistic Regression's testing error and the grey curve is Logistic Regression's training error. When the number of neighbours decreases on the x-axis from left to right, the model complexity increase. The error is defined by $(1 - Accuracy)$. In this case, when Fine KNN's model complexity increase, training error keep falling after 50 neighbours. However, testing error did not show clear patterns. Fine KNN's result shown in Figure 25 is not ideal compare to Figure 21.

Figure 27 shows the relationship between model complexity and error rate of SVM from 6+1 validation. The error was defined by $(1 - Accuracy)$. The orange curve and the blue curve were SVM's testing error and training error. The yellow curve and the grey curve were Logistic

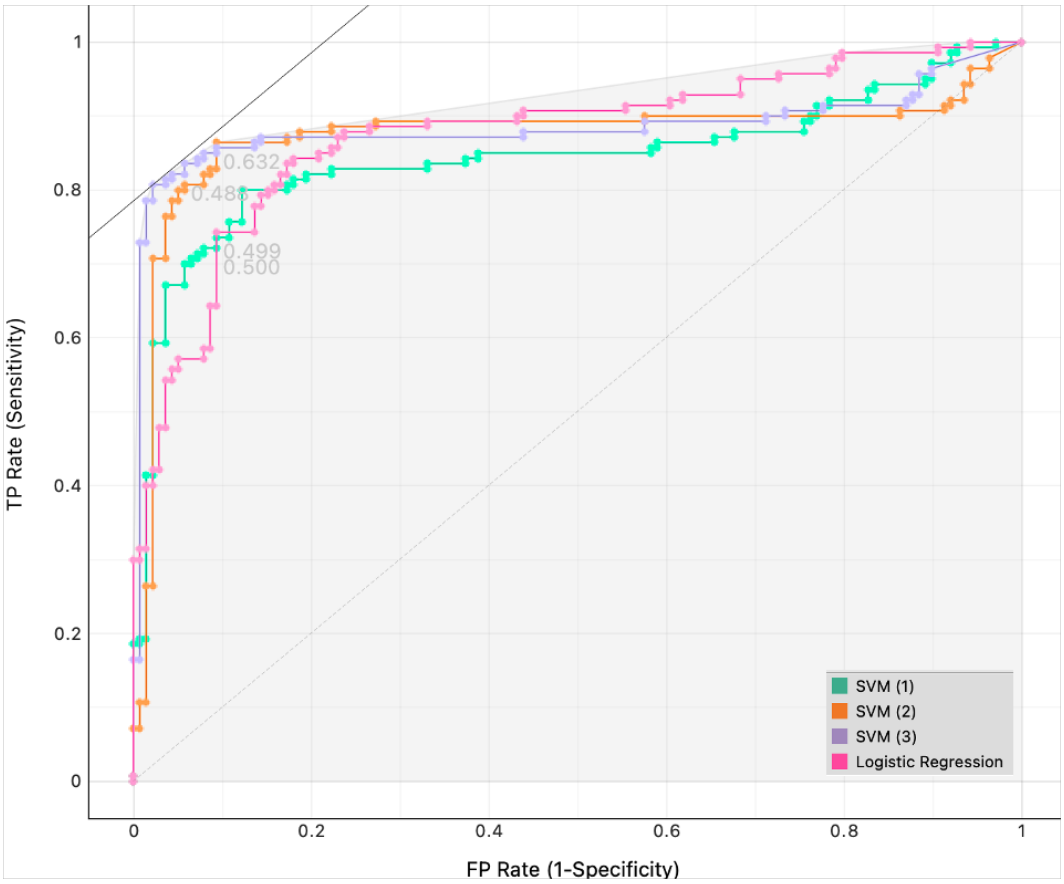

Figure 25: SVM's ROC Curve on Testing Meningioma Data from 5+2 Validation

Regression's testing error and training error. When model complexity increases, the overall trend of SVM's testing error was first to decrease then increase which matches the ideal case. However, the training error of SVM did not match the ideal case shown in Figure 21 as it increased instead of decrease.

