ASSIGNMENT TWO

PAPER NAME: Data Mining and Machine Learning
PAPER CODE: COMP615

TOTAL MARKS: 100

Students' Names: Anjali Kodagoda & Mirasha Fernando

Students' IDs: 2 3 2 0 9 0 9 9 & 2 1 1 5 1 1 4 4

- Due date: 08 Jun 2025 midnight NZ time.
- Late penalty: maximum late submission time is 24 hours after the due date. In this case, a **5% late penalty** will be applied.
- Submit the actual code (no screenshot) separately with appropriate comments for each task.

Note: This assignment should be complemented by a group of two students and both students MUST contribute in each part.
Submission: a soft copy needs to be submitted through the canvas assessment link.

INSTRUCTIONS:

1. **The following actions may be deemed to constitute a breach of the General Academic Regulations Part 7: Academic Discipline,**
   - Communicating with or collaborating with another person regarding the Assignment
   - Copying from any other student work for your Assignment
   - Copying from any third-party websites unless it is an open book Assignment
   - Uses any other unfair means
2. **Please email DCT.EXAM@AUT.AC.NZ if you have any technical issues with your submission on Canvas immediately**
3. **Attach your code for all the datasets in**.

Table of Contents

List of Figures

Part A

a) KNN and Naïve Bayes Algorithms

K-Nearest Neighbors (KNN)

This is a simple and intuitive algorithm used mainly for classification and sometimes regression. It works on the idea that similar data points are likely to belong to the same class. When a new data point needs to be classified, the algorithm looks at the closest data points (neighbors) in the training set. It checks the labels of these neighbors and assigns the new data point to the most common class among its K neighbors.

For example, if K=5, and three of the closest neighbors belong to Class 1 and two neighbors belong to Class 2, the algorithm will classify the new data point as Class 1. This method is simple to understand and easy to implement but it can be slow with large datasets and is sensitive to irrelevant features.

Naïve Bayes (NB)

This is a classification algorithm that uses probability to predict which class a data point belongs to, with a strong assumption that all features are independent of each other. It calculates the probability of each class given the input features using Bayers' Theorem: P(Class|Data) = {P(Data|Class) * P(Class)} / P(Data). For each class, it computes the likelihood of the input features and assigns the class label with the highest probability. This method is fast, efficient but it may not perform well if feature independence is violated.

b) Perform Exploratory Data Analysis (EDA)

The dataset used in this assignment comes from a study on detecting Parkinson's disease using speech signals. Parkinson's disease is one of the most common neurological conditions mostly affecting older individuals. It is a disorder that affects movements and coordination due to the loss of brain cells that produce dopamine. The data were gathered from 188 patients with Parkinson's disease (107 men and 81 women) and 64 healthy individuals (23 men and 41 women). It includes various speech signal processing features extracted from recordings of both Parkinson's patients and healthy individuals.

Throughout this assignment, we aim to analyse the Parkinson's Disease speech features dataset to explore patterns and build three classification models and choose the best-performing model. The purpose of this analysis is to find whether a person has Parkinson's disease based on speech characteristics.

There are 756 instances in this dataset and 754 features (+ target variable) originally. These features include quantitative features extracted from voice recordings (e.g.: amplitude, noise metrics, frequency). Most of these features are continuous numerical variables. The target variable is Class, which is a binary classification of 1 and 0, with 1 being Parkinson's disease and 0 being healthy individuals.

The first step of preprocessing and data cleaning is understanding the dataset through summary statistics and then checking for missing values and outliers. Due to the large number of continuous features, we provide a representative sample of summary statistics for 10 features.

	count	mean	std	min	25%	50%	75%	max
PPE	756.0	0.746284	0.169294	4.155100e-02	0.762833	0.809655	0.834315	0.907660
DFA	756.0	0.700414	0.069718	5.435000e-01	0.647053	0.700525	0.754985	0.852640
RPDE	756.0	0.489058	0.137442	1.543000e-01	0.386537	0.484355	0.586515	0.871230
numPulses	756.0	323.972222	99.219059	2.000000e+00	251.000000	317.000000	384.250000	907.000000
numPeriodsPulses	756.0	322.678571	99.402499	1.000000e+00	250.000000	316.000000	383.250000	905.000000
meanPeriodPulses	756.0	0.006360	0.001826	2.107089e-03	0.005003	0.006048	0.007528	0.012966
stdDevPeriodPulses	756.0	0.000383	0.000728	1.060000e-05	0.000049	0.000077	0.000171	0.003483
locPctJitter	756.0	0.002324	0.002628	2.100000e-04	0.000970	0.001495	0.002520	0.027750
locAbsJitter	756.0	0.000017	0.000023	6.860000e-07	0.000005	0.000010	0.000018	0.000256
rapJitter	756.0	0.000605	0.000981	2.000000e-05	0.000150	0.000280	0.000650	0.011050

Figure 1:Summary Statistics

Figure 1 shows the summary statistics of the first 10 continuous variables in the dataset. PPE, DFA, and RPDE show relatively high mean values (around 0.49 – 0.75) and moderate variability. This indicates their potential in capturing vocal signal irregularities which are common indicators in Parkinson's patients. If we look at PPE, it has a notable high mean (0.746) and a narrow interquartile range meaning that most subjects exhibit elevated PPE levels, with outliers at both extremes. On the other hand, numPulses and numPeriodsPulses have large standard deviations (approx. 99), depicting a wide variability in voice frequency patterns across individuals. The jitter-related measures (locPctJitter, locAbsJitter, rapJitter) have smaller means but are highly skewed with long tails. This suggests that although most subjects have low jitter, some show significant vocal instability.

