PYTHON MACHINE LEARNING

CONTINUOUS ASSESSMENT

SA50 Team 9

Matriculation No.	**Name**
A0160548A	Foo Rui Hao James
A0087016H	Leng Chung Hiang
A0214876W	Ma LuLu
A0214896R	Yadanar Phyo
A0214875X	Chong Heng Tat
A0214907E	Guo JieYi
A0214877U	Ngui Kai Lin
A0214888N	Thun Su Nyi Nyi

Contents

1. Problem Statement / Introduction

Living in the times of a pandemic has once again thrown the focus on the importance of health to an individual as well as healthcare to a society. Naturally, the group has decided to centre our current learning on health-related topics given their universal appeal at the moment.

For the first part of our discussion, we will be looking at a dataset (Insurance.csv) consisting mainly of demographic information and the medical expenses incurred in a calendar year for multiple individuals. This dataset is of interest as it provides insights to the expected healthcare cost of a particular demographic, which can be used to inform insurance companies on the level of premiums to charge or for the case of a nationalised insurance entity, how much government budget will be required to keep the insurance programme afloat. Linear regression technique will be used to predict medical expenses given the demographic information.

The latter part of the discussion will be focusing on a chronic disease, Diabetes. The dataset (Diabetes.csv) consists mainly of bio-physical measurements and the diabetic statuses of a group of native American women. The objective would be to predict the diabetic status of a patient using the given bio-physical measurements and also to identify which measurements are of the greatest importance in the prediction. Unsupervised and supervised techniques will be employed here.

2. Datasets and Data Dictionary

2.1. Insurance

The insurance dataset was retrieved from Kaggle[1]. The original source of the dataset comes from the textbook "Machine Learning with R" by Brett Lantz. According to the text, the dataset was simulated using the demographic statistics from the United States Census Bureau, so it is representative of real-world conditions. A total of 1338 records are present in the dataset. The six features identified are age, sex, BMI, children, smoker and region, with the charges as the target variable. The group will attempt to build a linear regression model to study how the six features influence the target variable in Section 3.

Variable Name	**Definition**
Age	Age of the insurance plan's primary beneficiary
Sex	Insurance policy holder's gender, either male or female
BMI	Weight in kg divided by square of height in m
Children	Number of children/dependents covered by the insurance plan
Smoker	Either yes or no depending on whether the insured regularly consumes tobacco product
Region	Place of residence in the US - either northeast, southeast, northwest or southwest
Charges	Total expenses charged to the insurance plan for the calendar year

[1] https://www.kaggle.com/mirichoi0218/insurance

2.2. Diabetes

The diabetes dataset was retrieved from Kaggle[2]. The health information of the dataset originated from a study conducted by the United States' National Institute on Diabetes and Digestive and Kidney Diseases[3] on the Pima Indian population residing in Phoenix, Arizona. In total, the dataset contains records of 768 female subjects. There are eight independent variables identified, with a single dependent variable – the outcome of diabetes diagnosis. This dataset will be used in Section 4 for unsupervised learning (DBSCAN and K-Means) as well as supervised learning techniques (Logistic Regression, K-NN, Decision Trees and Neural Network).

Independent Variables

Variable Name	**Definition**
Pregnancies	Number of times the individual has been pregnant
Glucose	Plasma Glucose Concentration at 2 Hours in an oral glucose tolerance test
Blood Pressure	Diastolic blood pressure measured in mmHg
Skin Thickness	Triceps skin fold thickness measured in mm
Insulin	2-Hour serum insulin measured in μU/mL
BMI	Weight in kg divided by square of height in m
Diabetes Pedigree Function	A function which measures genetic influence of relatives on the subject's diabetes risk
Age	In years

[2] https://www.kaggle.com/uciml/pima-indians-diabetes-database
[3] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2245318/pdf/procascamc00018-0276.pdf

Dependent Variable: Outcome

A positive diabetes diagnosis is defined for 2-hour plasma glucose concentration of 200 mg/dL and above during an oral glucose tolerance test. A positive result is indicated as outcome 1 while a negative result is identified as outcome 0. There are 500 instances of negative diagnosis, and 268 instances of positive diagnosis in the dataset, giving a majority class to minority class ratio of approximately 65:35.