The relationship between model complexity and error rate of Fine KNN from 6+1 validation are shown in Figure 27. The error was defined by $(1 - Accuracy)$. The blue curve and the orange curve were Fine KNN's testing error and training error. The yellow curve and the grey curve were Logistic Regression's testing error and training error. When model complexity increases, the overall trend of SVM's testing error was first to decrease then increase which matches the ideal case. Training error of Fine KNN also fits the ideal case. However, the curve of Fine KNN's training error and testing error are noisy. For instance, when number of neighbours is smaller than 13, testing error

The ROC curve of Fine KNN with the number of neighbours from 1 to 10 and Logistic

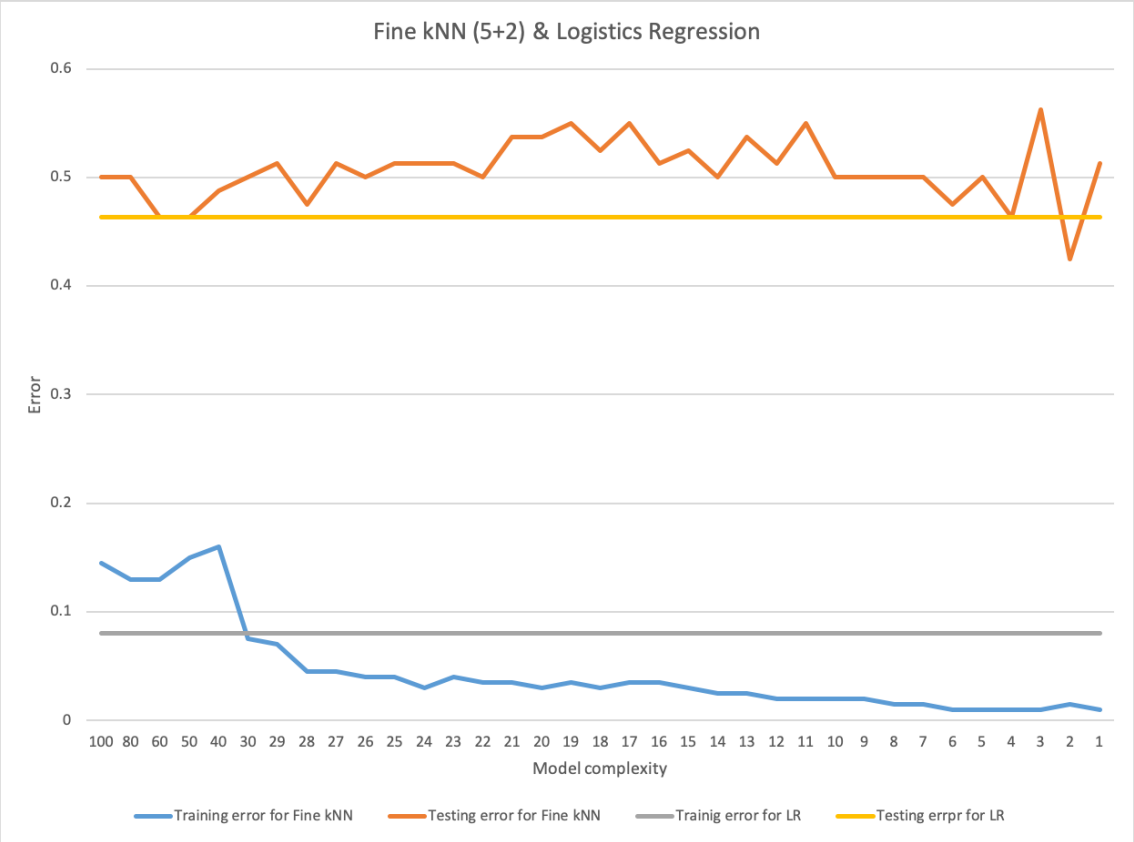

Figure 26: Relationship between Model Complexity and Classification Error from Fine KNN (5+2 Validation)

Regression was shown in Figure 29. The brown curve belonged to Logistic Regression and it had the worst performance. Fine KNN with the number of neighbours from 1 t0 10 had a similar performance.

5 Discussion

The results of this project were significant. Almost all classifiers in this project had over 90% accuracy, sensitivity and specificity. Moreover, classifiers' ignorable performance difference after two cutting range, 1800 to 900 cm^{-1} and 1430 to 900 cm^{-1}, indicated it is possible to diagnose cancer based on FTIR spectral data in vivo.

Nonetheless, when it comes to clinical application, the models were not ready to be used because underfitting and overfitting status of classifiers was unclear.