There were no missing values in this dataset (Appendix). We will now look at the outliers and see if they should be treated.

```
Total instances: 756
Number of outliers: 678
```

Figure 2:Outlier Count

As you can see figure 2 shows that there are 678 outliers based on the z-scores of the variables. Let's look at some of the visualizations to better understand the outliers. Based on the z-scores, below figure shows the boxplots of the top 5 variables that has the highest number of outliers.

```
Top 5 features with the most outliers:
tqwt_entropy_log_dec_1       31
tqwt_skewnessValue_dec_27    31
tqwt_entropy_log_dec_3       30
tqwt_entropy_log_dec_2       30
tqwt_kurtosisValue_dec_32    29
dtype: int64
```

Figure 3:Top features with the most outliers

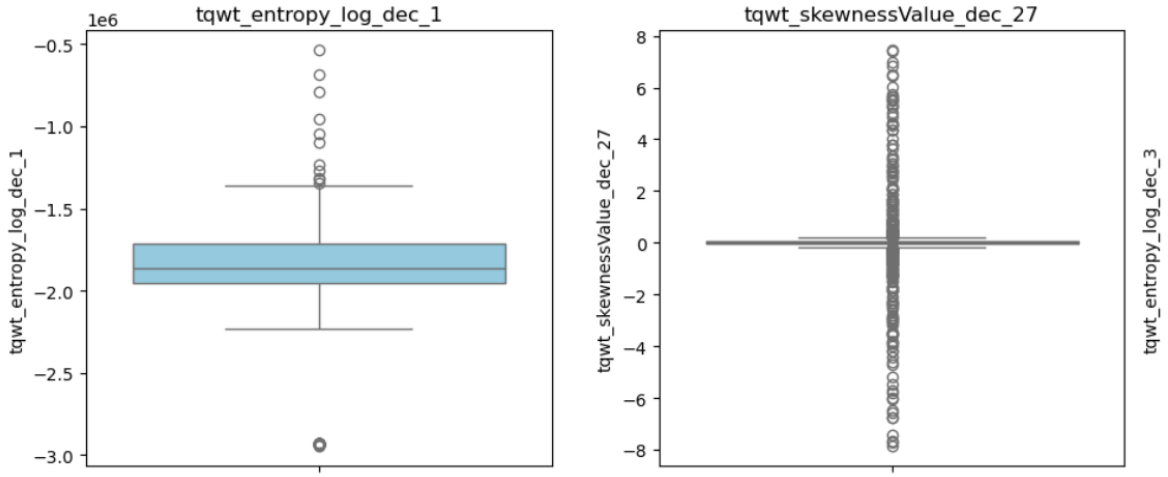


Figure 4:Boxplots of Top 5 features

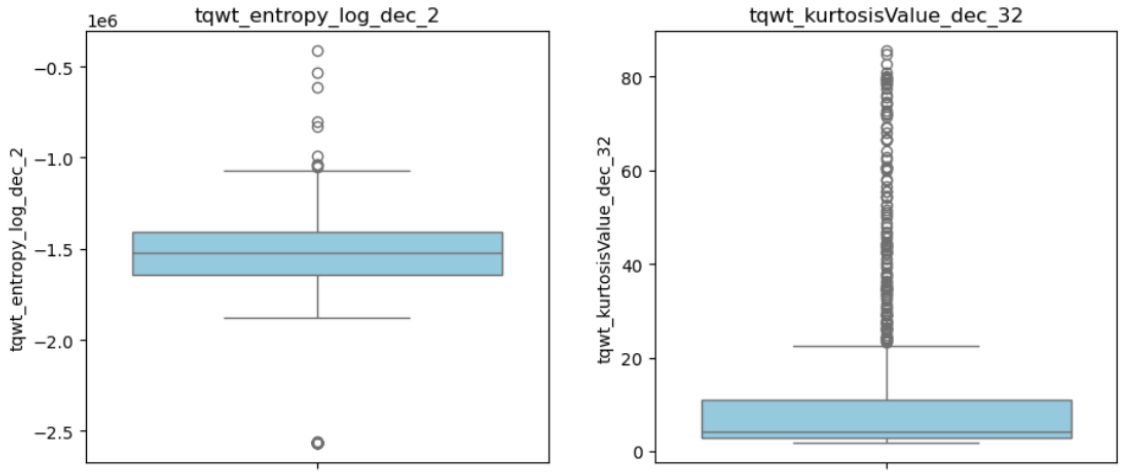


Figure 5:Boxplots of top 5 features

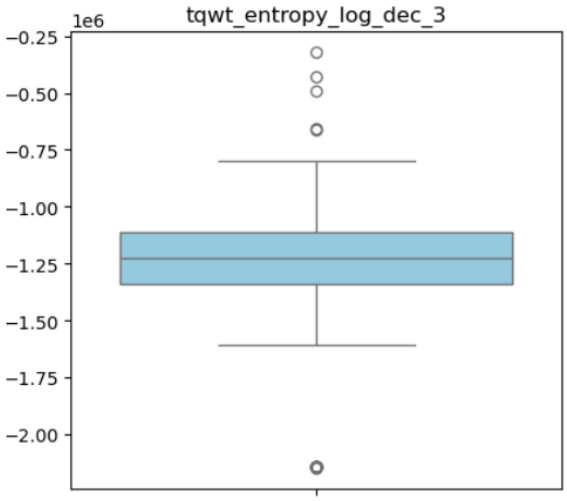
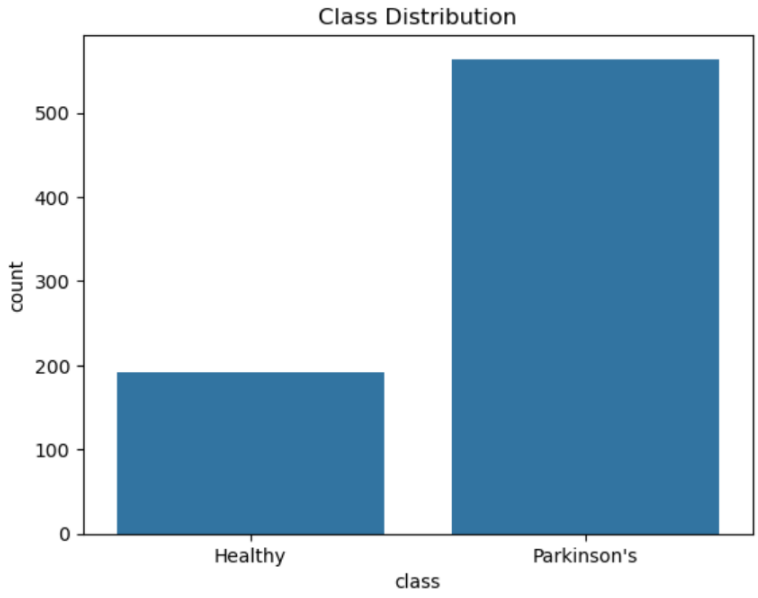


Figure 6:Boxplots of top 5 features

The purpose of treating outliers is to improve model performance and stability. However, it depends on the type of dataset. We know that this is a medical dataset and represents clinical measurements like PPE (Pitch Period Entropy), DFA (Detrended Fluctuation Analysis), RPDE, (Recurrence Period Density Entropy) and many more. In this case, outliers are not exactly errors or anomalies. They may reflect advanced stages of Parkinson's disease, unusual but valid physiological conditions or individual differences in disease progression. Therefore, removing or modifying them would erase the very important data that differentiate healthy individuals from the patients.