3. Discussion on Insurance Dataset

3.1. Data Engineering

Data Encoding

The dataset includes three categorical variables – sex, smoker and region – which store strings as their values. They will be encoded into numerical categories as follows:

- **Sex** – Male = 0, Female = 1
- **Smoker** – Non-smoker = 0, Smoker = 1
- **Region** – Northeast=0, Northwest=1, Southeast=2, Southwest=3

Data Cleaning

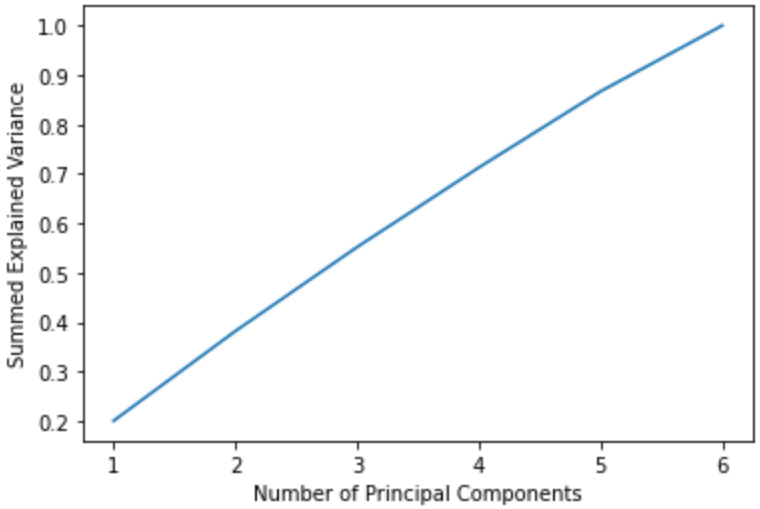
The dataset has no null values and unreasonable data. No data cleaning is required.

3.2. Feature Engineering

3.2.1. Feature Selection

Through the generated correlation matrix, one feature (smoker) is highly correlated to the dependent variable (charges). This is unsurprising, given that smoking is known to be detrimental to an individual's and their immediate family's health and in turn will increase health expenses as represented by 'charges'. Amongst the independent variables, the level of correlation observed between them was weak and hence the linear regression model will only be trained using all features selected.

3.2.2. PCA



Principal Component Analysis (PCA) is performed to reduce the dimension of the independent variable input. Looking at the graph, the summed explained variance appears rather linear to the number of principal components. To ensure that not too much information is lost, five principal components – corresponding to a summed explained variance of 0.87 – will be used in subsequent regression modelling. This translates to a dimension reduction of just one.

3.3. Supervised Learning – Linear Regression

a. Model Training with all 6 features

Duration - 0.00737762451171875 seconds

The r2 score of training model without feature engineering using all six features is 0.796, the intercept is 13273.16 and the coefficient for each features (independent variables) are 3.51872708e+03, (-1.82536077e+00), 1.96825828e+03, 5.42446655e+02, 9.52447231e+03 and (-3.76894428e+02).

b. Model Training with Principal Components

Duration - 0.0019960403442382812 seconds

The r2 score of the model trained using 5 principal components remains at a similar 0.796. The intercept is 13269.90 and the coefficient for 5 principal components are (-2859.73092866), 5223.90087473, 3034.74053574, (-3643.32338534) and 7037.30980773.

From comparing the r2 score, the performance of the model involving 5 principal components was almost as good as when all features were used in the regression model. The training duration taken was also less for the former. To sum up, the simplified model with principal components with 1 dimension reduced is the preferred model.

4. Discussion on Diabetes Dataset

4.1. Data Engineering

Data Encoding

The only categorical variable – outcome – has already been mapped to numerical values 0 and 1 as seen from the source dataset. No further data encoding is required.

Data Cleaning

While there were no missing values across the 768 rows in the dataset, the group noted that the values of various dependent variables which are physical measurements have been recorded as zero. The said variables are glucose, blood pressure, skin thickness, insulin and BMI. As such, while there were no missing values across the 768 rows in the dataset, the group noted that the values are not possible, so this suggests that zeroes have been used as placeholders for missing data.

Variable	Number of missing entries
Glucose	5
Blood Pressure	35
Skin Thickness	227
Insulin	374
BMI	11

In total, there were 376 distinct rows which had one or more missing values, with 199 rows missing two values, 28 rows missing three values and 7 rows missing four values. The group considered two approaches to data cleaning – 1. Dropping the rows with missing values, and 2. Substituting the missing values with either the mean or median value calculated from the remaining valid data. The second approach was not chosen as it surfaced poorer correlation between features as well as across-the-board degradation in accuracy scores during trial runs

of the supervised learning models. Owing to this, the approach the group decided to take is to drop these 376 records from the remaining discussion.

Consequently, the cleaned diabetes dataset will consist of 392 rows, with the majority class to minority class ratio roughly unchanged at approximately 67:33.

Data Balancing

The dataset has a higher representation of subjects testing negative for diabetes compared to subjects with positive diagnosis. As mentioned, the majority class to minority class ratio for the cleaned dataset is 67:33.

The group is of the view that the current distribution of the binary classes do not represent a severe skew towards either of the classes, as severely skewed distribution is usually characterised when the minority class makes up 10% or lower of the sample.