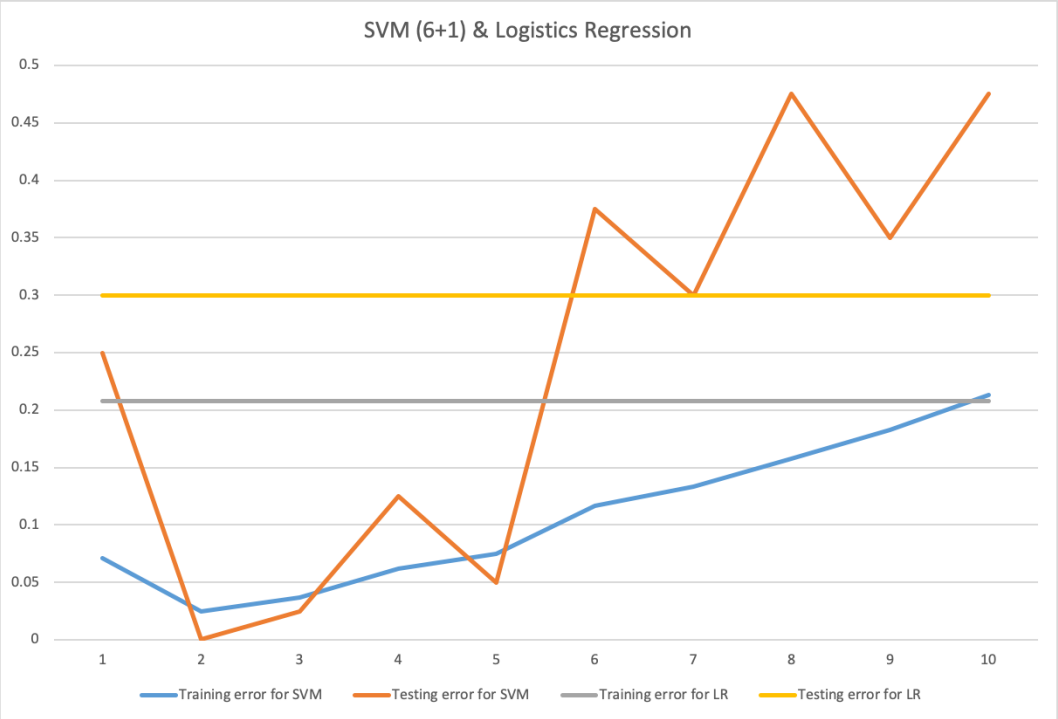

Figure 27: Relationship between Model Complexity and Classification Error from SVM (6+1 Validation)

