CS EE World
https://cseeworld.wixsite.com/home
May 2023
25/34
B

Submitter info:
https://docs.google.com/document/d/1YHvPDOzqVQ2hCGN2epyAEzCuTa-b4plV88EUglv426w/edit

Machine Learning for Parkinson's Disease diagnosis

How does "k-nearest neighbour algorithm" compare to "Naïve Bayes algorithm" in diagnosing Parkinson's Disease, when using striatum dimensional features as input data?

Computer Science

Word Count: 3997

Contents

1. Introduction

One of the worst symptoms of aging are brain related disorders. Although I have never met my great-grandmother, I've heard stories about her dementia, primary memory loss in her final years. It transformed her into a completely different person, someone who wouldn't even recognise her relatives. This showed me how important is our brain health and how any problems with it could be life changing.

In search for more answers about the brain, I visited the Champalimaud Foundation in Lisbon, more specifically the Nuclear Medicine department. They were working on detecting brain related disorders such as Parkinson's disease, Alzheimer's disease, etc. Most interestingly, they showed the possibility of implementing machine learning for detecting those specific diseases. Through my great-grandmother's story and my visit to Champalimaud, I decided to write an investigation into detection of brain related disorders with the help of machine learning.

In the department I was lucky to be shared with a refined dataset which contained dimensional features of striatum for patients with and without Parkinson's Disease (PD). As they explained, there is strong correlation between these features and whether a patient has PD, therefore making it suitable for a machine learning model. This dataset was used for conducting the experiment in this essay.

According to National Institute on Aging [1], PD is primarily developed in people older than 60 years old. PD main symptoms include unintended and uncontrolled movements like shaking. Dopamine transporter (DaT) loss in the brain is a key feature of PD which results in the symptoms [2]. A scan completed with a combination of SPECT and DaTSCAN scanners is a common way to evaluate DaT levels in the brain [3].

Figure 1 shows an example of such scan: the left scan shows a healthy subject, the right scan shows a PD patient. The bright yellow-red-blue regions represent the healthy

cells containing DaT. As one can see the PD patient has a clear decrease in healthy cells with DaT. Those regions also represent the size of striatum - region of the brain which controls the movement, as such in PD patients the striatum dimensions become smaller. This explains the correlation of striatum dimensional features to PD diagnosis (the dataset).

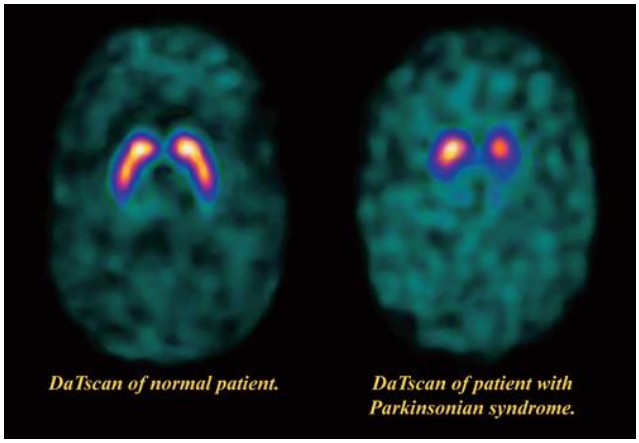

Figure 1 - DatSCAN for normal vs PD patients [3]

Visual examination of the dimensions of the striatum is not new, it is frequently used for the final diagnosis of possible PD patients. As seen in Figure 2 - width, length and thickness of the striatum can be extracted from a 3D scan. However, for medical staff, it can be time consuming and in some certain cases be hard to give an objective decision on whether the striatum dimensions are abnormal. Different quantification methods to help medical staff have been developed for more objective assessments, including machine learning.

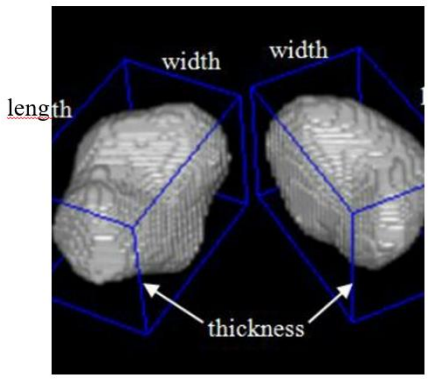

Figure 2 - Width, Length and Thickness of segmented Striatum

Use of machine learning increases the accuracy of automated diagnosis. Machine learning algorithms can consider many features at the same time making them multidimensional, which helps achieve high accuracy. An accurate machine learning model helps detect dopamine transporter loss early on and, therefore, assist a clinical decision for the diagnosis of PD. Spotting the disease early is important, because treatments such as levodopa/carbidopa will be more effective [4].

