The first section of this assignment covered data cleaning, so it will not be discussed here; that is visible in the jupyter file. This report will review the data exploration from the diabetes dataset of patient admissions from 1999 to 2008 in US hospitals as well as the predictive model built, clusters, and includes recommendations on the current model built.

Part 1.2:

Figure 1: Histogram of age Impact on readmissions

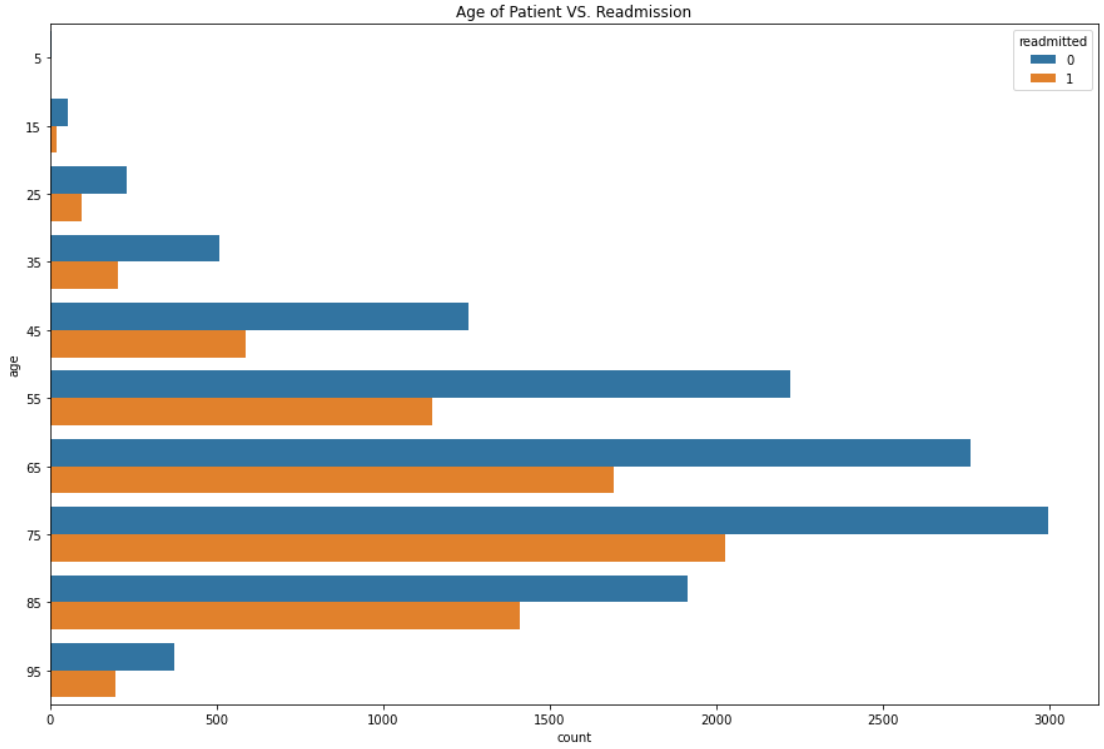


Figure 2: Percentages of all age groups and readmission

	age	Has been Readmitted (%)	Has not been Readmitted (%)
0	5	20.00	80.00
1	15	27.40	72.60
2	25	28.92	71.08
3	35	28.83	71.17
4	45	31.81	68.19
5	55	34.06	65.94
6	65	37.95	62.05
7	75	40.33	59.67
8	85	42.40	57.60
9	95	34.39	65.61

The age column of the graph has been set to be the middle value of an age group; for example, an age group of [40-50] was set to 45. In figure 1 above, the graph represents the number of times a patient has or has not been readmitted back into the hospital; the orange column (1) represents a patient who has been readmitted, while the blue column (0) represents a patient who has not been readmitted. The graph shows that as a patient's age rises, the rate of readmissions rises with it. This rise stops at 75 and declines in the last two columns; this decline could be affected by other factors like the average life expectancy in the US at the time was around 80 years old. Therefore, this graph can prove the hypothesis that age has a higher impact on readmissions until a certain age; in our graph, that age is 75 years old. A table (Figure 2) has also been made as it was noticed that more people in other age groups would have checked into a hospital for Diabetes, therefore percentages adding the total readmissions for each age group and dividing it by whether a patient has been readmitted or not has been done. This table further proves the hypothesis as age increases, the chance of readmission is higher.