Medical datasets often have skewed or non-normal distributions and by forcing them into a fixed shape may unintentionally bias the model. For example, a patient with high jitter may be a valid case of Parkinson. If removed, they the model we intend to create might underperform. Instead, we can standardize or normalize the data so that all features proportionally contribute to the calculations. (without eliminating extreme but informative values).




```
class
1    74.603175
0    25.396825
Name: proportion, dtype: float64
```

Figure 7:Class distribution

Above figure shows the distribution of the class variable and we can see that the dataset is imbalanced. There are more Parkinson's patients than healthy individuals. More than 50% is of class 1 which is Parkinson's (74.6%) and only 25.4% is of class 0 (healthy).

More exploration on the data will be done after feature selection is done in the next step.

c) Feature Selection and Analysis

In this analysis, we chose ANOVA (Analysis of Variance) as the method for selecting the top five features suitable for modelling. This method is mostly used for identifying features that have significantly different means across classes which makes it more suitable to determine to most influential predictors.

ANOVA assumes normality and homogeneity of variance within groups which can be affected by the outliers which we decided not to remove due to its importance. It is normal to not go by the assumptions when the dataset is considerably large (756 instances in our dataset). Since our goal is to rank features based on their relevance rather than performing hypothesis testing, the outliers are unlikely to make a huge impact on the results.

Furthermore, the features in this dataset represent clinical observations and extreme values may contain important information relevant to Parkinson's disease. Therefore, we decided to retain the outliers and proceed with ANOVA for feature selection.

Top 20 features based on ANOVA F-Score:

	Feature	F-Score
0	mean_MFCC_2nd_coef	142.506911
14	tqwt_minValue_dec_12	140.011407
11	tqwt_stdValue_dec_12	137.750103
17	tqwt_maxValue_dec_12	136.312113
10	tqwt_stdValue_dec_11	136.142253
8	tqwt_entropy_log_dec_12	128.039591
16	tqwt_maxValue_dec_11	126.070781
13	tqwt_minValue_dec_11	117.032224
15	tqwt_minValue_dec_13	116.085095
5	std_9th_delta_delta	115.403308
4	std_8th_delta_delta	115.347306
18	tqwt_maxValue_dec_13	113.563606
3	std_7th_delta_delta	108.192500
12	tqwt_stdValue_dec_13	107.502251
2	std_6th_delta_delta	104.606602
6	tqwt_entropy_shannon_dec_11	103.825711
19	tqwt_kurtosisValue_dec_27	103.625917
1	std_8th_delta	100.452493
7	tqwt_entropy_log_dec_11	100.088566
9	tqwt_TKEO_std_dec_12	98.814312

Figure 8:Top 20 features based on ANOVA F-score

The above figure gives the top 20 features based on the ANOVA F-score. The ANOVA F-score measures the ratio of variance between classes to the variance within classes. This highlights the features that are most likely to differ significantly between the Parkinson's and the healthy.

We will be selecting the below given top 5 as some features are the statistical summaries of the same group. The top 5 features identified are:

1. **mean_MFCC_2nd_coef** – This has the highest F-score (142.51), indicating it strongly differentiates between the two classes. MFCC (Mel-Frequency Cepstral Coefficients) are commonly used in audio analysis, and the second coefficient may capture subtle vocal characteristics affected by Parkinson's disease.

2. **tqwt_minValue_dec_12** – The minimum value in the 12th level of transform using the Tuneable Q-factor Wavelet Transform (TQWT). Its high F-score means distinct signal behaviour between classes at this frequency decomposition.

3. **tqwt_entropy_log_dec_12** – This feature calculates the logarithmic entropy, a measure of complexity, at the same low-frequency sub-band. When evaluating the signal's unpredictability or irregularity-both of which are frequently impacted by neurodegenerative diseases-entropy is helpful.

4. **std_9th_delta_delta** – This is the standard deviation of the 9th delta-delta coefficient, which monitors spectral feature changes over time.

5. **tqwt_entropy_shannon_dec_11 –** An 11tth sub-band Shannon entropy metric.

Below figures give the boxplots for the top 5 selected features.

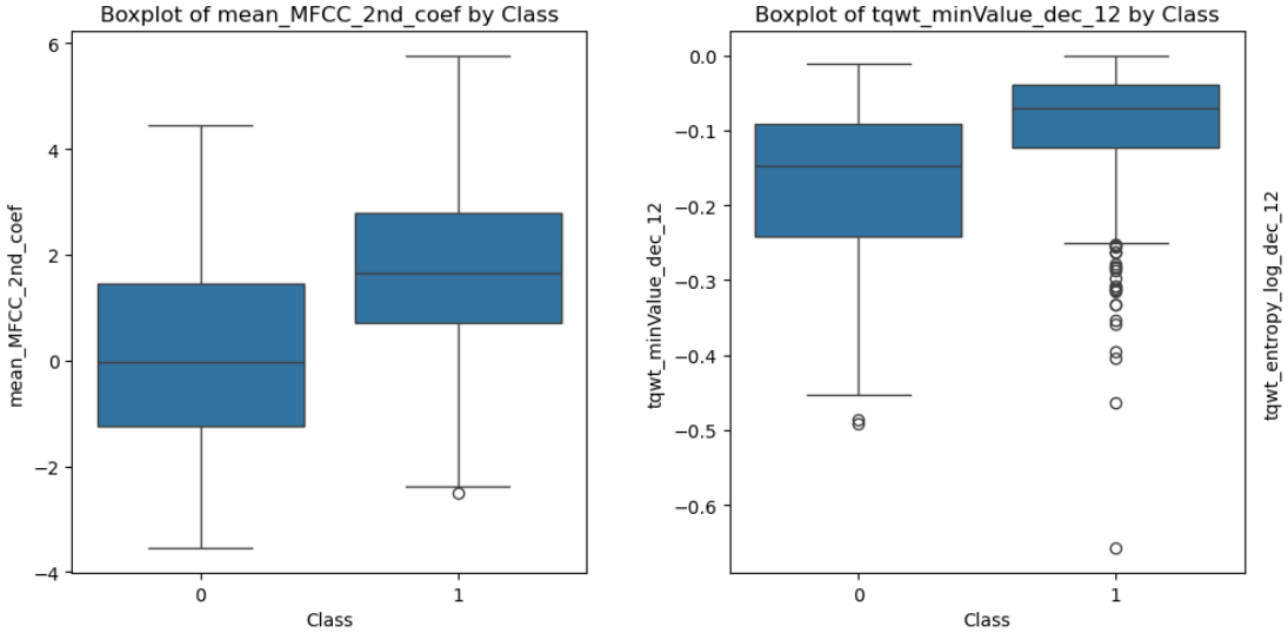


Figure 9: Top 5 Feature distribution by class

mean_MFCC_2nd_coef – shows a clear difference in the median values between healthy & Parkinson's patients. The Parkinson's has higher values.

tqwt_minValue_dec_12 – Parkinsons patients have a higher range while healthy shows more spread. Class 1 values are generally closer to zero compared to class 1.