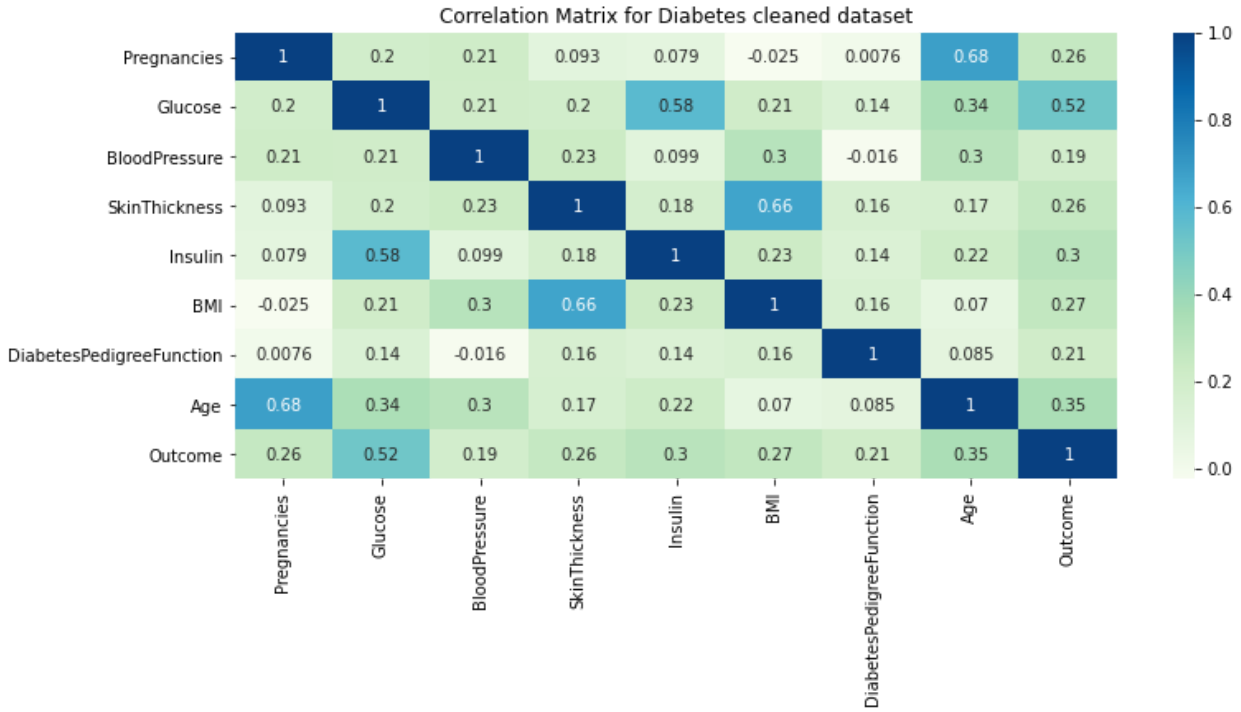
In their separate blog posts[4,5], both Harrell and Matloff posited that artificially balancing data might undermine the accuracy and utility of the predictions provided by supervised learning models when the prevalence of outcomes in reality deviates from the artificial prevalence enforced by data balancing.

With this in mind, given that the diabetes dataset is not afflicted with severe class imbalance, no data balancing will be performed on the dataset.

[4] https://www.fharrell.com/post/classification/
[5] https://matloff.wordpress.com/2015/09/29/unbalanced-data-is-a-problem-no-balanced-data-is-worse/

4.2. Feature Engineering

4.2.1. Feature Selection



From the above correlation matrix of the Diabetes dataset, there are 2 pairs of features that are correlated with a correlation value of above 0.6.

They are:

1. Pregnancies and Age with a correlation value of 0.68
2. BMI and SkinThickness with a correlation value of 0.66

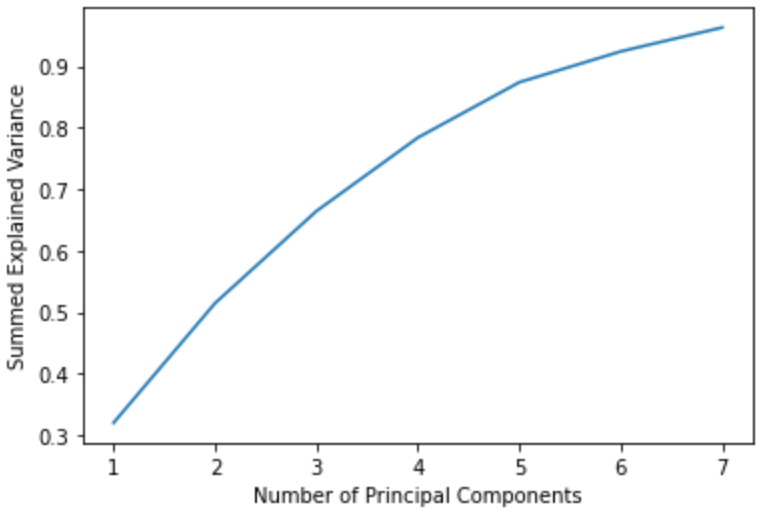
From this, our group has decided to first remove Pregnancies as a feature, since it has a lower correlation value of 0.26 against Outcome compared to Age's correlation value of 0.35.