The following hypothesis explores the impact race has on readmissions.

Figure 3: Histogram of race impact on readmissions

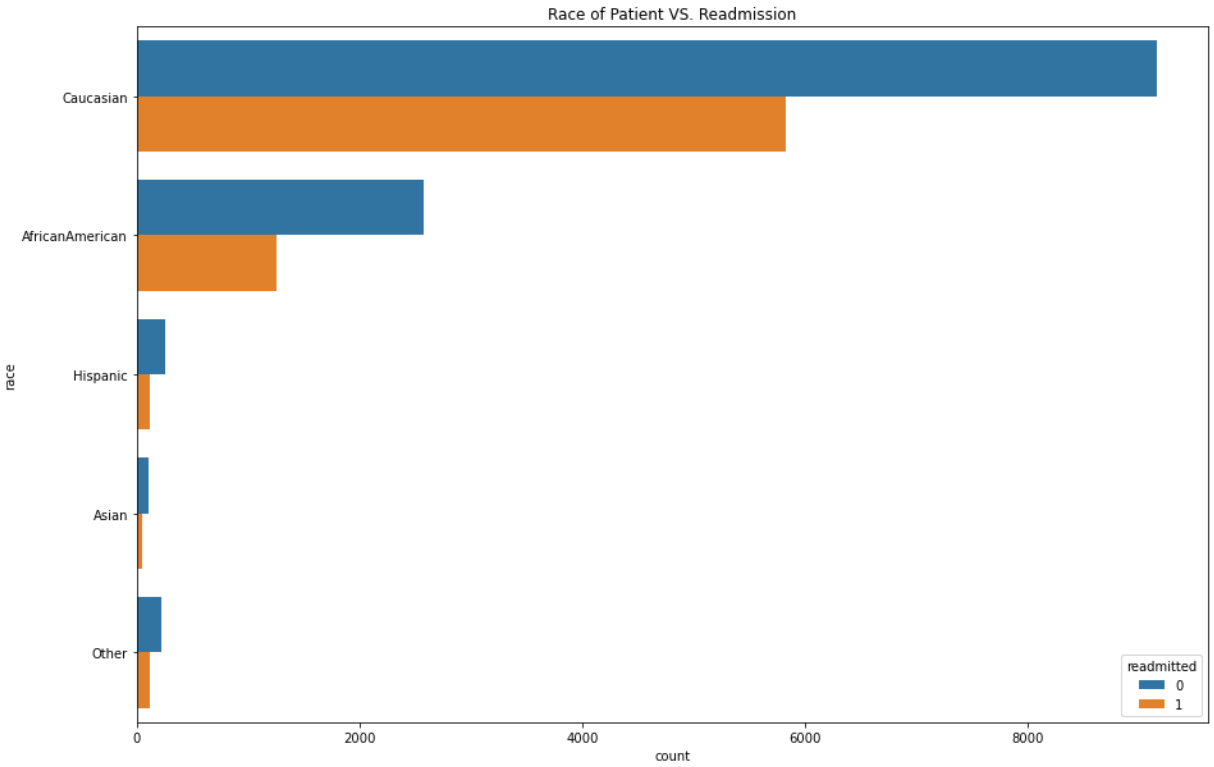


Figure 4: Percentages of all race groups and readmission

	race	Has been Readmitted (%)	Has not been Readmitted (%)
0	AfricanAmerican	32.77	67.23
1	Asian	31.87	68.12
2	Caucasian	38.89	61.11
3	Hispanic	31.99	68.01
4	Other	35.01	64.99

In figure 3, the graph represents the impact race has on readmissions. Similar structure to Figure 1 with the orange column representing who has been readmitted and the blue column which has not been readmitted. Figure 4 presents the percentage of readmissions for each ethnic group, which shows that African Americans have the third-highest readmission rate with 32.77%, compared to Caucasians, which have 38.89%. The group is called other, which has 35.01%. A relationship between race and readmissions cannot be seen. Therefore, race does not have any impact on whether a patient would be readmitted again.