This work aims to compare two machine learning-based algorithms: k-nearest neighbour (k-NN) and Naïve Bayes (NB). More specifically, *"How does "k-nearest neighbour algorithm" compare to "Naïve Bayes" algorithm in diagnosing Parkinson's Disease, when using striatum dimensional features as input data?"*. These algorithms were chosen due to their simplicity and quick implementability, as such they require little computational power allowing me to use my personal computer for the experiment. The algorithms are relatively basic, the experiment would demonstrate whether there is potential using these specific algorithms for PD diagnosis, and if so, which algorithm out of the two is the better one. Three features related to the dimensions of the striatum were considered: length, width, and thickness. The algorithms were trained and tested using 10-fold cross validation, the results were stored in a confusion matrix, and then were used to calculate various metrics to evaluate and compare the models in more detail. All human data studies in this work have been performed in accordance with the ethical standards laid out by IB.

2. Theoretical Background

A. Machine Learning

Machine learning is a branch of computer science and artificial intelligence (AI) which focuses on imitating the way humans learn, that way gradually improving accuracy over time. This process of learning is also referred to as training the algorithm. To create a machine learning model, a combination of data and algorithms is used [5]. By “data” I refer to inputs that the algorithms process to achieve “output”. Different “algorithms” differ in the way they process the “data”, both in training and testing. The terms of “algorithm” and “model” will be used interchangeably in this work. The final “output” depends on whether the algorithm used is a supervised or unsupervised learner.

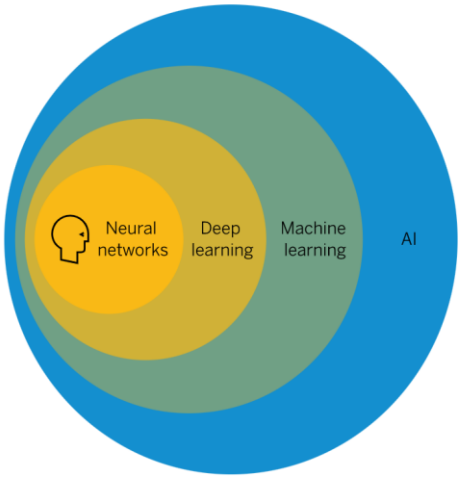

Figure 3 - Machine Learning

B. Training and testing

Training is an important procedure in machine learning, the algorithm in the model adapts in such a way that it can perform some certain tasks as successfully as possible. Usually, a model performs one kind of task, for example in this work: diagnosing a subject.

After the model is “trained”, it is “tested” to see how well it performs. That is done by giving the trained model data it has not previously seen, for example if the models in this work are

trained on subjects 1-400, we could test them on subject 401 and see whether they correctly classify the subject.

C. Machine learning categories

Machine learning algorithms are split into two categories based on their training method: supervised and unsupervised learning. The output of a supervised learning model is a prediction based on the input data, for example if an email is a spam or not a spam. The "input data" in such model could be the features of the email: number of words, types of words, etc. However, to train a supervised algorithm it requires experts that can "label" the data properly during the training stage [6]. A "label" is a correct tag to the data, using the email example, the "label" is a tag that classifies the email as spam or not spam.

On the other hand, unsupervised algorithms are used to get some new insights from large amounts of input data. As such often, there is no specified output for unsupervised learning algorithms.

The data I am using is already properly labelled by experts, which means every subject already has a label stating whether he/she has PD. The model needs to predict whether a subject has PD. As such for the purposes of this essay, supervised machine learning will be used, which is explained in more detail below.

D. Supervised machine learning

What defines supervised learning is its use of labelled datasets to train the model. The model is trained to do a certain task such as identify a disease. According to javatpoint.com *"The aim of a supervised learning algorithm is to find a mapping function to map the input variable (x) with the output variable (y)."* [7]