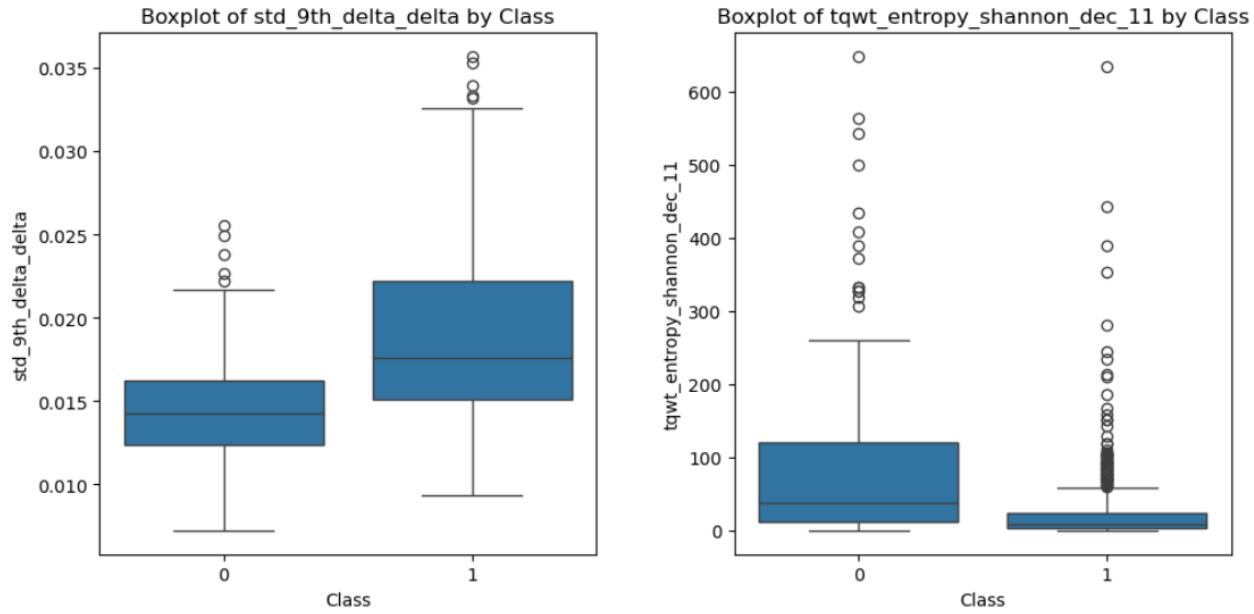


Figure 10:Top 5 Feature Distributions by class

std_9th_delta_delta: Plot shows higher median for Parkinsons, and it has more outliers. Some difference in variability for this feature between the classes.

tqwt_entropy_shannon_dec_11: Class 0 shows a wider spread and higher entropy values compared to class 1. Indicates strong skewness and variability.

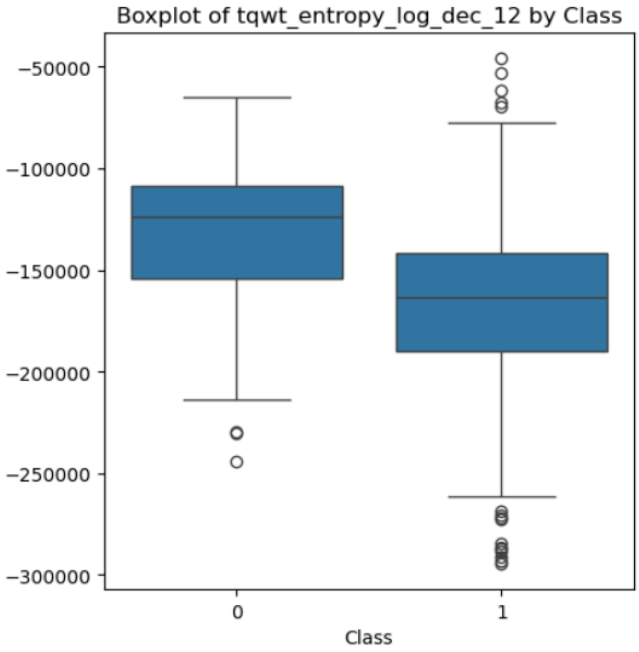


Figure 11:Top 5 Feature Distribution by class

tqwt_entropy_log_dec_12: both distributions are heavily skewed left. Class 1 has a slightly lower median and is significantly spread.

d) Independence Assumption in Naïve Bayes

The Naïve Bayes algorithm assumes all features used are conditionally independent given the class label. It assumes knowing the value of one feature doesn't give any information about the value of another within each class.

To evaluate this assumption, we can use the correlation heatmap which gives the feature correlations. If features are highly correlated it violates the assumption. Below figure shows the correlation heatmap of the chosen features.