We also decided to remove SkinThickness as a feature since it also has a slightly lower correlation value against outcome compared to BMI.

Moreover, Age and BMI are more generic features that are applicable to more people, i.e., not everyone might have been pregnant before and BMI is more easily measured compared to SkinThickness.

4.2.2. PCA

For the Principal Component Analysis, we managed to reduce the dimension of the dataset to include only 6 out of the 8 features.



As seen from the graph, the summed explained variance ratio starts to deteriorate as less features are selected. However, when either 5,6 or 7 features were selected, the tradeoff in summed explained variance ratio is relatively low.

Using PCA, we managed to obtain the following explained variance ratio for the 6 features : [0.31994031 0.19459756 0.14984544 0.11946188 0.09002175 0.05017765]

This led to a summed explained variance ratio of 0.92, which is a high explained variance ratio.

4.3. Unsupervised Learning

Assuming that the outcome for each row of features was not provided in the dataset, our group employed several unsupervised learning methods to classify the dataset based on the similarities of observations.

4.3.1. DBSCAN

Density Based Spatial Clustering attempts to classify unlabelled data into clusters based on euclidean distance and a minimum number of points within that radius

Selection of optimum epsilon Value

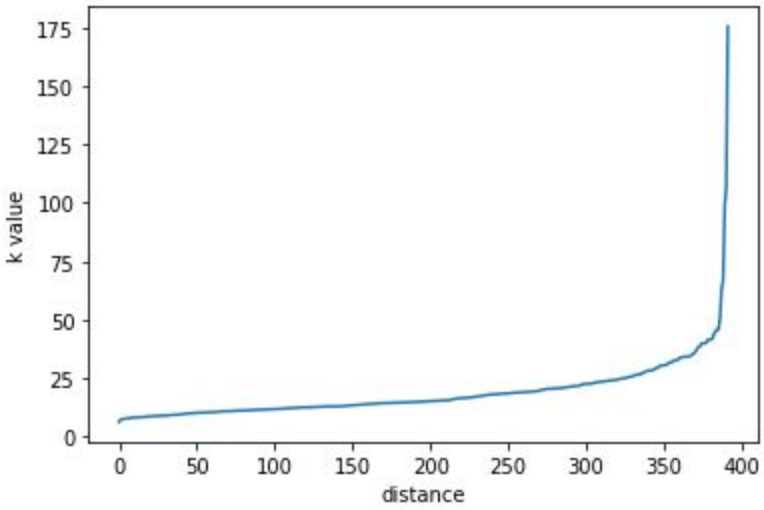
One way to determine the optimum value for epsilon is to calculate the distance of the nearest n points for each point, and then plotting the graph. The point where maximum change in curvature of the graph would be selected as the epsilon value.

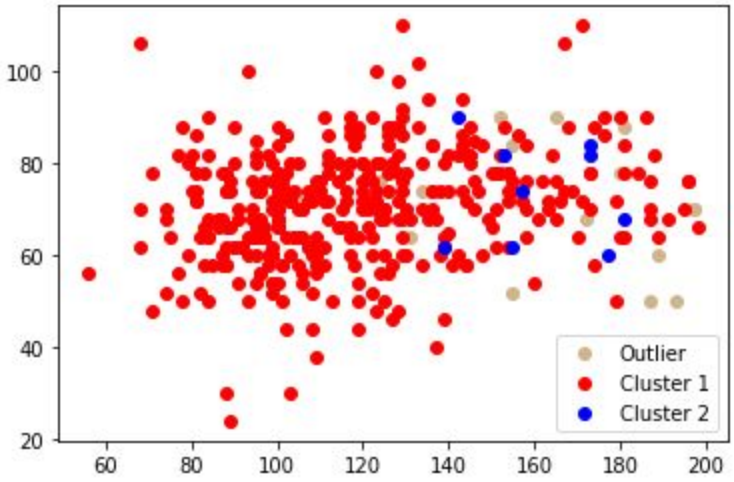
Selection of minimum points

Selection of the number of minimum points is important as it may affect the clustering algorithm. A small value of minimum points would result in more clusters from noise present in the data whereas a large value of minimum points may eliminate potentially smaller clusters. Hence a trial and error approach was used to determine the optimum number of minimum points to produce distinct clusters.