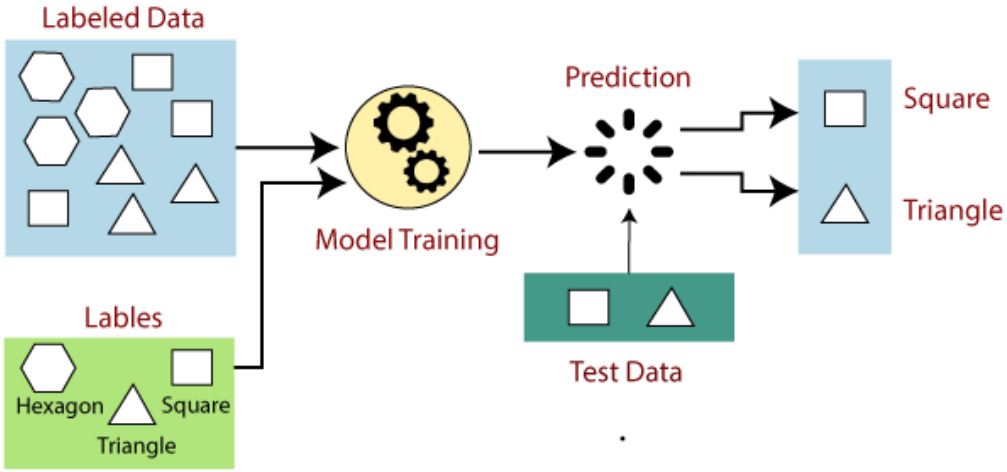

Figure 4 - Working of a Supervised Learning Algorithm [7]

Looking at figure 4, this supervised learning algorithm is trained to identify the 3 types of shapes: Hexagon, Triangle and Square. When training, the model receives the data x (the shape image) and the label y ("square" / "triangle" / "hexagon"). The model will look for patterns to be able to classify each shape. For instance, the "square" has 4 equal sides, the "triangle" has 3 sides, and so on. After the training process is complete, we can test the model with test data (similar but previously unseen) and find how well it performs.

Supervised learning can be further split into two subcategories: Regression and Classification.

Regression algorithms are used to find relationship between dependent and independent values. For example, it could be used to make projections such as sales revenue for a given business. As such, it is the task of producing a continuous quantity [8]. Common examples of regression algorithms are linear regression and polynomial regression [9].

Classification algorithms accurately assign data to specific categories. The previously mentioned 'shape identifier' model would be a good example; it puts each shape into a specific category. Classification is the task of predicting a discrete class label [8]. Support Vector Machine (SVM), k-nearest neighbour (K-NN), random forest are popular classification algorithms.

For the purposes of this work, supervised classification algorithms are the best choice since we want to classify the subjects into two categories: positive for PD or negative for PD. I will refer to the classification algorithms as: classifiers and algorithm interchangeably.

E. K-Nearest Neighbour Algorithm

The k-NN algorithm is a non-parametric (doesn't make assumptions about underlying data), supervised learning classifier that uses proximity to make classifications about the grouping of data points. It is also a lazy learner algorithm, which means it doesn't directly learn from the training data, instead it stores it, and at the time of classification it uses it to compare it to new data.

Imagine a model is built to identify dogs and cats, and the only two variables we have are: length of ears (X) and sharpness of claws (Y). Figure 5 below shows what this would look like.

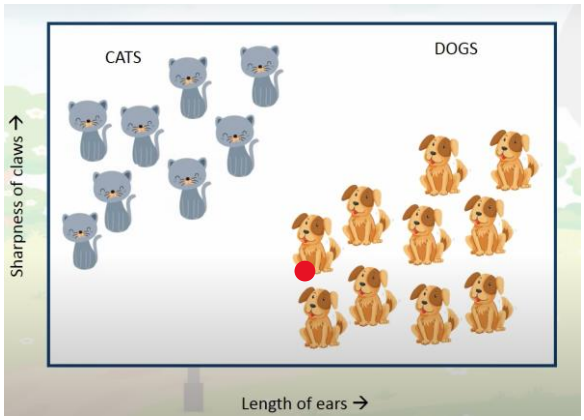

Figure 5 - Dog and Cat classifier algorithm [10]

As you can see the "Cat" class has sharper claws and shorter ears. Whereas the "Dog" class is longer eared, but the claws are less sharp. This is essentially a k-NN model after the "training" is complete. The input data is plotted, and the model also labels each data point as a dog or a cat. Next, imagine we have a query point (red dot) which we want to classify as a dog or a cat, based on these two features. Because the data point has more dog neighbours, it will be classified as a dog. This concept is also visualised in Figure 6 below.