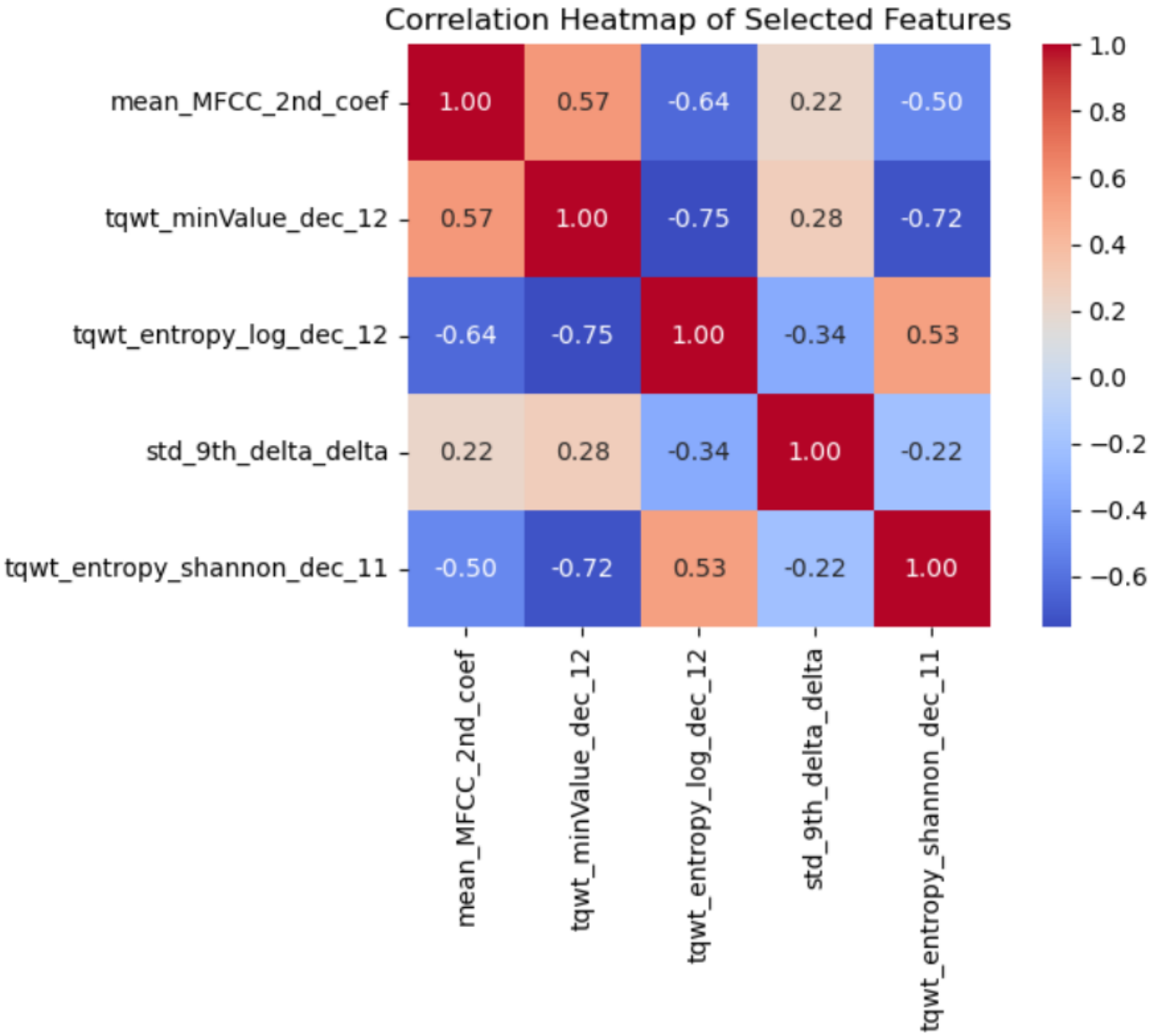


Figure 12:Corelation Heatmap

The correlation matrix of these features reveals that several of them are moderately to strongly correlated, particularly the TQWT features. For example, tqwt_minValue_dec_12 and tqwt_entropy_log_dec_12 have a correlation of -0.75 indicating a strong inverse relationship. This violates the independence assumption of Naïve Bayes. This can affect the model performance and reduce predictive accuracy.

e) Naïve Bayes Model Building and Evaluation

The Gaussian Naïve Bayes classifier was trained using the five selected features after applying standardization to ensure they were on the same scale. The model is evaluated on a test set using accuracy, precision, recall, and F1-score. Below figures show the confusion matrix and the evaluation metrics.

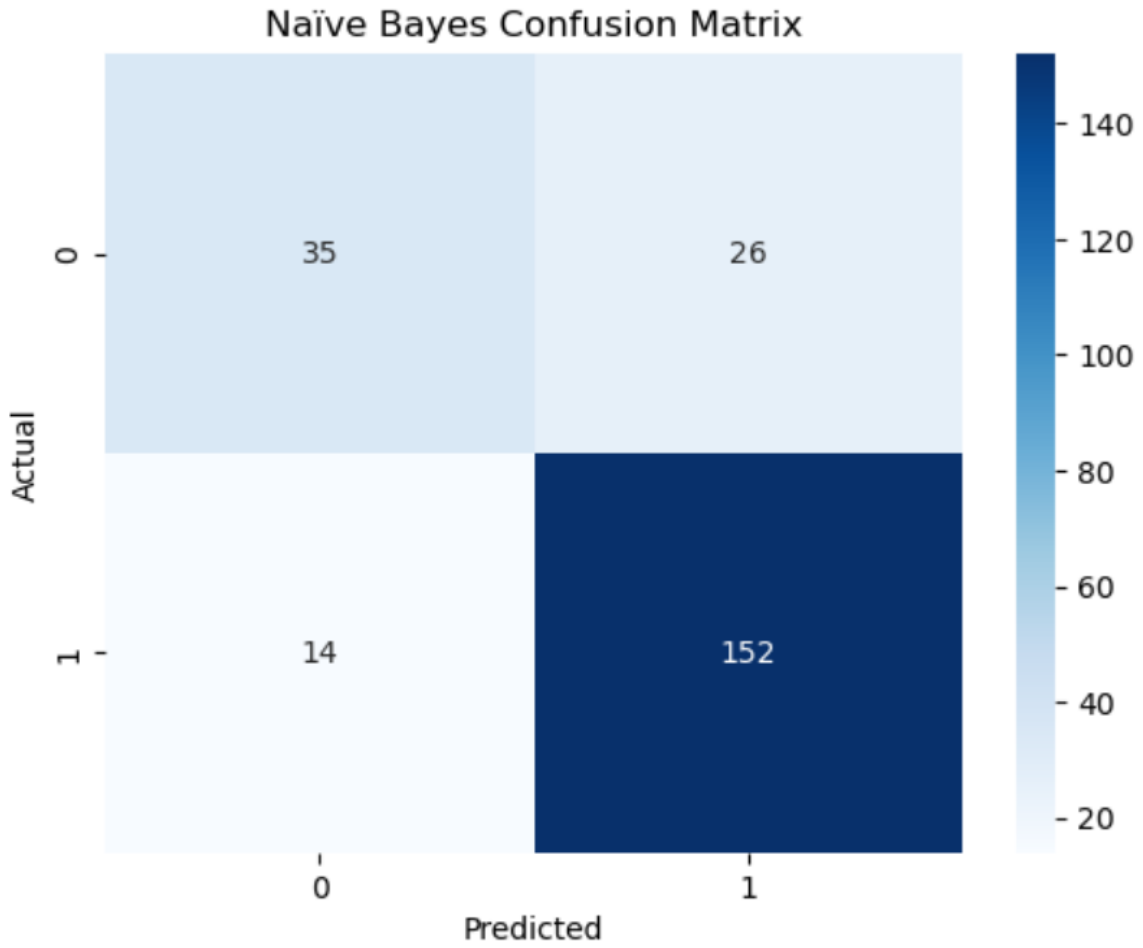


Figure 13:Naive Bayes Confusion Matrix

The confusion matrix provides a detailed breakdown of the model's predictions across the two classes: Healthy (0) and Parkinson's (1).