a. Model trained without PCA

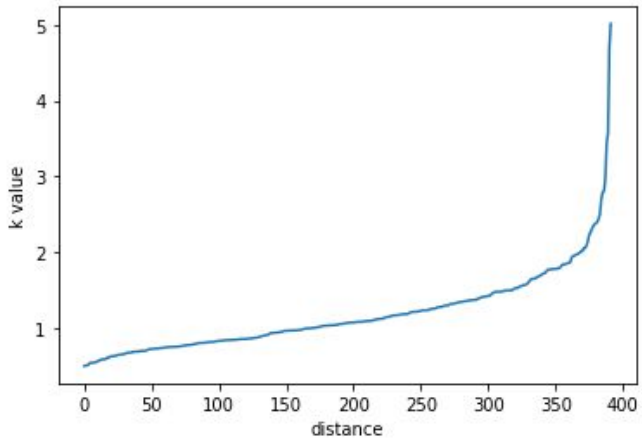
A majority of the dataset was grouped into a single cluster as seen in the plotted graph despite selecting a range of epsilon values from 25 to 50 from the K-NN graph.

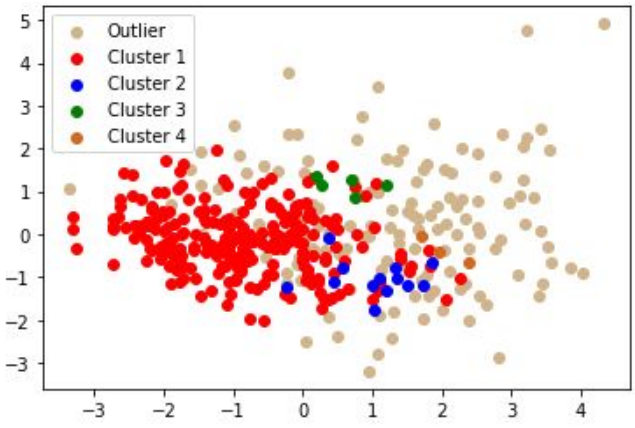




b. Model trained with PCA

From observation, it can be seen that the dataset that has undergone PCA produced a more distinct cluster. An epsilon value of 1.12 was selected based on the K-NN graph and a total of 4 clusters were formed with the majority of the datasets forming cluster 1.



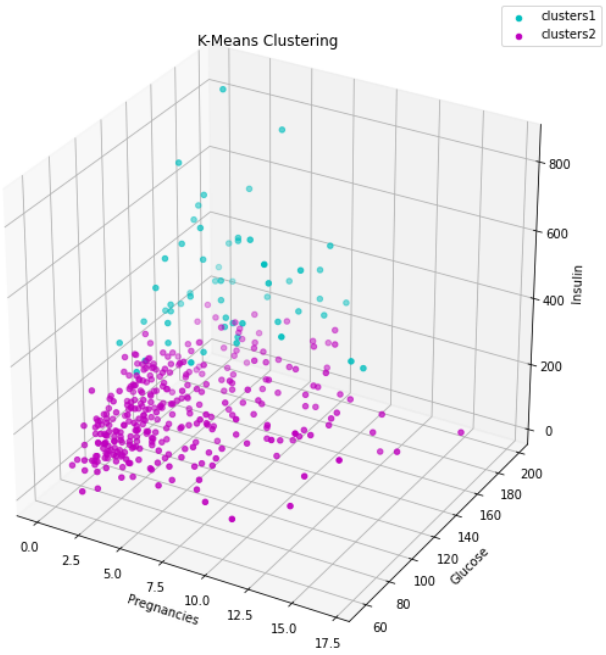


4.3.2. K-Means

K-Means algorithm is to divide the given sample into K clusters based on the distance between samples. The closer distance from the sample to its cluster while also the farther distance between each cluster, the better K-Means will be generated.

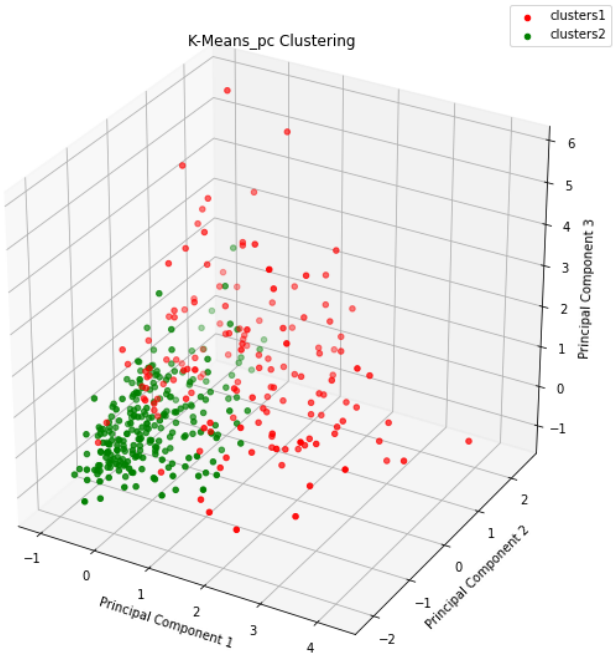
For K-Means analysis, x includes values from all features except the last column ‘Outcomes’ set as y and there will be two clusters.

