WQD7006 Machine Learning For Data Science

Group Project

Diabetic Retinopathy Detection using Machine Learning

Group Members

Cheah Jun Yitt	WQD180107
Chong Wei Hong	WQD180010
Choo Jian Wei	WQD180124
Tan Yin Yen	WQD180108

Table of Contents

CHAPTER 1: INTRODUCTION

1.1 Background

Diabetic retinopathy (DR) is an eye disease that causes damage to the retina of the diabetic mellitus patients. Diabetic eye disease is a leading cause of vision impairment and is caused by an elevated level of blood sugar, damaging the vessels that provide oxygen to the retina. An early detection of DR is important in preventing vision loss. However, the lack of ophthalmologists and absence of visual symptoms are major hurdles for early detection of DR. Hence, it is important to develop an automated screening system to predict DR. In recent years, many research was conducted on automated detection of DR. Different symptoms of DR have been used to detect the disease. Theera-Umpon et al. (2019) used exudates detection in the proposed work. Akram et al. (2014) detected DR based on three types of lesions, which are exudate, hemorrhage and microaneurysm. All the lesions appeared with different properties. Exudate is whitish and yellowish in colour, hemorrhage is a medium-sized dark red dot and microaneurysm is a circular-shaped red dot.

1.2 Objectives

The main objective of this project is to develop an automated DR detection system using different machine learning algorithms based on the features extracted from retinal images. This project also aims to compare the performance of each approach in terms of precision, recall, F1 score and accuracy to identify the best model.

1.3 Domain

This project involves classification of samples into presence of DR and absence of DR based on the features extracted.

CHAPTER 2: LITERATURE REVIEW

In this section, computer-aided DR diagnosis systems developed using various machine learning algorithms in previous studies will be discussed. Previous studies detected the presence of DR based on different symptoms such as microaneurysms, exudates and hemorrhages.

2.1 Predictive Modelling

In predictive modelling, the trained model is anticipated to capture the relationships between variables which are beneficial for DR prediction. Theera-Umpon et al. (2019) performed classification using four supervised learning algorithms, namely support vector machine (SVM), multilayer perceptron network (MLP), hierarchical adaptive network-based fuzzy inference system (ANFIS) and convolutional neural network (CNN). Manual feature extraction was not required for CNN as CNN took the entire retinal images as input and developed its own features. Akram et al. (2014) used a hybrid classifier with Gaussian Mixture Model (GMM) to classify the candidate regions as bright and dark lesions in accordance with the distribution, and M-Mediods based classifier to classify the lesions.

2.2 Discussion of Strength and Weakness

The strengths and weaknesses of the related works are discussed in this section. The study by Theera-Umpon et al. (2019) presented four predictive models, where MLP achieved the best result. Although the performance of MLP was the highest, it required complex manual extraction of the features. On the other hand, CNN did not require the manual feature extraction. However, CNN yielded the lowest performance as it could not work well when the edges of the exudates were not well defined. Besides that, it used only a specific symptom for the detection, which was exudate, which might appear to be very similar but being the symptoms of different disease. For instance, Age-related Macular Degeneration (AMD) which also causes vision loss is detected based on drusen, a yellowish deposit in the retina (Sidibé et al., 2015). In the study conducted by Akram et al. (2014), the results showed that the proposed method performed well and could be used retinal image grading. However, there existed a drawback in the blood vessel segmentation using wavelet transformation as the approach was exhausting and computationally complex (Amin et al., 2017).

Table 2.1: Summarisation of related works

Related Works	**Predictive Modelling**	**Symptoms Used**	**Pros**	**Cons**
Theera-Umpon et al. (2019)	• SVM • MLP network • H-ANFIS • CNN	Exudate	• MLP achieved high performance	• Used only a type of symptom for detection which may be similar and being signs for different disease
Akram et al. (2014)	Hybrid classifier • GMM • M-Mediods	Lesions • Exudate • Hemorrhage • Microaneurysm	• Used three types of signs for detection	• Used wavelet transformation for vessel segmentation which was exhausting and complex

CHAPTER 3: METHODOLOGY

3.1 Bloom's taxonomy

Bloom's taxonomy is used as a reference to structure the discussion of each machine learning model implemented as shown below:

Table 3.1: Bloom's taxonomy

Level	**Objective**	**Description**
Knowledge	Remember facts.	Each machine learning model presented was defined.
Comprehension	Demonstrate an understanding of facts.	The characteristics of each model were identified.
Application	Solve problems in new situations by applying acquired knowledge.	The machine learning models were trained to predict the presence of diabetic retinopathy.
Analysis	Break information in parts and find evidence to support generalizations.	The results of the training were analyzed.
Synthesis	Put parts together.	The results of the training were put together.
Evaluation	Make judgments about information.	The best model was determined to detect the presence of diabetic retinopathy.