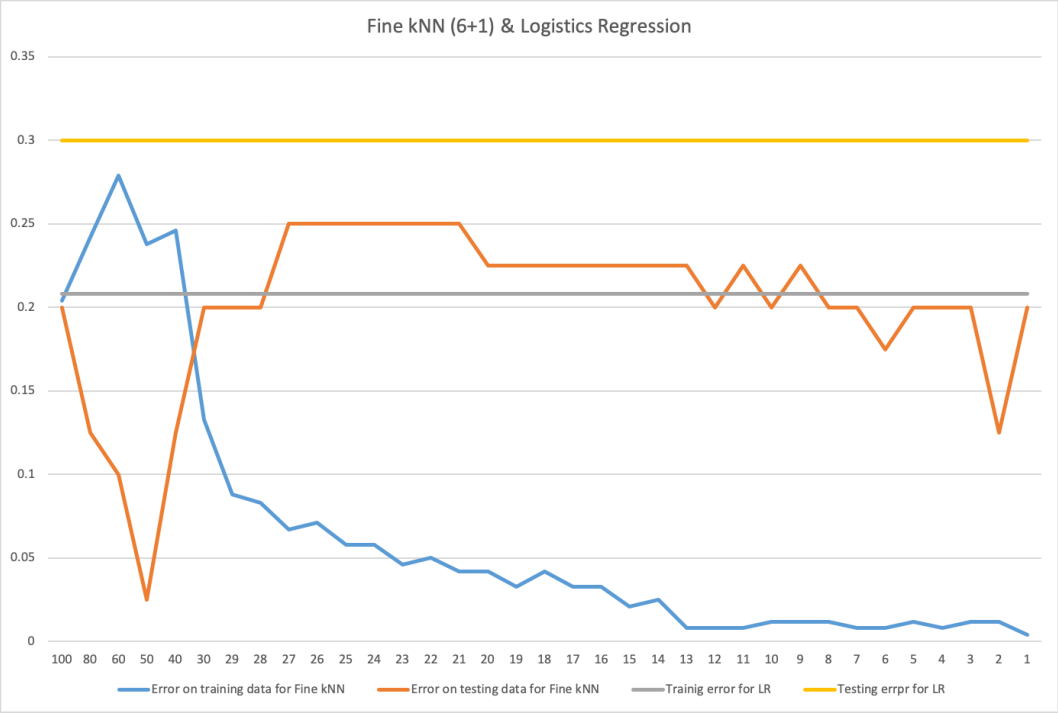

Figure 28: Relationship between Model Complexity and Classification Error from Fine KNN (6+1 Validation)

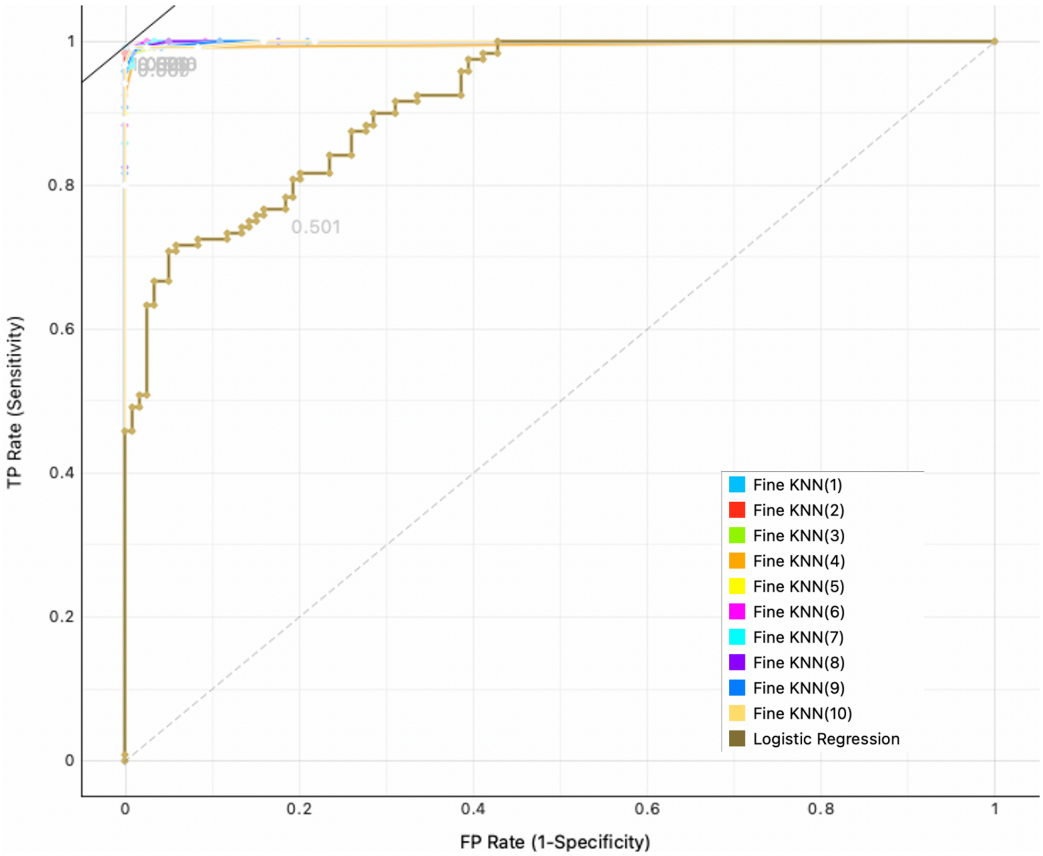

Figure 29: ROC Curve of Fine KNN (6+1 Validation)

6 Conclusion

This project verified the feasibility of machine learning aided cancer diagnosis based on FTIR spectral data. This project found the best pre-processing method which is cutting the spectra from 1430 to 900 cm^{-1}, applying the SG filter with polynomial order 5 and derivative order 1, then using Vector Normalisation and PCA. This project tested the performance of LDA, KNN, SVM, Neural Network and Logistic Regression when classifying the brain cancer and normal brain tissue. LDA reached 91.8% accuracy and sensitivity and 89.7% specificity. SVM had 98.4% accuracy and sensitivity and 96.6% specificity. KNN had 99.5% accuracy and sensitivity and 99.4% specificity. All classifiers had the result higher than 75% which is the least requirement for clinical application. Although LDA's performance was not as good as other classifiers', it did not have the overfitting problem. However, this study could not give a solid conclusion on whether KNN and SVM were under good fitting range or not due to the limitation of the dataset.