a. K-Means without PCA



Samples of 2 clusters are keeping a slight distance, which can be seen from the vague gap between 2 clusters. While, the samples among the first cluster showed significant distance between each other. This could be caused by data imbalance.

b. K-Means with PCA



According to the graph, it seems clusters have not been better classified by undergoing PCA since the distance of outcomes between each cluster is even more indistinct. However, the second cluster is gathering the samples better than the results trained without PCA cause samples are closer to the center.

4.3.3. Observations

K-Means provided clearer classification on clusters compared to DBSCAN as shown by the above graphs.

4.4. Supervised Learning

4.4.1. Logistic Regression

In order to apply Logistic Regression to the diabetes detection problem, all features, selected features and principal components are extracted from the dataset_clean file.
Using the seaborn package to visualize the dataset, we can see different levels of a categorical variable by the color of plot elements.

a. Model Training with all 8 features

All 8 features are extracted from the dataset as X, and the column named “Outcome” as Y. The dataset is split to train_data and test_data with random_state 0. The number of rows of training data is 294, and testing data is 98.

x_train and y_train are fitted by the logistic regression algorithm; training time is about 0.0647s. The model is applied to x_test to predict the outcome that can be compared with the real outcome to get the model accuracy score. The accuracy score of this training model is 0.806. It is relatively high and shows this model is reasonable and reliable. The confusion matrix between y_test and y_predict is [[61,4],[15,18]]. The True-Positive is 60, the False-Negative is 5, the False-Positive is 20, and the True-Negative is 13. The error rate is 0.194, the True-Positive (Sensitivity) rate is 0.938, and the True-Negative rate (Specificity) is 0.545.

Training Time(Average)	Accuracy Score
0.065	0.806

b. Model Training with 6 features

After removing the 2 selected features through the correlation matrix, the remaining 6 features are used to train the model.

The 6 features (Glucose, BloodPressure, Insulin, BMI, DiabetesPedigreeFunction and age) are applied to the logistic regression model. The fitting-time is between 0.04 to 0.06s.

The accuracy score is also 0.806. It can be seen that these two removed features have little effect on the accuracy of this model. Hence, we observe that we can achieve reliable outcomes with lesser features, which may help reduce the processing time and operational costs.

Training Time(Average)	Accuracy Score
0.04-0.06s	<=0.806

c. Model Training with PCA

The 6 generated Principal Components were used in this model. The accuracy score is lower than the previous two models, but the training time (0.01-0.02s) is the shortest.

The accuracy score is 0.775. The confusion matrix is [[59,6],[16,17]].

This training model is acceptable and can be used for preliminary detection because of its time-saving factor.

Training Time(Average)	Accuracy Score
0.01-0.02s	0.775

4.4.2. K-NN

K-Nearest Neighbour classification technique is to find out the optimum value of K. Elbow method is used to find out the optimum number of neighbours for K-NN. First of all, all of the features from the dataset are set as X and the target 'Outcome' is set as Y. Then, the dataset is split into a training set and test set with random state 0. Next, the training set is cross-validated with the test set, where the train set is X_train and the test set is X_test. For loop is created which trains the models with different k values. In this case, the model is created with k value 1 and training time is recorded. The process repeats and the k value increases by 1 each time. The k value then increases from 1 to 14, in increments of 2, which is the difference between each number in the loop.

The best k is determined by the maximum accuracy output, which is shown after executing the code. To make the accuracy score to be printed as percentage, the maximum score is multiplied by 100, and for k to be uneven, x*2+1 function is used in the map. Lastly, the X_test is predicted and training time is printed by subtracting the stop and start time.

a. Model Training with all 8 features

It is found out that when using all 8 features to train the model, it gives the highest accuracy score of 78.57%. The average training time is 0.001372, which is also the slowest and k=9.

Training time (average)	Accuracy Score(%)
0.001372	78.57

b. Model Training with 6 features

When using 6 features, after removing Pregnancies and Skin Thickness, the accuracy score reduced from 78.57% to 77.55%. The average training time is 0.001257, and k=7,11.