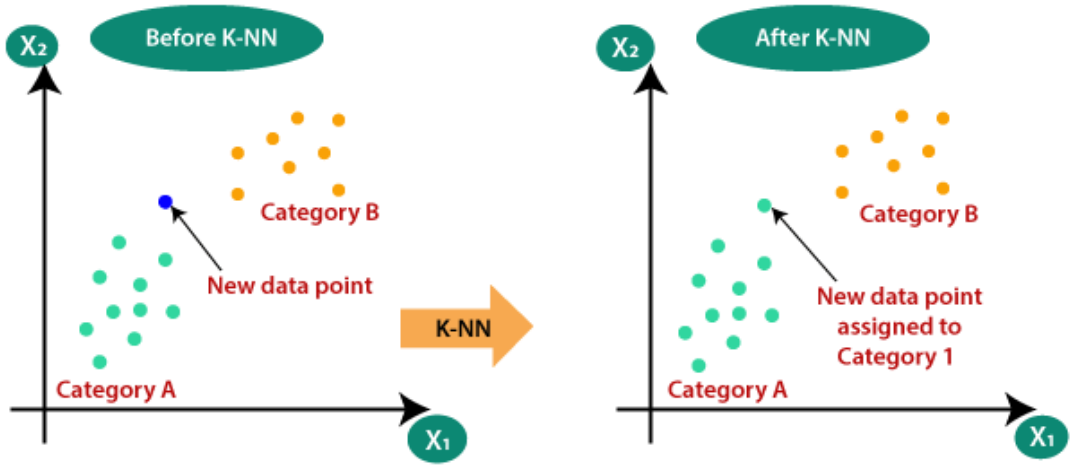

Figure 6 – k-NN clustering [11]

To recap, the goal of a k-NN algorithm is to identify the nearest neighbours of a query point, so that to assign it to the nearest class. To do that the algorithm has two requirements: choosing the k-value and choosing a distance metric. The k-value specifies the number of neighbours that will be checked to give a classification to the query point [12]. Figure 7 demonstrates the importance of the k-value, when the k-value is set. An imaginary circle can be visualised that captures k nearest neighbours. When k=3, there is two Class B neighbours and one Class A, hence the query point will be labelled as Class B as there is a Class B majority. But if k=7, the majority is Class A, hence the query point will be labelled as Class A.

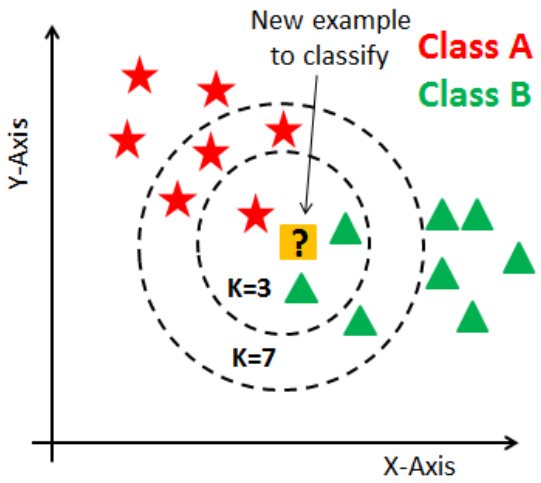

Figure 7 - Example of k-NN classification [10]

This demonstrates how choosing the value of k can be an act of balancing, as different values may lead to different classification. The choice of the best k-value can largely depend upon the size of the inputs. The value of k is recommended to be a whole odd number, so that to avoid ties.

To classify the query points to a certain class, the distance between the query point and other data points needs to be calculated. The distance measured helps to identify the neighbours which in turn help classify the query points.

There are many ways of measuring the distance between points, for the purposes of this work, Euclidian distance will be used since it is the most used distance metric. Using the formula below a straight line between the query point and the other point is measured.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

F. Naïve Bayes

Naive Bayes algorithms are a set of supervised learning algorithms based on applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable. [13]

Consider a dataset that describes the conditions to play golf (Figure 8). Where the output is "Yes" or "No" for playing golf. The deciding features or X variables for playing golf are "Outlook", "Temperature", "Humidity" and "Windy".

	Outlook	Temperature	Humidity	Windy	Play Golf
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes

Figure 8 - Fictional golf dataset

The fundamental Naïve Bayes assumption is that each X variable makes an independent and equal contribution to the output. In relation to our dataset this can be understood as no X variable is dependent on the other. For example, "Hot" temperature has nothing to do with the humidity. Secondly, since all features contribute equally, knowing only outlook and temperature alone can't give accurate prediction. Even though these assumptions are generally not correct in real life situations, the algorithm often works well in practice.

The specific algorithm used in this experiment was the Gaussian NB classifier, in which the likelihood of the features is assumed to be Gaussian, hence, the conditional probability is given by:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Figure 9 - Gaussian conditional probability [14]

3. Evaluating machine learning algorithms

A. Confusion matrix

A detailed evaluation technique used for ML algorithms is a confusion matrix. It is a table which helps get insight into the type of errors the model is making and allows to calculate other more specific metrics.