The recommendation of future work in this field are:

1. **Use the dataset which includes samples from more patients.** This project failed to give a solid conclusion on whether the models were overfitting or underfitting because of the limitation of the dataset. If there is a larger dataset, it will be easier to find the good fitting range of models. Moreover, models can gain better generalization ability.

2. **Experiment more types of cancer dataset.** This project only tested brain cancer. However, there are many more types of cancer that are threatening human's lives. Different types of cancer may need different pre-processing and different classifiers. There is so much work need to do in this field before this technology can be used in hospitals to save people's lives.

7 Acknowledgement

Thank you to my supervisor, Dr Sendy Phang, for giving me the chance to participate in this amazing project and answering all my questions patiently all the time. Thank you to other professors and doctors in this research group, especially Dr Angela Seddon, for pointing out my mistakes and places I need to pay attention to help me improve. Thank you to David Mabwa, the PhD student in this project who gave me many useful advises. Thank you to you all. It is my pleasure to work with such a group of brilliant doctors and professors.

References

[1] L. Siegel Rebecca and M. K. D. J. Ahmedin, "Cancer statistics, 2020.[j]," *CA. Cancer J Clin*, vol. 70, pp. 7–30, 2020.

[2] K. Honda, V. A. Katzke, A. Hüsing, S. Okaya, H. Shoji, K. Onidani, A. Olsen, A. Tjønneland, K. Overvad, E. Weiderpass, *et al.*, "Ca19-9 and apolipoprotein-a2 isoforms as detection markers for pancreatic cancer: a prospective evaluation," *International journal of cancer*, vol. 144, no. 8, pp. 1877–1887, 2019.

[3] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, "Compressed sensing mri reconstruction using a generative adversarial network with a cyclic loss," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1488–1497, 2018.

[4] I. Pupeza, M. Huber, M. Trubetskov, W. Schweinberger, S. A. Hussain, C. Hofer, K. Fritsch, M. Poetzlberger, L. Vamos, E. Fill, *et al.*, "Field-resolved infrared spectroscopy of biological systems," *Nature*, vol. 577, no. 7788, pp. 52–59, 2020.

[5] M. J. Baker, J. Trevisan, P. Bassan, R. Bhargava, H. J. Butler, K. M. Dorling, P. R. Fielden, S. W. Fogarty, N. J. Fullwood, K. A. Heys, *et al.*, "Using fourier transform ir spectroscopy to analyze biological materials," *Nature protocols*, vol. 9, no. 8, p. 1771, 2014.

[6] B. Bird, K. Bedrossian, N. Laver, M. Miljković, M. J. Romeo, and M. Diem, "Detection of breast micro-metastases in axillary lymph nodes by infrared micro-spectral imaging," *Analyst*, vol. 134, no. 6, pp. 1067–1076, 2009.

[7] D. Naumann, P. Lasch, and H. Fabian, "Cells and biofluids analyzed in aqueous environment by infrared spectroscopy," in *Biomedical vibrational spectroscopy III: Advances in research and industry*, vol. 6093, p. 609301, International Society for Optics and Photonics, 2006.

[8] M. J. Walsh, S. E. Holton, A. Kajdacsy-Balla, and R. Bhargava, "Attenuated total reflectance fourier-transform infrared spectroscopic imaging for breast histopathology," *Vibrational Spectroscopy*, vol. 60, pp. 23–28, 2012.

[9] G. J. Ooi, J. Fox, K. Siu, R. Lewis, K. R. Bambery, D. McNaughton, and B. R. Wood, "Fourier transform infrared imaging and small angle x-ray scattering as a combined

biomolecular approach to diagnosis of breast cancer," *Medical physics*, vol. 35, no. 5, pp. 2151–2161, 2008.

[10] G. R. Lloyd and N. Stone, "Method for identification of spectral targets in discrete frequency infrared spectroscopy for clinical diagnostics," *Applied spectroscopy*, vol. 69, no. 9, pp. 1066–1073, 2015.