3.2 Programming Language and Libraries

This machine learning project is implemented using the Python programming language. Various libraries were used with their purposes as shown below:

Table 3.2: Python libraries used in this project

Library	**Purpose**	**Description**
SciPy	Data loading	To import arff file format using the scipy.io.arff module.

Pandas	Data wrangling	To clean the data.
Matplotlib	Visualization	To visualize cross-validation performance etc.
Plotly	Visualization	To visualize features generated from Principal Component Analysis (PCA).
Scikit-learn	Machine learning	To do data partitioning, model training, grid search cross-validation etc.

3.3 Data Acquisition

The dataset used contains features extracted from the Messidor database, which comprises 1200 compressed eye fundus images with different resolutions. The dataset is retrieved from UCI Machine Learning Repository. The dataset contains 1151 observations, 19 features and a target class.

3.4 Features

The features of the dataset are presented as follows:

Table 3.3: Features of the dataset

Features	**Description**
quality	The binary result of quality assessment, where 0 represents bad quality and 1 represents sufficient quality.
prescreen	The binary result of pre-screening, where 1 indicates severe retinal abnormality and 0 indicates its lack.
ma_detection_0.5 - ma_detection 1.0	The number of microaneurysms (MAs) found at the confidence level alpha = 0.5, 0.6, 0.7, 0.8, 0.9,1.0.
exudates_0.1 - exudates_0.8	The number of exudates found at confidence level alpha = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 with normalization.
dist_macula_optic	The euclidean distance of the center of the macula and the center of the optic disc to provide important information regarding the patient's condition. This feature is also normalized with the diameter of the ROI.

diameter_optic	The diameter of the optic disc.
am_fm	The binary result of the Amplitude Modulation and Frequency-Modulation (AM/FM) based classification.
Class	Class label where 1 indicates the presence of signs of DR (Accumulative label for the Messidor classes 1, 2, 3) and 0 indicates the absence of signs of DR.

3.5 Data Visualization

Data visualization is useful to have a basic understanding of the data. It helps to determine if there is any anomaly, error, missing values or patterns in the data.

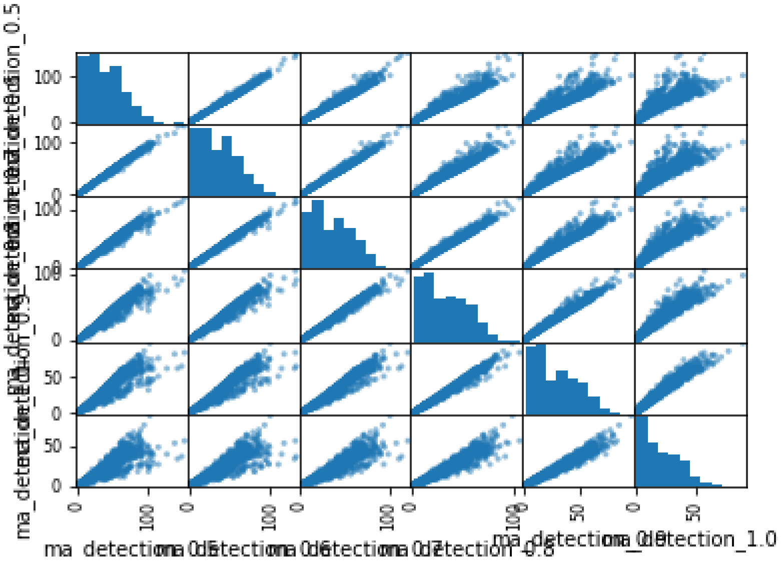


Figure 3.1: Scatter plot matrix of all *microaneurysms* features

The scatter plot matrix shown above shows the histograms and scatter plots between all the *microaneurysms* features. The plot suggests that the *microaneurysms* features are highly positively correlated to each other. Correlated features need to be treated before fitting a regression model.

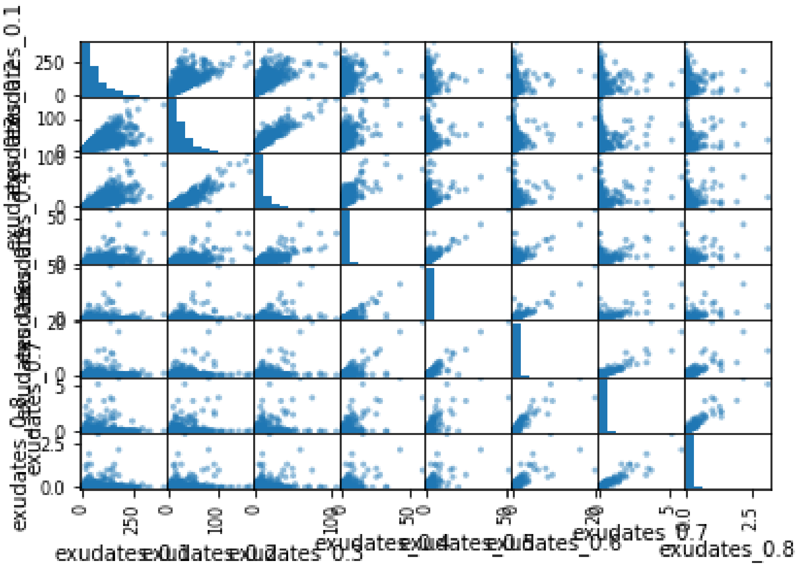


Figure 3.2: Scatter plot matrix of all *exudates* features

The scatter plot matrix shown above shows the histograms and scatter plots between all the *exudates* features. When the confidence level is above 0.4, the histograms suggest that the features are highly saturated around 0. These features may not be useful on predicting the target classes.