As seen in Figure 10 below the matrix has two axis “predicted” (horizontal axis) and “actual” (vertical axis). 0 stands for HC and 1 for PD subject. True Negative (TN) holds number of correctly predicted negatives. True Positive (TP) holds number correctly predicted positives. False Negative (FN) holds incorrectly predicted negatives. And False Positive (FP) holds incorrectly predicted positives. Generally, you want to minimise both FP and FN. However, in some scenarios minimising one over another is more important. For example, a possible metal detector would want to have no False Negatives, since not detecting a gun may cost lives of many. On the contrary a spam detector would want to decrease False Positives, since it would be very annoying for the user to have to search an important email in spam.

For my scenario it would be best to have a low number of False Negatives, since like stated earlier, if the disease is spotted early on, medication can be administered to decrease the total damage of the disease. However, having a low number of false

positives is also important, since it removes the possible cost of administering medication which is not required.

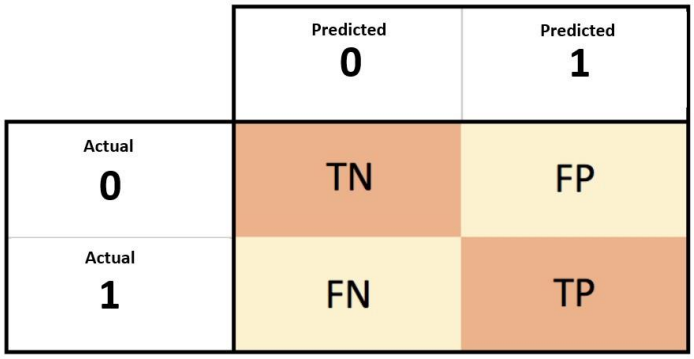

Figure 10 - Confusion Matrix example

B. Evaluation metrics

Firstly, the most basic evaluation metric is the classification accuracy. As the name suggests it is just a fraction of right predictions out of total number of predictions. And is defined by simple formula below.

$$\text{classification accuracy} = \frac{\text{correct predictions}}{\text{total predictions}}$$

However, this metric is very basic and doesn't tell us much information about what errors the model is making.

Sensitivity is the probability of testing positive for diseased patients. It will be used to determine whether the models are sufficiently sensitive to pick up the disease.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Specificity refers to probability of testing negative for non-diseased patients i.e., it represents the proportion of patients without disease who have negative test result.

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

Finally, the Mathews Correlation Coefficient will be included. Some might argue that the F1 score should be included since it is one of the most used metrics used to evaluate classification models. However, research shows it is not as accurate as MCC and will not be included in this work [15].

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}$$

In the MCC formula we can see a balanced consideration of all boxes of the confusion matrix, unlike sensitivity or specificity which consider only two boxes.

C. K-fold cross validation

Finally, the models will be evaluated on their ability to generalise – ensuring that the models perform well with different training data. This will be done by performing k-fold cross validation, more specifically 10-fold cross validation which is explained below.

First the dataset is randomly shuffled to reduce bias, and then is split into 10 folds like seen in Figure 11.

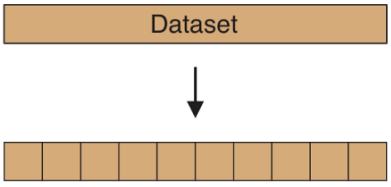

Figure 11 - 10-fold cross-validation

Initially, 9 folds are used to train the models and 1 to test the models. The predictions are obtained from the models produced. Then, the procedure is repeated until all folds have been used for testing (Figure 12).

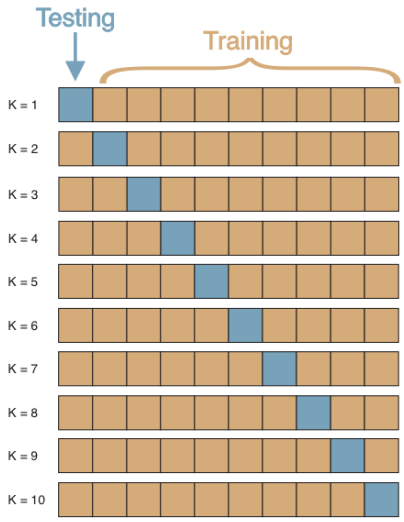

Figure 12 - 10-fold cross validation

4. Hypothesis

I hypothesise that the k-NN algorithm will perform the best. I base the hypothesis primarily due the Naïve Bayes' assumption of independence between all the features which in this case is not true. The dimensional features of striatum must be closely related to each other. For example, as width decreases, thickness and length may also decrease, this is because the striatum does not decrease in size one dimensionally but instead three dimensionally.