The relationship between gender and readmission will be explored next.

Figure 5: Histogram showing gender impact on readmissions

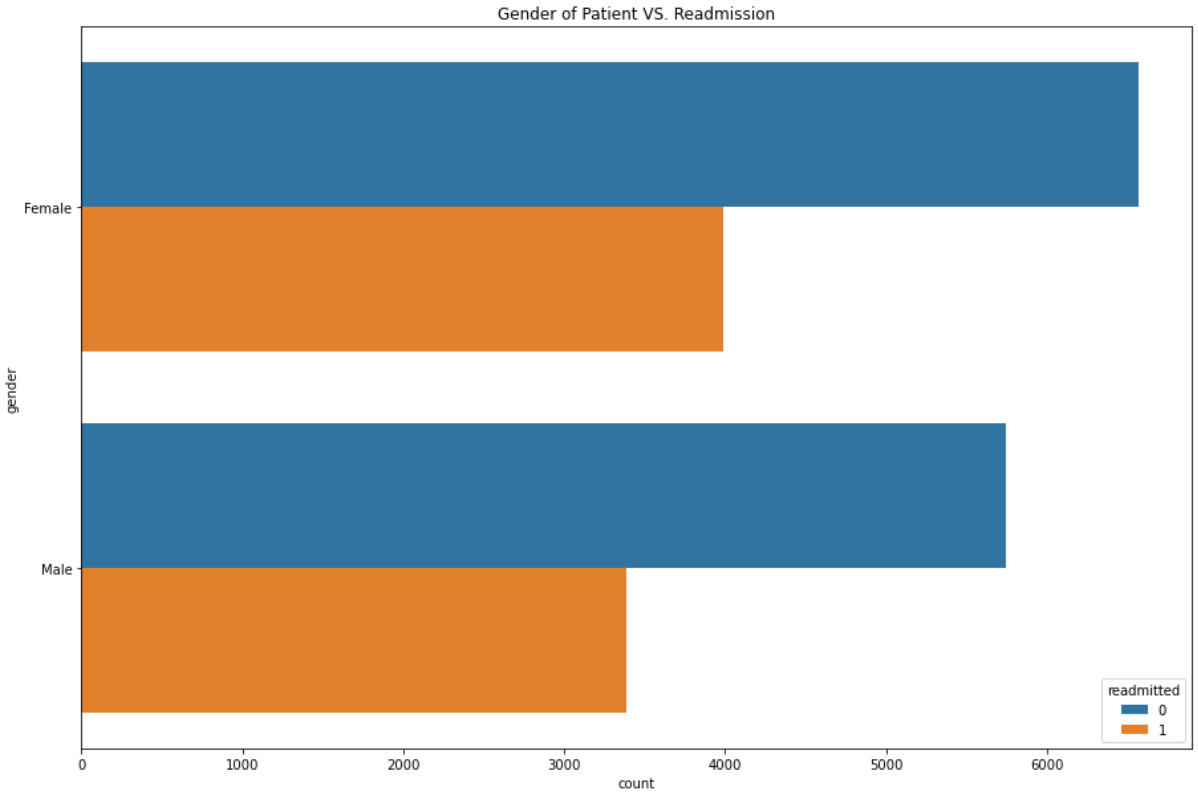


Figure 6: Percentages of both gender groups and readmission

	gender	Has been Readmitted (%)	Has not been Readmitted (%)
0	Female	37.75	62.25
1	Male	37.09	62.91

In figure 6, the histogram represents gender impact on readmissions. Overall, more women patients were being readmitted than men patients. This proves the hypothesis that women are more likely to be readmitted than men. On the table, the woman's section is larger due to a higher number of female patients than males, a percentage shown in figure 6, which was calculated to find if gender impacts readmissions. Women have a slightly higher readmission rate than the very close values of a difference of 0.66% between male and female readmissions, which could be seen as insignificant.