Training time (average)	Accuracy Score(%)
0.001257	77.55

c. Model Training with Principal Components

In the model training with PCA, K is 11, and even though the accuracy score is the same as the score produced when the model is trained with 6 features, which is 77.55%, the training time is a lot faster, which is 0.0004594. It is almost half times faster than the average training time of 6 features. This is because PCA reduces the dimensions of the features, and gives lesser data for training the model. Therefore, this is the ideal training set for the K-NN model since it has the fastest training time, and the accuracy score does not have a big difference compared to the highest accuracy score.

Training time (average)	Accuracy Score(%)
0.0004594	77.55

4.4.3. Decision Trees

Selecting Decision Tree characteristics

a. Choosing criterion

Overall, changing the criterion of the decision tree had a great effect on the accuracy of the model. When the criterion was changed to ‘entropy’ from ‘gini’ the accuracy_score had a drastic drop from 0.77 to 0.65. This is due to the information gain of the criterion.The gini index compared to entropy had caused a larger information gain due to the split point of the model. The final criterion of the training model of the decision tree used was ‘gini’ as it proved to be the most accurate selection measure.

b. Choosing max_depth

A range of 1-30 was used for tree depth to determine the most accurate max_depth for the training model. A max depth of 3 was selected as it showed a high accuracy_score (0.7846) without being too high or too low. Having a high max_depth would lead to an increase in complexity of the model, resulting in overfitting as it will not generalise well with the test or predictive set. Furthermore, if the max_depth was too low, it would result in underfitting.

c. Choosing min_samples_leaf and min_samples_split

A range of 1-30 was also used to determine the most accurate min_samples_leaf and min_samples_split for the model. This determines the minimum number of samples for a leaf node for smoothing a model and minimum number of samples to split an internal node. A min_samples_leaf of 10 was taken as it gave the highest accuracy while min_samples_split taken was 4 as it did not have an effect on accuracy_score.

Training model Features

a. Model training with 8 features

Training time (average)	Accuracy Score
0.003281	0.7755

Using all 8 features to train the model gave the highest accuracy_score of 0.7755 and the slowest average training time of 0.003821. As the model contains the most data among all training models, it would be the most complex and time consuming due to the volume of data being processed to train the model.

b. Model training with 6 selected features

Training time (average)	Accuracy Score
0.002478	0.7755

While using 6 of the features (excluding Pregnancies and Skin Thickness), the model gave the same accuracy_score of 0.7755 as the model trained with 8 features. The average training time of 0.002478 was in between the training models using 8 features and Principal Components. The accuracy_score is the same as the training model with 8 features as the excluded features pregnancies and skin thickness had little effect on the outcome.The training time is faster due to the reduction in data volume compared to using 8 features. Overall, this is the ideal training set for the decision tree as it had the highest accuracy along with the 8 features set and had a quick training time which was close to the fastest model using principal components.

c. Model training with Principal Components

Training time (average)	Accuracy Score
0.002174	0.7551

The model using PCA had the lowest accuracy_score of 0.7551 and the fastest average training time of 0.002174. This is due to the model using a lower volume of data as the PCA had reduced the dimensions of the features using the new principal components which lead to less data processing to train the model. After testing the 3 models features, the accuracy variation among the training models is small while the training time differed largely between 8 features and the rest due to it having the highest volume of data.

Overall, the accuracy for decision tree models is in general slightly lower as compared to the other methods. However, the data input does not need to be cleaned and processed as much.

4.4.4. Neural Network

Selection of Activation Functions:
The neutral network in this discussion was built upon the Keras sequential model. As the problem here was one of binary classification between positive and negative diabetes diagnosis and subsequent prediction of the diagnosis outcome, the activation function chosen for the hidden layer(s) was sigmoid. For the output layer, the more general softmax activation function (used for multi-class classification) has been used.

Approach to Parameter Tuning:
A single hidden layer was first considered. A trial and error method was taken to vary the number of neurons, epochs and batch size to determine the optimum number for each parameter using the average accuracy score calculated by training the model 10 times with all 8 features. With the optimum number of the three parameters found, a second hidden layer was added to assess its impact on the accuracy score and whether it was needed.

- Number of Neurons, n

n	120	180	200	240	300	**360**	380
Epoch	120						
Batch Size	32						
Average Accuracy	0.7745	0.7959	0.7898	0.7980	0.7959	**0.8061**	0.8020

- Epoch

n	360						
Epoch	10	30	50	100	150	**200**	240
Batch Size	32						
Average Accuracy	0.7327	0.7367	0.7755	0.7959	0.8133	**0.8184**	0.8143

- Batch Size

n	360			
Epoch	200			
Batch Size	16	32	**64**	96
Average Accuracy	0.8122	0.8184	**0.8327**	0.7806

- Hidden Layer

A second hidden layer with n = 10 was added, while the number of neurons, epoch, batch sizes were unchanged from the optimum values (bolded in the previous tables) determined. The average accuracy obtained was 0.7871, indicating that a second hidden layer did not improve the performance of the neural network.