3.6 Pre-processing of Data

Pre-processing of data is performed after data acquisition. The data is first checked to ensure that there is no missing value. Data cleaning is performed next by changing the class type of the 'Class' variable into integer value. Data cleaning is followed by data partitioning, where the data is partitioned into features as X and target class as Y. The data is split into 60% training set and 40% test set. Stratify split is then performed to maintain the class proportion among training and testing set. The proportion of target class with value of 1 is 53.1%, which is approximately 50% after the stratify split. Hence, the target class is balance and active measures such as oversampling is not required to balance the target class.

3.7 Model Training

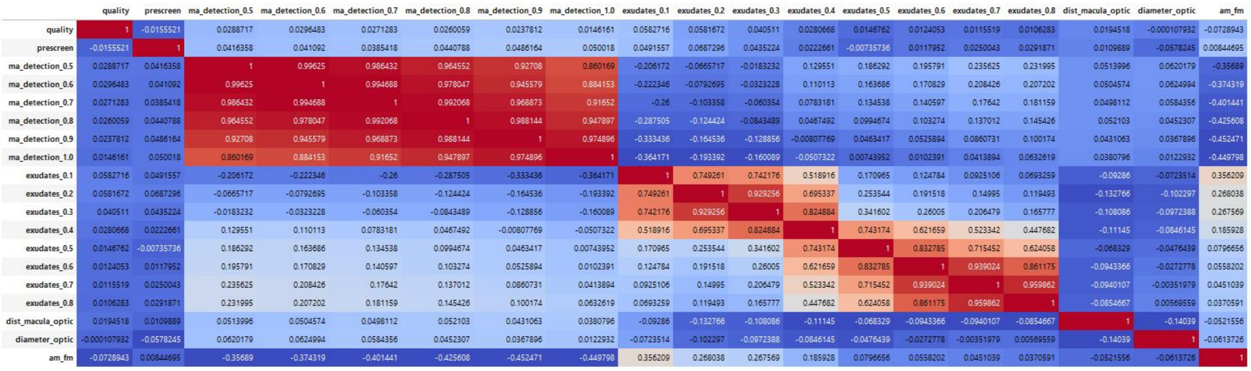
3.7.1 Model 1: Logistic Regression

Logistic regression (LR) is a statistical regression model used to model the probability of a discrete set of classes. In LR, the logs of odds of dependent variable is modelled as a linear combination of the independent variables. The LR can be modelled as

$$log\frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \varepsilon$$

where p is the predicted probability of the outcome, β_0 is the intercept, β_1 is the slope of the variable and Ɛ is the random error component.

In this project, LR is used to predict the binary outcome, which are cases with DR (coded as 1) and without DR (coded as 0). Since LR assumption includes absence of multicollinearity, the correlation between features is first examined to remove features that are highly correlated to each other. A heatmap with correlation matrix between features is created to ease the assessment of multicollinearity.