Still, it would be seen as valid for the hypothesis of women being more likely to be readmitted than men.

The following section will discuss the results on whether diagnosis types impact readmission rates.

Figure 7: Diagnosis types impact on readmissions

	Description	Has been Readmitted (%)	Has not been Readmitted (%)
0	Abnormality of forces of labour	8.33	91.67
1	Abnormality of organs and soft tissues of pelvis	8.70	91.30
2	Abscess of anal and rectal regions	42.11	57.89
3	Abscess of lung and mediastinum	50.00	50.00
4	Acquired deformities of toe	50.00	50.00
...	...	...	...

For this section, we had drawn histograms to visualise if a trend exists between diagnosis types and readmission. As there were too many graphs, none were placed in this report, but they can be found in the jupyter file.

Figure 7 shows the percentage of patients who have been readmitted and those who have not, according to their diagnoses received. Diagnoses do not impact re-admission rates. After carefully examining each graph, it can be concluded that the statement is wrong.

Part 1.3 Model Building:

The following subset columns provided were used as the features while targeting the readmitted column:

['num_medications', 'number_outpatient', 'number_emergency', 'time_in_hospital', 'number_inpatient', 'encounter_id', 'age', 'num_lab_procedures', 'number_diagnoses', 'num_procedures', 'readmitted']

After this step, the data frame was immediately split into training and test sets with a test size of 20% and a training size of 80%. We used the logistic regression model as this was more time-efficient than other models; this was prioritised over accuracy. Once this was completed, a confusion matrix was plotted.

A confusion matrix is used to evaluate and summarise the model's performance.

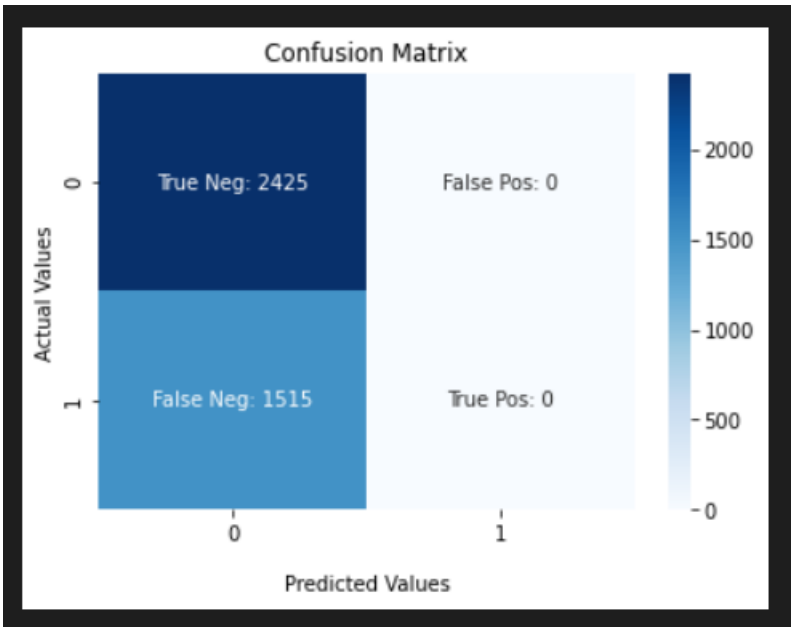
True negative values on the confusion matrix below show the number of negative values classified accurately; in this model, the true negative was 2425, which means that the model has accurately predicted that 2425 patients will not be readmitted.

False-negative shows the number of actual positive values that are classified as negative. In this model, the false-negative value was 1515, which means that this model predicted 1515 patients to be readmitted, which was not the case.

False-positive shows the number of actual negative values that are classified as positive; in this model, the false positive was 0, which means that the model did not falsely predict any patient as readmitted when they weren't.

True positive shows the number of positive values classified accurately; in the model, the true positive value was 0, which means there are no correctly classified values.

Figure 8: Confusion matrix of logistic regression model



Following the confusion matrix, a cross-validation evaluation was made to find the model's accuracy, which was 0.63.