5. Methodology

A. Dataset

The dataset (Figure 13) used in the experiment was obtained with the help of Olivera et al. [16]. Who in turn extracted all the features from images obtained from a Parkinson's Progression Markers Initiative database [17]. The dataset contains 652 subjects, for the groups: control female (73), control male (136), PD female (157) and PD male (286). Overall, the healthy control (HC) subjects' age was 61.8 ± 11.3 years old, and the PD subjects' age was 61.7 ± 9.7 years old.

Each row holds the data for a different subject. The Y values are in the first column of the figure 13, it stores the real diagnosis of the subject, where 0 is for HC and 1 is for PD. The X values in columns 2-4 store the dimensional features of the striatum for each subject. They are the Width, Length, and the Thickness of the striatum, same as in figure 2.

1	Diagnosis	Width	Length	Thickness
2	0	23.98	39.16	27.38
3	0	28.69	34.83	28.27
4	0	23.23	36.40	24.73
5	0	23.17	35.96	29.15
6	0	27.93	35.24	28.27
7	0	23.14	35.61	25.62
8	0	19.25	31.60	22.08
9	0	30.22	33.42	29.15
10	0	20.82	29.63	23.85
11	0	30.16	38.12	28.27
12	0	19.97	33.17	22.08
13	0	23.01	33.29	24.73
14	0	18.59	26.52	23.85
15	0	25.46	33.64	27.38
16	0	20.78	30.88	22.08
17	0	22.38	33.26	23.85
18	0	27.09	33.70	25.62
19	0	25.58	36.93	22.08
20	0	23.04	33.26	26.50
21	0	26.99	34.52	25.62

Figure 13 – Snapshot of Dataset

B. Experimental Procedure

1. Use Python to extract the X and Y values from the dataset.
2. Experiment with different values of k to find the one that gives the best accuracy.
3. Create the k-NN and NB models using the sklearn library.
4. Perform 10-fold cross-validation on each model and store all the outputs of each model in two separate confusion matrices.
5. Store the metrics of accuracy of each fold in both models in an array.
6. Find the average value of accuracy, specificity, sensitivity and MCC for each model.
7. Show all the metrics in tables for easier visual comparison. The percentages range from 0 to 100%. While MCC ranges from -1, to +1, with extreme values of -1 and +1 reached in case of perfect misclassification or perfect classification.

6. Results and Analysis

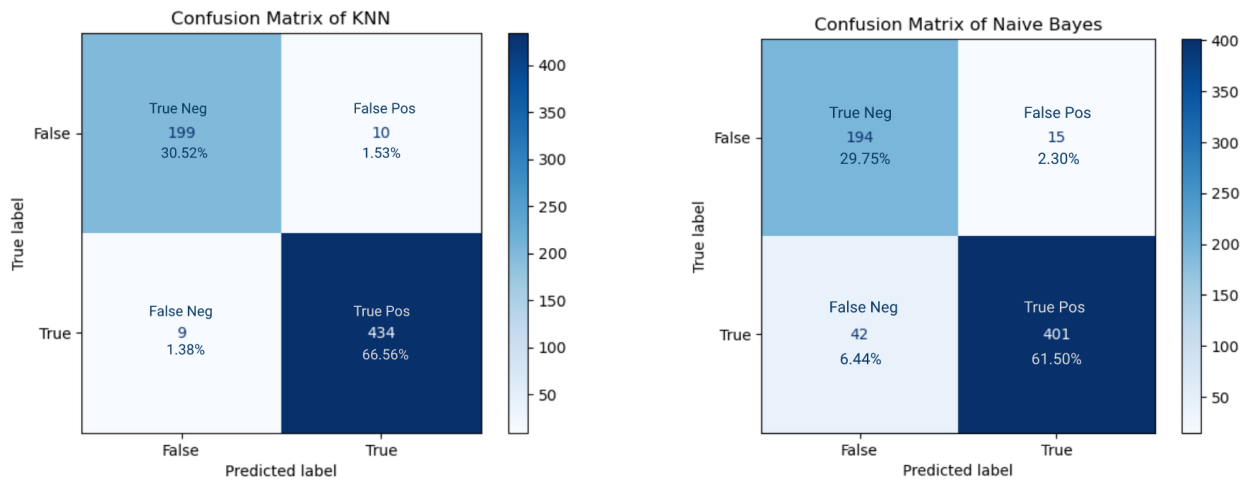

Figure 14 - k-NN and Naïve Bayes confusion matrices

To begin with, the confusion matrices in figure 14 provide us with the most direct illustration of the models' performances by indicating the number of true and false prediction in each class. I will be referring to positive as a subject with PD and vice versa. The left confusion matrix has outputs from all 10 folds for k-NN, so does the right but for NB.