	quality	prescreen	ma_detection_0.5	ma_detection_0.6	ma_detection_0.7	ma_detection_0.8	ma_detection_0.9	ma_detection_1.0	exudates_0.1	exudates_0.2	exudates_0.3	exudates_0.4	exudates_0.5	exudates_0.6	exudates_0.7	exudates_0.8	dist_macula_optic	diameter_optic	am_fm
quality	1	-0.0155521	0.0288717	0.0296483	0.0271283	0.0260059	0.0237812	0.0146161	0.0582716	0.0581672	0.040511	0.0280668	0.0146762	0.0124053	0.0115519	0.0106283	0.0194518	-0.000107932	-0.0728943
prescreen	-0.0155521	1	0.0416358	0.041092	0.0385418	0.0440788	0.0486164	0.050018	0.0491557	0.0687296	0.0435224	0.0222661	-0.00735736	0.0117952	0.0250043	0.0291871	0.0109889	-0.0578245	0.00844695
ma_detection_0.5	0.0288717	0.0416358	1	0.99625	0.986432	0.964552	0.92708	0.860169	-0.206172	-0.0665717	-0.0183232	0.129551	0.186292	0.195791	0.235625	0.231995	0.0513996	0.0620179	-0.35689
ma_detection_0.6	0.0296483	0.041092	0.99625	1	0.994688	0.978047	0.945579	0.884153	-0.222346	-0.0792695	-0.0323228	0.110113	0.163686	0.170829	0.208426	0.207202	0.0504574	0.0624994	-0.374319
ma_detection_0.7	0.0271283	0.0385418	0.986432	0.994688	1	0.992068	0.968873	0.91652	-0.26	-0.103358	-0.060354	0.0783181	0.134538	0.140597	0.17642	0.181159	0.0498112	0.0584356	-0.401441
ma_detection_0.8	0.0260059	0.0440788	0.964552	0.978047	0.992068	1	0.988144	0.947897	-0.287505	-0.124424	-0.0843489	0.0467492	0.0994674	0.103274	0.137012	0.145426	0.052103	0.0452307	-0.425608
ma_detection_0.9	0.0237812	0.0486164	0.92708	0.945579	0.968873	0.988144	1	0.974896	-0.333436	-0.164536	-0.128856	-0.00807769	0.0463417	0.0525894	0.0860731	0.100174	0.0431063	0.0367896	-0.452471
ma_detection_1.0	0.0146161	0.050018	0.860169	0.884153	0.91652	0.947897	0.974896	1	-0.364171	-0.193392	-0.160089	-0.0507322	0.00743952	0.0102391	0.0413894	0.0632619	0.0380796	0.0122932	-0.449798
exudates_0.1	0.0582716	0.0491557	-0.206172	-0.222346	-0.26	-0.287505	-0.333436	-0.364171	1	0.749261	0.742176	0.518916	0.170965	0.124784	0.0925106	0.0693259	-0.09286	-0.0723514	0.356209
exudates_0.2	0.0581672	0.0687296	-0.0665717	-0.0792695	-0.103358	-0.124424	-0.164536	-0.193392	0.749261	1	0.929256	0.695337	0.253544	0.191518	0.14995	0.119493	-0.132766	-0.102297	0.268038
exudates_0.3	0.040511	0.0435224	-0.0183232	-0.0323228	-0.060354	-0.0843489	-0.128856	-0.160089	0.742176	0.929256	1	0.824884	0.341602	0.26005	0.206479	0.165777	-0.108086	-0.0972388	0.267569
exudates_0.4	0.0280668	0.0222661	0.129551	0.110113	0.0783181	0.0467492	-0.00807769	-0.0507322	0.518916	0.695337	0.824884	1	0.743174	0.621659	0.523342	0.447682	-0.11145	-0.0846145	0.185928
exudates_0.5	0.0146762	-0.00735736	0.186292	0.163686	0.134538	0.0994674	0.0463417	0.00743952	0.170965	0.253544	0.341602	0.743174	1	0.832785	0.715452	0.624058	-0.068329	-0.0476439	0.0796656
exudates_0.6	0.0124053	0.0117952	0.195791	0.170829	0.140597	0.103274	0.0525894	0.0102391	0.124784	0.191518	0.26005	0.621659	0.832785	1	0.939024	0.861175	-0.0943366	-0.0272778	0.0558202
exudates_0.7	0.0115519	0.0250043	0.235625	0.208426	0.17642	0.137012	0.0860731	0.0413894	0.0925106	0.14995	0.206479	0.523342	0.715452	0.939024	1	0.959862	-0.0940107	-0.00351979	0.0451039
exudates_0.8	0.0106283	0.0291871	0.231995	0.207202	0.181159	0.145426	0.100174	0.0632619	0.0693259	0.119493	0.165777	0.447682	0.624058	0.861175	0.959862	1	-0.0854667	0.00569559	0.0370591
dist_macula_optic	0.0194518	0.0109889	0.0513996	0.0504574	0.0498112	0.052103	0.0431063	0.0380796	-0.09286	-0.132766	-0.108086	-0.11145	-0.068329	-0.0943366	-0.0940107	-0.0854667	1	-0.14039	-0.0521556
diameter_optic	-0.000107932	-0.0578245	0.0620179	0.0624994	0.0584356	0.0452307	0.0367896	0.0122932	-0.0723514	-0.102297	-0.0972388	-0.0846145	-0.0476439	-0.0272778	-0.00351979	0.00569559	-0.14039	1	-0.0613726
am_fm	-0.0728943	0.00844695	-0.35689	-0.374319	-0.401441	-0.425608	-0.452471	-0.449798	0.356209	0.268038	0.267569	0.185928	0.0796656	0.0558202	0.0451039	0.0370591	-0.0521556	-0.0613726	1