The classification report helps in checking how accurate a model is; the report is shown below returns precision, recall and f1-scores for each class; with the help of the classification report, we can see the precision and recall of 1 is 0.00, so the f1-score for it would also be 0.00, this means the model is not as accurate as it only predicting not readmitted values. This would need to be fixed to improve the model.

Figure 9: Classification report of logistic regression model

	precision	recall	f1-score	support
0	0.62	1.00	0.76	2425
1	0.00	0.00	0.00	1515
accuracy			0.62	3940
macro avg	0.31	0.50	0.38	3940
weighted avg	0.38	0.62	0.47	3940

Part 2.1 – Improved model

For the first part of trying to improve the model, random over_sampling was be used. We began by splitting the readmitted column into classes of 0 and 1. Class 1 contained all the one values (Has been readmitted), and class 0 contained the zero values (Has not been readmitted).

The results were as follows:

```
Class 1 Count: 7375
Class 0 Count: 12321
```

From the results above, there is a clear imbalance between class 1 and 0, so we try to remove the balance by using the random over_sampling method; this will increase the count of class 1 to make the results more even.

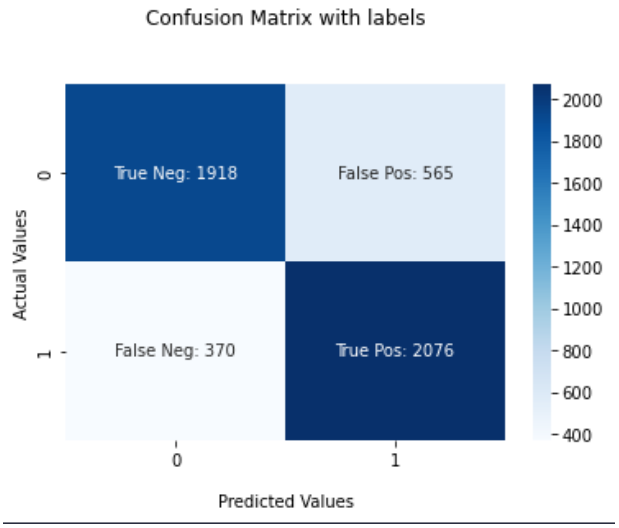
Afterwards, both classes are concatenated together, and the data is split into train and test data with a ratio of 80:20. This is the same ratio we used in the logistic regression model above (20% test size and 80% train size).

To get a better model, we sometimes use many pre-processing techniques like normalisation, transformation, removing or adding some features or missing values before training the model. We have to apply each method individually to data before sending it for model training and creating many variables to store them. We used a pipeline instead, which helps us combine all steps we want to perform on data. Pipeline applies all steps on data in sequence and sends the result of the previous step to the next step.

In our project, we combined Standard Scaler, which is used to normalise data, with Random Forest classifier using pipeline instead of applying both steps individually on data. Pipeline will always normalise data first using Standard Scaler then use results of Standard Scaler to train random forest classifier. When we predict a value for test data first, it will always normalise the test data and then send the result of Standard Scaler to the classifier to predict the outcome. A pipeline is used to automate the complete pre-processing and model training steps in simple wording.

After normalising the data, a confusion matrix was plotted with predicted values against actual values.

Figure 10: Confusion matrix for improved model



The Confusion matrix shows the model predicted 1918 patients to be readmitted correctly (True negative). It has wrongly predicted that 565 patients would be readmitted, which is higher than our logistic regression model, and they were not (False positive). 370 patients were falsely predicted to be readmitted but were not; this value is much lower as compared to the first model, which falsely predicted 1515 patients showing that this improved model is more accurate (False negative). Lastly, 2076 patients were predicted not to be readmitted and were not (false negative); this shows an improvement from the first model, which accurately predicted 0 patients not to be readmitted.

The improved model was then run through k-fold cross-validation that split the data into 10 random groups. We had used this because it takes each group as a holdout or test data set and compares it to the remaining groups as a training data set. A model was created, then combining the standard Scalar and Random Forest classifier. The cross-validation then fits the model created onto the training set and runs it against the test set. This gives us an evaluation score that shows how accurate our model is.