[11] C. L. Morais, K. M. Lima, M. Singh, and F. L. Martin, "Tutorial: multivariate classification for vibrational spectroscopy in biological samples," *Nature Protocols*, pp. 1–20, 2020.

[12] M. J. Baker, H. J. Byrne, J. Chalmers, P. Gardner, R. Goodacre, A. Henderson, S. G. Kazarian, F. L. Martin, J. Moger, N. Stone, *et al.*, "Clinical applications of infrared and raman spectroscopy: state of play and future challenges," *Analyst*, vol. 143, no. 8, pp. 1735–1757, 2018.

[13] M. Paraskevaidi, C. L. Morais, O. Raglan, K. M. Lima, E. Paraskevaidis, P. L. Martin-Hirsch, M. Kyrgiou, and F. L. Martin, "Aluminium foil as an alternative substrate for the spectroscopic interrogation of endometrial cancer," *Journal of biophotonics*, vol. 11, no. 7, p. e201700372, 2018.

[14] M. Sadeghi, F. Behnia, and R. Amiri, "Window selection of the savitzky-golay filters for signal recovery from noisy measurements," *IEEE Transactions on Instrumentation and Measurement*, 2020.

[15] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures.," *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[16] J. Chen, P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh, "A simple method for reconstructing a high-quality ndvi time-series data set based on the savitzky–golay filter," *Remote sensing of Environment*, vol. 91, no. 3-4, pp. 332–344, 2004.

[17] K. S. Chia, H. A. Rahim, and R. A. Rahim, "Evaluation of common pre-processing approaches for visible (vis) and shortwave near infrared (swnir) spectroscopy in soluble solids content (ssc) assessment," *Biosystems engineering*, vol. 115, no. 1, pp. 82–88, 2013.

[18] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[19] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial," *arXiv preprint arXiv:1905.12787*, 2019.

[20] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[21] F. Marini, "Classification methods in chemometrics," *Current Analytical Chemistry*, vol. 6, no. 1, pp. 72–79, 2010.

[22] J. Grus, *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

[23] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI communications*, vol. 30, no. 2, pp. 169–190, 2017.

[24] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[25] M. Z. Gashti, "A modified model based on flower pollination algorithm and k-nearest neighbor for diagnosing diseases," *IIUM Engineering Journal*, vol. 19, no. 1, pp. 144–157, 2018.

[26] S. A. Ghauri, "Knn based classification of digital modulated signals," *IIUM Engineering Journal*, vol. 17, no. 2, pp. 71–82, 2016.

[27] A. ALI, M. ALRUBEI, L. F. M. HASSAN, M. AL-JA, and S. ABDULWAHED, "Diabetes classification based on knn,"

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[29] P. Konar and P. Chattopadhyay, "Bearing fault detection of induction motor using wavelet and support vector machines (svms)," *Applied Soft Computing*, vol. 11, no. 6, pp. 4203–4211, 2011.

[30] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[31] F.-F. Li, A. Karpathy, and J. Johnson, "Convolutional neural networks for visual recognition," 2015.

[32] J. Luts, F. Ojeda, R. Van de Plas, B. De Moor, S. Van Huffel, and J. A. Suykens, "A tutorial on support vector machine-based methods for classification problems in chemometrics," *Analytica Chimica Acta*, vol. 665, no. 2, pp. 129–145, 2010.

[33] N. J. Perkins and E. F. Schisterman, "The inconsistency of optimal cutpoints obtained using two criteria based on the receiver operating characteristic curve," *American journal of epidemiology*, vol. 163, no. 7, pp. 670–675, 2006.

[34] K. Babalyan, R. Sultanov, E. Generozov, E. Sharova, E. Kostryukova, A. Larin, A. Kanygina, V. Govorun, and G. Arapidi, "Logloss-beraf: An ensemble-based machine learning model for constructing highly accurate diagnostic sets of methylation sites accounting for heterogeneity in prostate cancer," *PloS one*, vol. 13, no. 11, p. e0204371, 2018.

[35] J. D. RodríGuez, A. Pérez, and J. A. Lozano, "A general framework for the statistical analysis of the sources of variance for classification error estimators," *Pattern recognition*, vol. 46, no. 3, pp. 855–864, 2013.