Figure 3.3: Heatmap with Correlation Matrix between Features

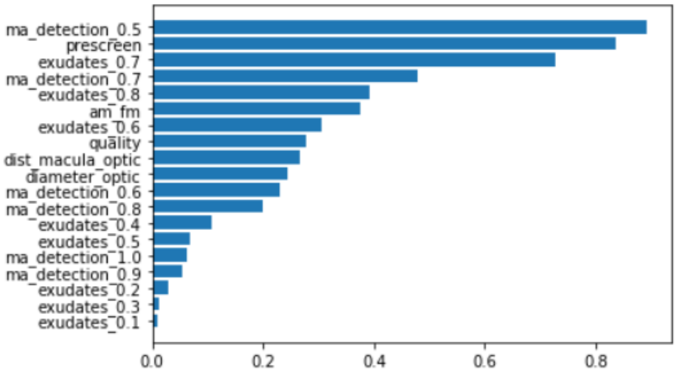


Figure 3.4: Result of Features Importance

From the features importance plot, *ma_detection_0.5* and *exudates_0.7* are the best feature among the microaneurysms and exudates features respectively. In order to avoid multicollinearity, *ma_detection* is selected to represent microaneurysms features and *exudates_0.7* is selected to represent the exudates features. All other variables seem to be significant for the prediction and are preserved as the final list of features. Seven features are selected, namely *quality*, *prescreen*, *ma_detection_0.5*, *exudates_0.7*, *dist_macula_optic*, *diameter_optic* and *am_fm*.

The selected features are then fitted into the LR and the fitted model can be written as

$$\begin{aligned} log\,\frac{p}{1-p} = {} & -0.41547947 + \ 0.33007597x_1 - 0.72855563x_2 + 0.02204903x_3 \\ & + 2.67820208x_4 - 0.12654618x_5 - 0.14282742x_6 + 0.170742687x_7 \end{aligned}$$

where x_1, x_2, x_3, x_4, x_5, x_6, x_7 represent *quality*, *prescreen*, *ma_detection_0.5*, *exudates_0.7*, *dist_macula_optic*, *diameter_optic* and *am_fm* features respectively and the slope of each feature is the regression coefficient. The regression coefficient represents an average increase of log odds per unit increase of features. A positive regression coefficient indicates that the mean of the dependent variable increases with the value of independent variable, hence increases the likelihood of the DR, whereas a negative regression coefficient indicates that the mean of the dependent variable decreases with the value of independent variable, hence reduces the likelihood of DR. The higher the regression coefficient, the higher the influence of the feature on the log odds. As seen from the fitted model equation, *exudates_0.7* with the regression coefficient value of 2.67820208 has the highest influence on log odds whereas *ma_detection* with the regression coefficient value of 0.02204903 has the least influence on log odds. *Quality*, *ma_detection_0.5*, *exudates_0.7* and *am_fm* have positive correlation with the log odds whereas *prescreen*, *dist_macula_optic* and *diameter_optic* have negative correlation with the log odds.

3.7.2 Model 2: Support Vector Machine with Dimensionality Reduction using Principal Component Analysis

In this section, dimensionality reduction is first performed on the training features using Principal Component Analysis (PCA), and a Support Vector Machine (SVM) model is fitted on the newly generated set of training features. This SVM model can then be evaluated on the testing set.

3.7.2.1 Dimensionality Reduction using Principal Component Analysis (PCA)

PCA is used to transform a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. Based on the correlation heatmap shown earlier, the *exudates* and *ma_detection* features are correlated to each other respectively. Therefore, it is useful to perform PCA to:

1. Reduce the dimensionality of the features.
2. Generate a set of features that are uncorrelated to each other.
3. Visualize the features and the target class on a 2-dimensional (2D) scatter plot.

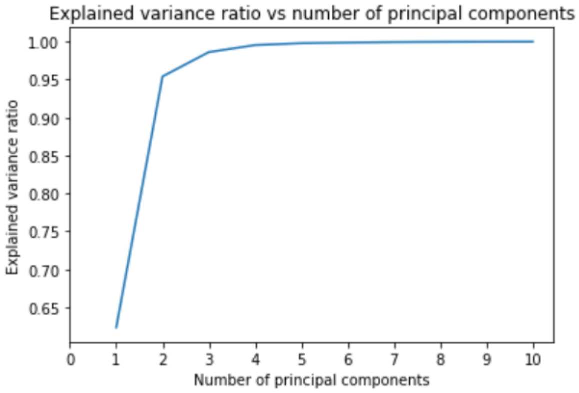


Figure 3.5: Explained variance ratio vs number of principal components

First, PCA is performed on the training set. Then, the explained variance ratio is examined as shown in Figure 3.5. The higher the number of principal components included, the higher the explained variance ratio. However, it is optimal to select a low number of principal components to reduce the dimensionality of our newly generated features, at the same time, these selected principal components can explain as much variation of the original features. This can be achieved by using the elbow method, where the marginal gain of explained variance ratio is highest when two principal components are selected. The explained variance ratio is 95.4% when two principal components are selected.