152 Parkinson's patients were correctly classified (true positives) and 35 healthy individuals too (true negatives). The model incorrectly classified 26 healthy individuals as Parkinson's patients (false positives). It also misclassified 14 Parkinson's patients as healthy individuals (false negatives).

The confusion matrix shows that the model is better at predicting Parkinson's than Healthy. While the high number of true positives shows the model's strength in detecting Parkinson's, the high number of false positives for the healthy class (26) indicates a notable rate of misclassification, which could lead to false alarms in a medical setting. This imbalance reflects a bias towards the majority class (Parkinson's), possibly due to the class distribution in the dataset (approx. 75% Parkinson's).

Let's look at the evaluation metrics for the model performance.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.71      0.57      0.64        61
           1       0.85      0.92      0.88       166

    accuracy                           0.82       227
   macro avg       0.78      0.74      0.76       227
weighted avg       0.82      0.82      0.82       227

Accuracy: 0.8237885462555066
```

Figure 14:Classification Report – Naïve Bayes

When the model predicts Parkinson's, it is correct 85% of the time which means only a few false positives (Precision – 0.85). The model identifies 92% of actual Parkinson's cases which shows a strong sensitivity to the disease (Recall – 0.92). However, only 57% of healthy individuals are correctly identified, showing the model struggles with that class (Recall – 0.57). The F1-score balances the precision and recall. The F1 for Parkinson's is excellent (0.88), but its low for Healthy (0.64). This shows an imbalance in predictive power.

Overall performance accuracy is 82% suggesting solid performance but it can be misleading sometimes with unbalanced datasets. As we look closely, we can see that the macro average is lower than the weighted average. Meaning that the model favors the dominant class (Parkinson's). Despite imbalance, the model maintains strong generalization and balanced accuracy but there is room for improvements.

f) KNN Model Building and Evaluation

Similar to the Gaussian Naïve Bayes classifier, the same features were fit into the KNN model. In KNN, k represents the number of nearest neighbours used to classify a sample. The model was fitted to different k values (1-21). Below figure shows the accuracy levels at each k value.

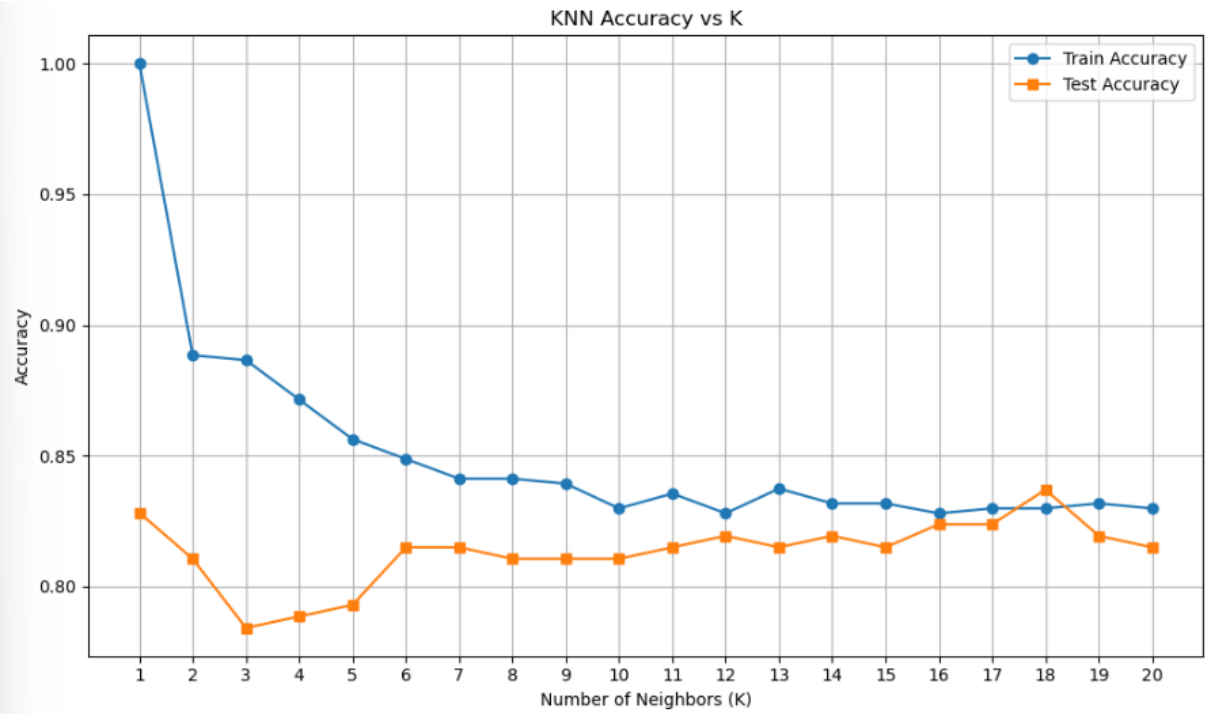


Figure 15:KNN Accuracy

Based on the plot, the best K is 18 with an accuracy of 83.70% in the test set. The value strikes a balance between underfitting and overfitting. Now we can use k=18 and generate the confusion matrix and the classification report.

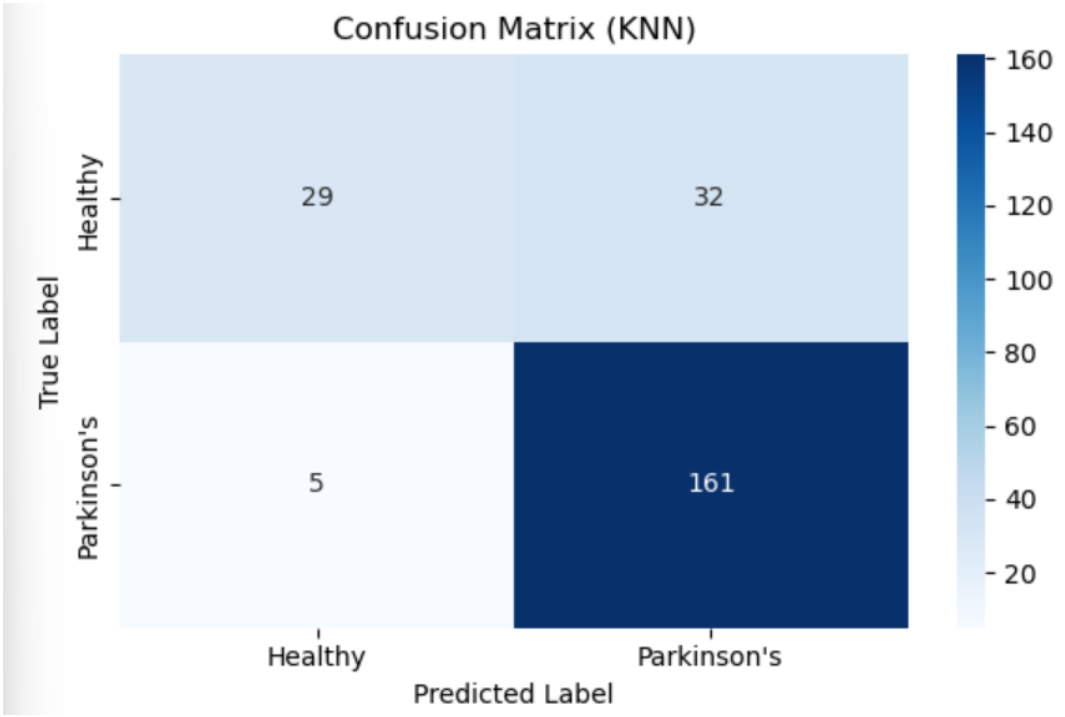


Figure 16:Confusion Matrix - KNN

The model identified Parkinson's cases effectively (true positives – 161). It also identified 29 Healthy cases which is lower than Parkinson's (true negatives). The model misclassified 32 Healthy individuals as Parkinson's (false positives) and 5 Parkinson's patients as Healthy (false negatives).

Overall, the model had difficulty in identifying Healthy individuals. On the other hand, we can see that false negatives are lower than false positives which can be seen as a positive mark as it is less dangerous to misclassify a Healthy person than missing a patient with Parkinson's.

Below figure has the classification report.

Classification Report:

	precision	recall	f1-score	support
Healthy	0.85	0.48	0.61	61
Parkinson's	0.83	0.97	0.90	166
accuracy			0.84	227
macro avg	0.84	0.72	0.75	227
weighted avg	0.84	0.84	0.82	227

Figure 17:Classification Report - KNN

The KNN model achieved an accuracy of 84.14% and demonstrated excellent performance in detecting Parkinson's disease. With a recall of 0.97 and a precision of 0.83, it is confirmed that the model catches nearly all Parkinson's patients. The F1-score (0.90) has a very good overall balance between precision and recall. On the other hand, the F1-score of Healthy individuals is 0.61 which is lower due to poor recall (0.48). As said earlier only 48% of real Healthy cases were correctly identified and many were misclassified.

g) Model Comparison

With an ideal k value of 18, the KNN (K-Nearest Neighbors) model outperformed the Naïve Bayes (NB) model by a small margin, achieving 84.14% test accuracy. Additionally, KNN showed a higher recall (97%) for the Parkinson's class than NB, which only showed a 92% recall for the same class. This implies that KNN performed better at detecting Parkinson's disease, which is crucial in reducing false negatives in a clinical setting. Nevertheless, KNN's recall (48%) for the Healthy class was comparable to NB's (57%), suggesting that both models had some difficulty differentiating between Healthy people in this unbalanced dataset.

From the standpoint of model behaviour, KNN depends on distance metrics and is more susceptible to noise and feature scaling, whereas Naïve Bayes is quicker and easier, presuming feature independence. Although its assumptions were partially broken (as the correlation heatmap illustrates), NB managed the unbalanced dataset quite effectively, whereas KNN needed to have K adjusted in order to function at its best. Although KNN demonstrated a slight performance advantage and greater adaptability overall, NB is still a solid baseline model, particularly considering its ease of use and effectiveness.

Part B

Exploring Artificial Neural Networks

a) Activation Function and Learning Rate in MLP

Activation Function is used in each neuron, and it adds nonlinearity to the model. Without activation function, MLP would be just linear calculations, and it wouldn't be able to learn complex patterns in the data. This is important because without it, no matter how many layers the MLP has, it will still behave like a linear model. So basically, the activation function makes the network capable of learning complicated, nonlinear relationships. It makes the MLP model more powerful.

Learning rate is a value that controls how fast or slow the model learns and affects how fast and accurately it learns. It shows up in the formula:

$$\Delta wjk = r\,Error(k)\,Oj$$

Here 'r' is the learning rate. If the learning rate is too high, the module might never find the best solution. If the learning rate is too low, training becomes slow and might get stuck. Finding the right learning rate is important for effective and stable learning.

b) Baseline Model with MLPClassifier