Both models have a very high number of True Positives and False Negatives. k-NN has 66.56% of true positives and 30.52% true negatives, and if summed we get the accuracy of 97.08%. This is a high score; it shows how most patients were predicted/diagnosed correctly. Similarly, the Naïve Bayes also has a high number of true positives being 61.50% and true negatives being 29.75%, with accuracy of 91.25%. But overall, Naïve Bayes performed slightly worse, given that its true positives value is less by 5.06% compared to k-NN. This is because it classified lots of false negatives (6.44%), and this is bad as the goal of testing is to classify the disease and give medication as early as possible to the patients.

Looking at the accuracy scores for each fold in Table 1 we can see how most folds of k-NN were much more accurate than those of NB. In fact, in the first and ninth folds of k-NN were able to achieve 100% accuracy. The accuracy of k-NN ranges from 95%-100% therefore demonstrating its excellent generalisation ability. NB on the other hand performed

considerably worse in terms of generalisation, even though the highest accuracy was 96.9% the lowest was 84.6%. This shows how NB can't perform as well on previously unseen data as k-NN.

Fold	**k-NN accuracy (%)**	**Naïve Bayes accuracy (%)**
1	100.0	96.9
2	95.5	84.8
3	96.9	90.8
4	96.9	95.5
5	96.9	90.8
6	95.4	89.2
7	95.4	89.2
8	96.9	84.6
9	100.0	95.4
10	96.9	95.4
Average accuracy	97.1	91.3

Table 1 - Accuracy for each fold and the average

Table 2 has the summary of main metrics evaluated. Firstly, k-NN has an average accuracy of 97%. The average sensitivity value of 98% demonstrates how k-NN is very successful at identifying sick patients and misses out a very small number. The average specificity is slightly lower being at 95.2% shows how the model is slightly worse at identifying healthy patients, which could although not as bad as not spotting sick patients can still be problematic. The achieved MCC of k-NN is 0.933.

Naïve Bayes on the other hand had an average accuracy of 91%. The specificity being at 90.5% is considerably worse than k-NN's. Interestingly, Naïve Bayes was more successful at identifying healthy patients than sick, with specificity at 92.8%. Finally, NB achieved MCC of 0.809.

	k-NN	Naïve Bayes
Average Accuracy (%)	97	91
Average Sensitivity (%)	98.0	90.5
Average Specificity (%)	95.2	92.8
Average MCC	0.933	0.809

Table 2 - Average of metrics

Overall, it is fair to say that both algorithms achieved relatively high scores in terms of predicting PD in patients. However, k-NN was by far the better classifier, outscoring NB in all the metrics considered in this experiment.

7. Evaluation of experiment

This experiment had strong positive aspects of it. Most importantly the data used for training the algorithms was properly labelled by experts which enabled the possibility of using supervised learning in this experiment. Additionally, the x values used in experiment (striatum dimensions), are commonly used by medical staff to give clinical diagnose. As such, the data used was already previously highly relevant for the diagnose, and this is confirmed by very high scores.

However, the experiment had limitations. Firstly, there was uneven distribution of male and female as well as of PD and HC subjects. As seen in Figure 15, almost three quarters of patients were male.

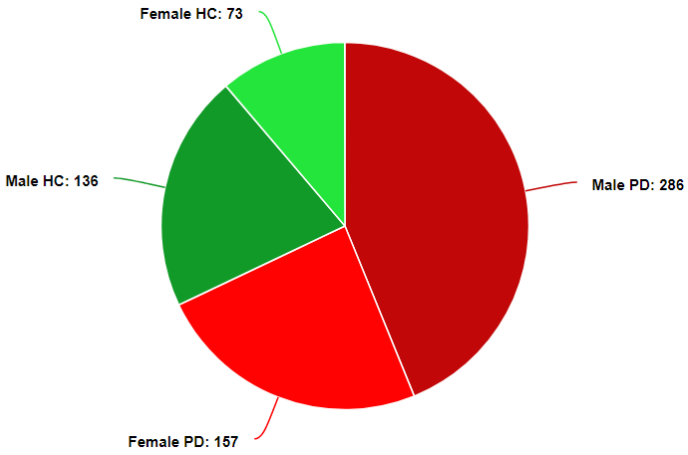

Figure 15 - Pie Chart representing Males and Females in the database

Same can be said for the distribution of healthy controls and sick patients. There are 443 PD patients and only 209 HC. For possible improvements it would be beneficial to also look at how the accuracies differed when taking the dimensional based features individually and not together.