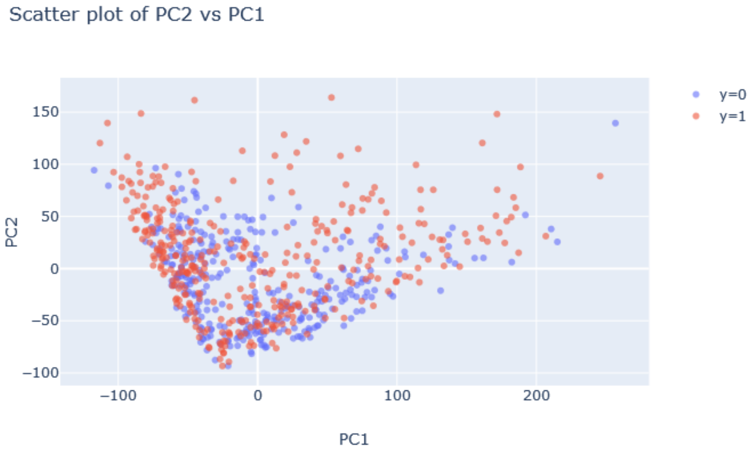


Figure 3.6: Scatter plot of second principal component against the first principal component

With the selection of the top two principal components as our new features, a scatter plot can be plotted to examine the relationship between these two features and the target class as shown in Figure 3.6. This scatter plot suggested that it is not possible to draw a linear hyperplane or line that could separate the two target classes (0 and 1), hence the data is not linearly separable.

3.7.2.2 Modelling using Support Vector Machine (SVM)

SVM is a type of supervised learning algorithm used for classification problems. SVM constructs a set of hyperplanes to separate samples belonging to different target classes. By default, SVM works best on linearly separable classification tasks. With the use of kernel trick, SVM can also perform non-linear classification by mapping the inputs into high-dimensional feature spaces.

Since, the data is not linearly separable, therefore it is suggested to use a non-linear kernel for SVM. An SVM with Radial Basis Function (RBF) kernel is fitted on the new training features (2 principal components). To evaluate the performance of the SVM on the testing set, the fitted PCA is used to transform the testing features. Similarly, the top 2 principal components of the transformed testing set are used as the new testing features. Then, the fitted SVM took the new testing features as the input and predict the target class for each observation in the testing set.

3.7.3 Model 3: Random Forest

Random Forests (RF) is an ensemble machine learning method for classification and regression prediction. RF algorithm incorporates numbers of decision trees, trains each on a different set of observations, separating nodes in each tree by taking into consideration of selected number of features. This algorithm works by building a multitude of decision trees of individual trees. The advantages of using RF algorithm include its ability to handle many features and relatively good prediction performance as compared to decision tree prediction algorithm.

In this project, RF prediction algorithm is used as one of the prediction models for DR disease using features data extracted from the Messidor database. In tuning the RF prediction model, 5-folds grid search cross-validation (GridSearchCV) has been applied to identify the best hyperparameter (*n_estimators* evaluated from n = 5 to 100) for RF Model.

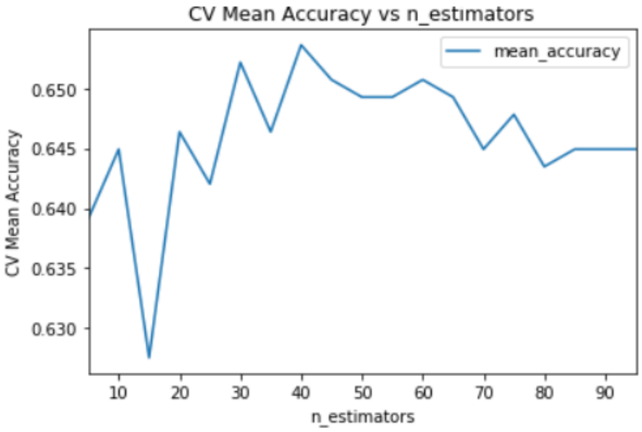


Figure 3.7: Cross-validation mean accuracy against the hyperparameter, n_estimators

Based on the Figure 3.7, the most satisfactory CV mean accuracy is yielded by using hyperparameter of 40. Hence, the hyperparameter that will be used for the RF model is 40.

3.7.4 Model 4: k-Nearest Neighbors Classifier

k-Nearest-Neighbors (k-NN) classifier is a supervised machine learning model. A supervised machine learning model learns from a set of data which is labeled and takes in a group of input variables to produce a specific output variable. The model usually trains with the input data and learns how to relate the input to the desired output to make predictions on a set of unseen data. A k-NN classifier model works by taking a data point and compares it with a number of ‘k’ closest labeled data points. The distance metric is used to determine the ‘k’ number of nearest points from the data point. In this project, Euclidean distance is calculated using the formula, $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$, which is the distance between the data point and its ‘k’ nearest points. Then, the data point is grouped according to the label of the majority of the group of ‘k’ closest points (See Figure 3.8).