```
Best accuracy: 0.8882 at 200 iterations
```

Figure 18: Highest accuracy and best no: of iterations

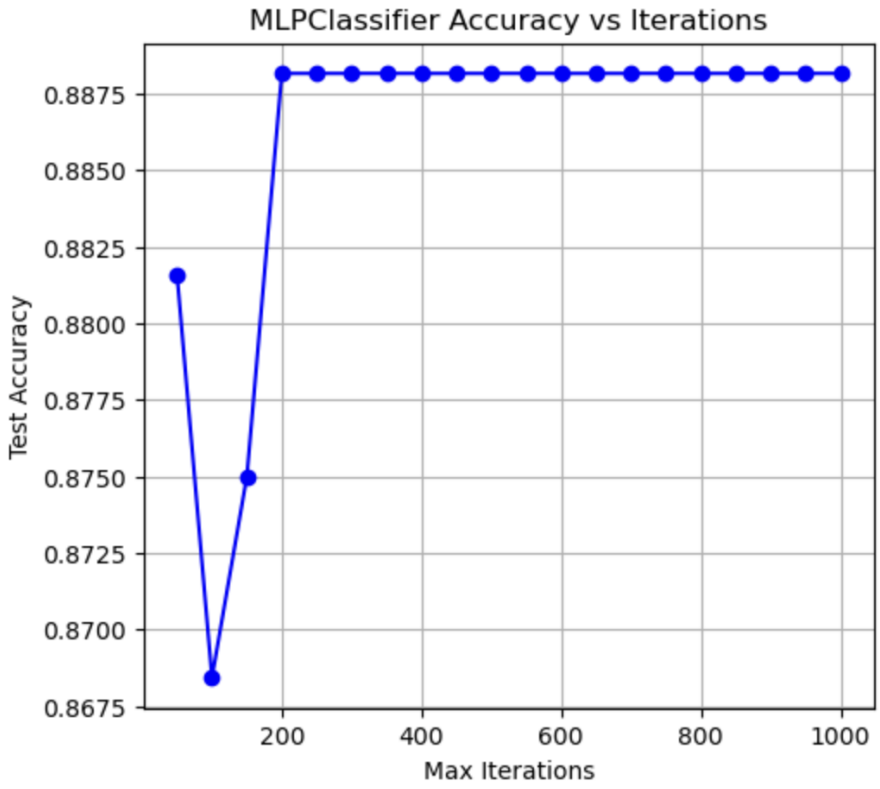


Figure 19: MLPClassifier

We used MLPClassifier from sklearn with:

- A single hidden layer of 25 neurons (k=25)
- Iterations tested from 50 to 1000, increasing by 50
- Data was scaled using StandardScaler.

This model achieved best accuracy of 88.82% at 200 iterations. It shows how simpler models can perform well.

c) Tracking Loss Value

We used clf.loss_curve_ to track the training loss. The loss curve for the best iteration (200) showed that the training loss steadily decreased with each iteration, indicating that the model was learning. But something we noticed is that loss and error don't always decrease in sync. Loss is a function of probabilities and model confidence, while accuracy counts correct predictions. We may observe small changes in the loss even as accuracy improves. It's because loss measures error in prediction probabilities while accuracy measures whether the predicted class is correct.

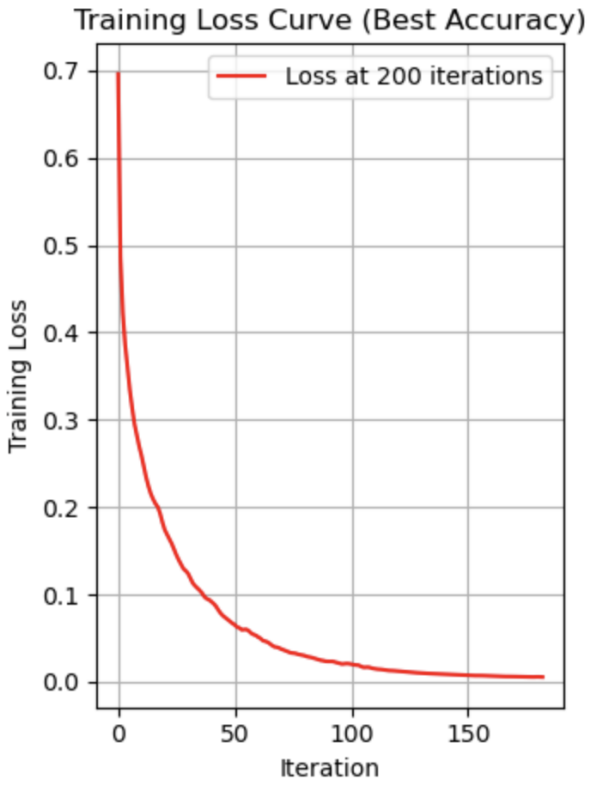


Figure 20:Loss Curve

d) Experimenting with Two Hidden Layers

In this part, we tested how splitting 25 neurons across two hidden layers affects the performance of the MLP model. Each combination was tested using the best iteration count (200) and accuracy was recorded. We used 10-fold cross validation to evaluate each combination.

The best accuracies are from:

- (25,0)

- (22,3)
- (18.7)
- (17,8)
- (9,16)

Some combinations like (21,4) performed terribly which is only 25% accuracy. Overall adding a second hidden layer did not improve the performance beyond single layer setup.