Parameter Tuning Results:

The following parameter values were used in the model training hereafter.

n	360
Epoch	200
Batch Size	64

a. Model Training with all 8 features

Average Accuracy	0.8327
Range of Accuracy	0.8163 - 0.8469
Average duration of training (s)	4.109

b. Model Training with 6 selected features

Average Accuracy	0.8153
Range of Accuracy	0.7551 - 0.8571
Average duration of training (s)	4.025

c. Model Training with Principal Components

Average Accuracy	0.7735
Range of Accuracy	0.7653 - 0.7857
Average duration of training (s)	3.897

As expected, the neural network model trained using all the 8 features gave the highest average accuracy score but also took the longest duration to train. The prediction performance of the

models degraded when the dimension of the input was reduced, probably because the features omitted (skin thickness and pregnancies) were only moderately correlated to the retained ones.

4.4.5. Observations/Comparison

a. Accuracy

In general, all supervised training models with different training features had a consistent and relatively high accuracy score of 0.7551-0.8327. The model with highest average accuracy_score was the Neural Network model while the model with the lowest average accuracy_score was the Decision Tree model. The highest accuracy_score (0.8327) was achieved using all 8 features in the Neural Network model and the lowest accuracy_score (0.7551) used Principal Components in the Decision Tree model. Overall, for all models, accuracy increased when using the 8 feature set due a larger training data volume for the model. However, accuracy decreased across the board when using principal components due to a smaller training data volume.

b. Average Training Time

However, the average training time differed significantly between the various models (0.0004594-4.109). The fastest model on average was the K-NN model while the slowest was the Neural Network model. The fastest average training time achieved was using Principal Components in the K-NN model and the slowest used 8 features dataset in the Neural Network model. This difference in time was due to the complexity and volume of data used to train the models. As the Neural Network model had a higher complexity and a larger volume of data, the computational requirements increased, resulting in an increase in training time.

c. Complexity

Out of thn;e four supervised learning techniques, the Neural Network model is the most complex, followed by Decision Trees and K-NN model. These three techniques can cater to non-linear solutions, unlike Logistic Regression which requires a certain level of linearity. Apart from Neural Network and Decision Trees, K-NN and Logistic Regression techniques also require some understanding of the input variables as collinearity between them can have adverse effects on the models' accuracy. With the higher level of complexity, Neural Network

usually requires a larger volume of training data. In the current context, the volume of training data provided appears to be sufficient to train a functional neural network with high accuracy. When the volume of training data is a constraint, falling back to less complex techniques should provide more meaningful outcomes.

d. Dataset

The various datasets had affected all models in terms of accuracy and training time. On average, the 8 feature dataset produced the slowest training model but had the highest accuracy while the fastest training dataset was the Principal Components dataset but had the poorest accuracy. This is due to the difference in data volume.

e. Best model

The most effective model and best dataset depends on the usage scenario.

Poor knowledge

A user with poor knowledge and who is unable to perceive a complex model might use the Decision Tree Model since it is a relatively more understandable training model.

High Accuracy

While a higher level user with a need to attain the most accurate data might use the 8 feature dataset in the Neural Network model.

Fastest Time

A user who faces time constraints might use the fastest model: Principal Components in the K-NN model.

Balanced model

Lastly, a user who requires a more balanced model with good accuracy and relatively fast training time might seek to use the Logistic Regression using the 8 feature training set.

5. Conclusion and Insights

In general, all models and methods used through this assignment had both advantages and disadvantages. Some of the methods were more specialised in terms of use. The usage of these models depended on the dataset present and usage scenario (accuracy vs training time). Each model produced varying results of average training time and accuracy.

Dataset

The target of each dataset and model heavily affects the type of model used. The outcome of the dataset determined if a supervised or unsupervised learning model was needed. The predicted outcome (Regression vs Classification) was also a contributing factor used in determining the model type.

In general, data engineering had reduced training time through a reduction of training data volume. However, this was at the expense of accuracy. We minimised this drop accuracy through different parameters in both the correlation matrix and Principal Component analysis.

Models

The models for both supervised learning and unsupervised learning heavily affected training time and accuracy due to the varying complexity in the various training models. This led to an understanding that different models had different uses. The different conditions of the usage scenarios heavily skewed which model was the best fit as these modes had different complexities, accuracies and average training times.

Supervised learning

With the different models used in supervised learning, the models produced a close but high range of accuracy but differed in training time greatly. The understanding of each training model also depended on the complexity. On average, the more complex model was, the slower the training time but the higher the accuracy.

Unsupervised learning

By comparing results from DBSCAN and K-Means, we observed that the latter produced a better fit model. DBSCAN does not work well as the dataset has multiple densities or varying densities. It also does not work well in high dimensionality of data.