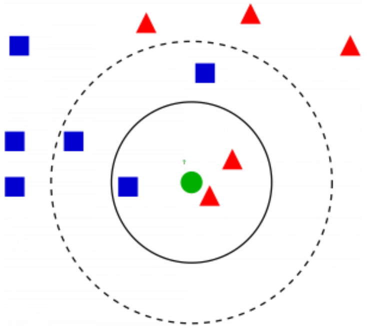


Figure 3.8: Example of k-NN Classifier

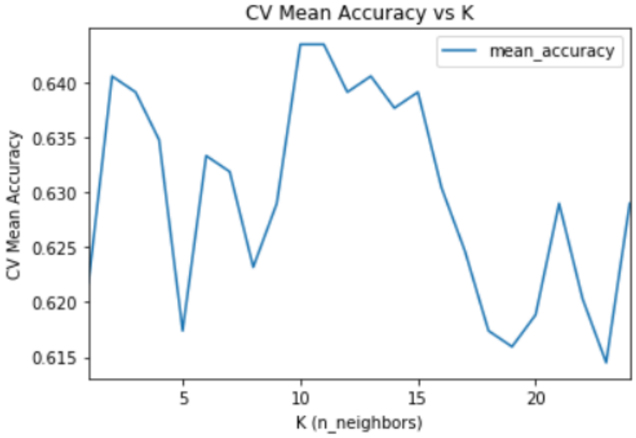


Figure 3.9: Cross-validation mean accuracy against the hyperparameter, ‘k’ value

A hyper tuning of the model’s parameter is performed by using GridSearchCV to obtain an optimum value of ‘k’. Based on Figure 3.9, the best CV Mean Accuracy is obtained by using k = 10. Thus, the ‘k’ value of the k-NN classifier model is set to 10.

3.7.5 Model 5: XGBoost

XGBoost is a powerful tree boosting system used by many data scientists to achieve state-of-the-art results on various machine learning challenges. In 2015, 17 solutions used XGBoost out of 29 Kaggle challenge winning solutions published at Kaggle’s blog (Chen and Guestrin, 2016). To improve the performance of DR detection further, XGBoost is selected for model training. A simple 5-folds grid search cross validation on XGBoost is implemented to search for the best *n_estimators* hyperparameter on the training set. The search space for *n_estimators* was 5 to 295 inclusive by a step size of 5.

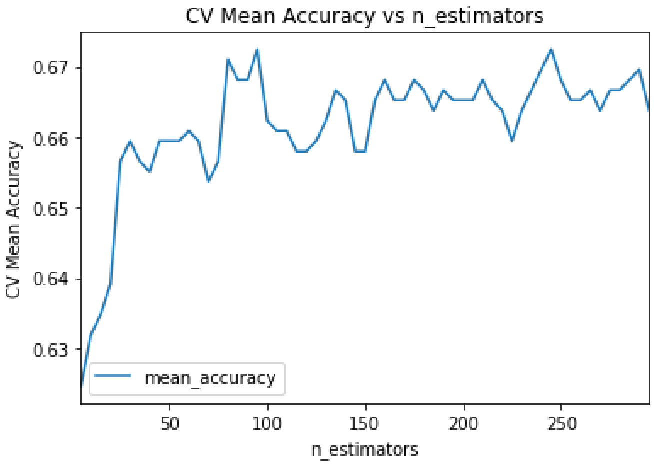


Figure 3.10 Cross-validation mean accuracy against the hyperparameter, *n_estimators*.

A total of 59 models were trained, and the best *n_estimators* hyperparameter was found to be 95 as shown in Figure 3.10. The best XGBoost model trained with the hyperparameter *n_estimators* of 95 is evaluated on the training and testing set.

3.8 Source Code

The source code for the implementation of the machine learning models is available at:

https://github.com/junyitt/machine_learning_project

CHAPTER 4: RESULTS

4.1 Performance of Logistic Regression

The performance of the logistic regression model is shown below:

Table 4.1: Model Evaluation of Logistic Regression

Sample	Precision (%)	Recall (%)	F1 score (%)	Accuracy (%)
Training	64.03	64.21	64.12	61.88
Testing	65.65	70.20	67.85	64.64

The model is able to achieve an accuracy of 64.64% with the use of seven features.

4.2 Performance of Support Vector Machine

The performance of the SVM model is shown below:

Table 4.2: Model Evaluation of Support Vector Machine

Sample	Precision (%)	Recall (%)	F1 score (%)	Accuracy (%)
Training	69.96	48.36	57.19	61.59
Testing	76.33	52.65	62.31	66.16

With the use of only 2 PCA-generated features, the model is able to predict the target class (DR) on samples not seen by the model with an accuracy of 66.16%.

4.3 Performance of Random Forest

The performance of the RF model is shown below:

Table 4.3: Model Evaluation of Random Forest

Sample	Precision (%)	Recall (%)	F1 score (%)	Accuracy (%)
Training	84.95	74.04	79.12	79.28
Testing	67.90	67.35	67.62	65.73

Based on the performance result above, it showed that RF performed relatively satisfactory results. However, in terms of accuracy, RF scores as the second lowest accuracy prediction model.