Our classification report showed the following results:

Figure 11: Classification report for improved model

	precision	recall	f1-score	support
0	0.84	0.77	0.80	2483
1	0.79	0.85	0.82	2446
accuracy			0.81	4929
macro avg	0.81	0.81	0.81	4929
weighted avg	0.81	0.81	0.81	4929

From the report, the precision section tells us that the model predicted those not to be readmitted with 84% accuracy and accurately predicted 79% of the readmitted patients to be readmitted. This is a higher value than the logistic regression, which predicted 62% of patients not to be readmitted but could not predict which patients would be. Furthermore, the recall shows that the model had predicted 85% of the readmitted patients to be readmitted, which is significant compared to the call for the first model, which was 0%.

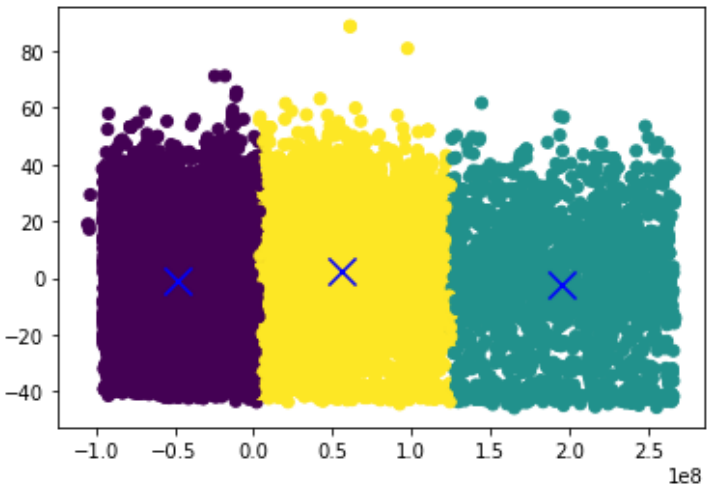
Sometimes, checking accuracy using a simple accuracy formula (correct predictions divide by total predictions) gives us high accuracy even if the model is not accurate. For example, actual values are 0, 0, 0, 0, 1, 0, 0, 1, 0, 0 and model predicts 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 so when we check accuracy it will be 80% but in predictive models, it tends to predict 0 so it will never predict 1 so the model is not accurate. This issue is due to class imbalance during the training of the model. This was the case for the logistic regression model as it only predicted the value 0 and not 1. The F1-score shows accuracy using the recall and precision values. In the same situation, it will not give high accuracy. The F1 score is a weighted harmonic mean of precision and recall such that the best score is 1.0 and the worst is 0.0.

Therefore, the classification report helps in checking how the accurate model is. The accuracy after the evaluation was 0.83. This is much higher as compared to the logistic regression model, which only yielded 0.63.

Part 2.2 – K means algorithm and clustering.

Clustering divides data points into a number of groups such that data points in the same groups are more like other data points in the same group than those in other groups. In simple words, the aim is to segregate groups with similar traits and assign them into clusters. In our project, the number of features in the dataset is more than 2 as we cannot visualise more than 2 or 3 features simultaneously, so we reduced the number of features using Principal component analysis (PCA). PCA helps to reduce data dimensions that still contain most of the critical information in data. It can slightly minimise accuracy, but also provides fewer dimensions, and the data becomes easier to explore and visualise.

Figure 11: PCA cluster



In our cluster, we used 2 features and split it into 3 clusters using the Kmeans algorithm.

In the plot above, we visualise a scatter plot using 2 features reduced from the data using PCA and divided into 3 clusters using the Kmeans algorithm. Points in each group have more similarities than points in other groups.

Part 2.3 –Any decisions that should be taken from the model

From the improved model, it should be noted that random over_sampling has been used because both classes are imbalanced, which reduces the model's accuracy. Hence, using this method aids in removing the imbalances. Other techniques, such as random under_sampling, could be used to handle the imbalances within the classes.

For clustering, the principal component analysis technique is used to reduce the data dimensions to help visualise it more; this is because anything more than 3 dimensions becomes challenging to visualise.