```
Part d – Classification Accuracy for Two Hidden Layers:
   Neuron Combination  Accuracy
0             (25, 0)    0.8882
1             (24, 1)    0.8355
2             (23, 2)    0.8487
3             (22, 3)    0.8882
4             (21, 4)    0.2500
5             (20, 5)    0.8684
6             (19, 6)    0.8816
7             (18, 7)    0.8882
8             (17, 8)    0.8882
9             (16, 9)    0.8750
10           (15, 10)    0.8816
11           (14, 11)    0.8816
12           (13, 12)    0.8618
13           (12, 13)    0.8750
14           (11, 14)    0.8684
15           (10, 15)    0.8618
16            (9, 16)    0.8882
17            (8, 17)    0.8816
18            (7, 18)    0.8684
19            (6, 19)    0.8684
20            (5, 20)    0.8618
21            (4, 21)    0.8421
22            (3, 22)    0.8289
23            (2, 23)    0.7763
24            (1, 24)    0.7961
```

Figure 21: Classification Accuracy

e) Explaining Accuracy Variation

Highest accuracy was achieved in multiple configurations like

- (25,0)
- (22,3), (18,7), (17,8) and (9,16)

Very uneven splits like (21,4), (2,23) performed poorly only had 25% and 77.63% Accuracy is varied across different neuron splits because of how feature abstraction is handled.

1. Too few neurons in the first layer might not capture enough patterns. In some combinations, the second hidden layer had only 1 or 2 neurons. That's not enough to learn meaningful patterns.
2. Too many in the second layer might not generalize well.
3. When the neurons are distributed in a more balanced way across the two layers, the model can learn better as both layers are contributing.

4. Overfitting vs Underfitting: deeper networks overfit if not enough data is available while shallow ones underfit.

The model preforms best when neurons are well-balanced. The accuracy depends a lot on how neurons are split across the layers.

f) Comparing MLP Classifier Performance

MLPClassifier: Achieved the highest accuracy at 88.82% using just a single hidden layer with 25 neurons and max iterations at 200. It handled the non-linear nature of the data well.

KNN had more variation depending on the k value. Achieved 84% accuracy with variability across different k values.

Logistic Regression: Achieved 86% accuracy, strong for a linear model but limited by non-linear patterns.

Naïve Bayes: Achieved 82% accuracy, the lowest accuracy overall. It's Because Naïve bayes relies on the assumption that all features are independent of each other.

Best Model Choice

The best model overall was the MLP classifier with (25,0) neurons and max iterations at 200. It achieved the highest accuracy of 88.82% which is about:

- 2.82% higher than logistic Regression
- 10.82% higher than KNN

This model handled the complexity of 754 features better than KNN and Logistic Regression. The MLP was stable throughout and converged by 200 iterations.

MLP is the best option for this dataset, it gave the best accuracy and adapted well to the complex and high dimensional data.

Appendix

```
Series([], dtype: int64)
Number of missing values:
id                          0
gender                      0
PPE                         0
DFA                         0
RPDE                        0
                           ..
tqwt_kurtosisValue_dec_33   0
tqwt_kurtosisValue_dec_34   0
tqwt_kurtosisValue_dec_35   0
tqwt_kurtosisValue_dec_36   0
class                       0
Length: 755, dtype: int64
```

As you can see there are no missing values in this dataset.