4.4 Performance of k-Nearest Neighbors

Table 4.4: Model Evaluation of k-Nearest Neighbors

Sample	Precision (%)	Recall (%)	F1 score (%)	Accuracy (%)
Training	78.55	59.02	67.39	69.71
Testing	75.84	55.10	63.83	66.81

Based on Table 4.4, it is suggested that a simple, non-parametric model like k-NN can produce a fairly good performance in terms of precision and accuracy.

4.5 Performance of XGBoost

Table 4.5: Model Evaluation of XGBoost

Sample	Precision (%)	Recall (%)	F1 score (%)	Accuracy (%)
Training	96.63	86.07	91.04	91.01
Testing	71.06	68.16	69.58	68.33

The performance of XGBoost on the training set is significantly higher than on the testing set, suggesting that the model overfit on the training sample. However, the model can capture some non-trivial patterns in the training set and generalize well into the testing set, where the performance on the testing set is comparatively higher than the other models implemented.

4.6 Comparison of Performance on Testing Set

Table 4.6: Overall performance of the machine learning models implemented

Machine Learning Model	**Precision (%)**	**Recall (%)**	**F1 score (%)**	**Accuracy (%)**
Logistic Regression (LR)	65.65	70.20	67.85	64.64
Support Vector Machine (SVM)	76.33	52.65	62.31	66.16
Random Forest (RF)	67.90	67.35	67.62	65.73
k-Nearest Neighbors (KNN)	75.84	55.10	63.83	66.81
XGBoost	71.06	68.16	69.58	68.33

In this project, the data is partitioned into a separate 40% testing set, which is never seen by the machine learning models implemented earlier. This is useful to evaluate the model's performance realistically and to avoid exaggerating the performance metrics due to overfitting on the training samples.

In terms of accuracy, XGBoost achieved the best performance followed by KNN as shown in the table above. This suggests that XGBoost can capture complex relationships between the features and achieve a higher performance through boosting. The relatively good performance of KNN suggests that samples with similar features tend to imply the same target class.

In terms of precision, SVM achieved the best performance. This suggests that the use of uncorrelated features generated from PCA is useful to make sure that the true positive rate of DR is high. Nevertheless, KNN and XGBoost also achieved relatively good precision.

In terms of recall, LR achieved the best performance, followed by XGBoost. This suggests that the two models capture sufficiently large proportion of samples with the presence of DR. SVM has the worst recall, which suggests that the model captured only a small proportion of samples with the presence of DR.

F1 score, which is the harmonic mean of precision and recall, is a more robust metric to measure the performance of a model for classification task. Among the 5 models implemented, XGBoost and LR achieved a high F1 score of 69.58% and 67.85% respectively. This suggests that the two models are robust in terms of both precision and recall.

Overall, in terms of performance, XGBoost is the best machine learning model. However, XGBoost is difficult to interpret. On the other hand, LR is robust in terms of F1 score, and the regression coefficients can be interpreted. This is useful to both researchers and decision makers to understand the impact of various features extracted from the image scan of patients on the presence of DR.

CHAPTER 5: CONCLUSION

In this project, an automated DR system is developed using several machine learning algorithms, namely logistic regression, support vector machine, random forest, k-Nearest Neighbors and XGBoost based on the features extracted from the Messidor database. The results showed that the model generated are robust. Among all the models generated, XGBoost achieved the highest overall performance. However, there existed a drawback using XGBoost as it is difficult to interpret. Conversely, LR produced a high F1 score and the regression coefficients can be easily interpreted. Interpretability is important for both the decision makers and researchers to understand the influence of features on the outcome of the prediction, that is the presence of DR.

Future improvement of the DR system can include the grading of DR severity. The severity of DR can be classified into mild, moderate, severe non-proliferative and proliferative. Besides that, the detection system can be expanded in order to recognize different signs of diabetic retinopathy such as abnormal growth of blood vessels and cotton wool spots as different symptoms appear in each stage of DR. Different machine learning algorithms can also be used to further enhance the results of DR detection.

REFERENCES

Akram, M. U., Khalid, S., Tariq, A., Khan, S. A., & Azam, F. (2014). Detection and classification of retinal lesions for grading of diabetic retinopathy. *Computers in biology and medicine*, *45*, 161-171.

Amin, J., Sharif, M., Yasmin, M., Ali, H., & Fernandes, S. L. (2017). A method for the detection and classification of diabetic retinopathy using structural predictors of bright lesions. *Journal of Computational Science*, *19*, 153-164.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Sidibé, D., Sadek, I., & Mériaudeau, F. (2015). Discrimination of retinal images containing bright lesions using sparse coded features and SVM. *Computers in biology and medicine*, *62*, 175-184.

Theera-Umpon, N., Poonkasem, I., Auephanwiriyakul, S., & Patikulsila, D. (2019). Hard exudate detection in retinal fundus images using supervised learning. *Neural Computing and Applications*, 1-18.