Finally, the accuracies of male and female subjects were not compared separately. In future it would be interesting to see whether male and female subjects had any notable differences in the classification accuracy.

As such for improvements a dataset with the same number of PD and HCs should be used, and perhaps the male and female subjects should be compared separately.

8. Conclusion

In conclusion, the combination of supervised machine learning algorithms and striatum dimensional features undoubtedly performed positively. Though there are some inaccuracies present in the algorithms, overall, the experiment shows how these algorithms can be used for assisting the clinical decision of diagnosing Parkinson's disease. In terms of comparing Naïve Bayes and k-NN, it can be safely said that k-NN is the better algorithm, which was confirmed by higher classification accuracy and all the other metrics used. As

such, k-NN shows strong potential to be used in a real-life scenario of diagnosing Parkinson's Disease.

9. Further research

Whilst this essay demonstrated that k-NN is the better algorithm for identifying Parkinson's disease when using striatum dimensional features as input, it leaves many more possible questions to be answered. It would be interesting to see how other supervised machine learning algorithms like neural networks or random forest would perform on the same task. This would help identify which algorithm out of the supervised learning family has the most potential. If possible, it would be interesting to compare how the accuracy changes if instead of the dimension values, a real scan image of the striatum is used as input, such as in Figure 1. In addition, it would also be interesting using an unsupervised learning algorithm, and see whether it can spot patterns in this data that a human might not.

Parkinson's disease is known to be more present in males than females [18]. It would be interesting to see if there are any possible correlations between the gender and the degeneration of striatum. Perhaps there could be found a relationship between the dimensional features of the striatum and the gender of the patient with Parkinson's disease. Whether such relationship exists or not can also be investigated using supervised machine learning algorithms.

10. Bibliography

[1] "Parkinsons disease," National Institute of Health, [Online]. Available: https://www.nia.nih.gov/health/parkinsons-disease. [Accessed 20 October 2022].

[2] "Datscan," [Online]. Available: https://parkinsonsnewstoday.com/parkinsons-disease-tests-diagnosis/datscan/. [Accessed 19 October 2022].

[3] "Image: DaTSCAN, normal vs abnormal," [Online]. Available: https://www.cedars-sinai.org/programs/imaging-center/exams/nuclear-medicine/datscan/information.html.

[4] "Parkinson's Disease: Causes, Symptoms, and Treatments," National Institute on Health, [Online]. Available: https://www.nia.nih.gov/health/parkinsons-disease. [Accessed 12 September 2022].

[5] IBM, "Machine Learning," International Business Machines Corporation, [Online]. Available: https://www.ibm.com/cloud/learn/machine-learning. [Accessed 11 August 2022].

[6] IBM, "Supervised vs. Unsupervised Learning: What's the Difference?," International Business Machines Corporation, [Online]. Available: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning. [Accessed 12 August 2022].

[7] javatpoint, "Supervised machine learning - javatpoint," [Online]. Available: https://www.javatpoint.com/supervised-machine-learning. [Accessed 11 August 2022].

[8] "Cassification versus regression in machine learning," [Online]. Available: https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/. [Accessed 11 September 2022].

[9] IBM, "What is supervised learning?," International Business Machines Corporation, [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning. [Accessed 11 August 2022].

[10] Simplilearn, "KNN Algorithm In Machine Learning," [Online]. Available: https://www.youtube.com/watch?v=4HKqjENq9OU. [Accessed 11 September 2022].

[11] javatpoint, "k-nearest-neighbor algorithm for machine learning," [Online]. Available: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning. [Accessed 11 September 2022].

[12] IBM, "K-Nearest Neighbors Algorithm," International Business Machines Corporation, [Online]. Available: https://www.ibm.com/topics/knn. [Accessed 11 September 2022].

[13] "Naive Bayes Classifiers," [Online]. Available: https://www.geeksforgeeks.org/naive-bayes-classifiers/. [Accessed 14 November 2022].

[14] "Naive Bayes, Scikitlearn," [Online]. Available: https://scikit-learn.org/stable/modules/naive_bayes.html. [Accessed 21 November 2022].

[15] D. Chicco and G. Jurman, "The advantages of the MCC over F1 score and accuracy in binary classification evaluation," [Online]. Available: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-019-6413-7. [Accessed 27 November 2022].

[16] F. P. Oliveira, M. Castelo-Branco, D. B. Faria and D. C. Costa, "“Extraction, selection and comparison of features for an effective automated computer-aided diagnosis of parkinson’s disease based on [123i]fp-CIT SPECT images,”," *European Journal of Nuclear Medicine and Molecular Imaging,* vol. 45, no. 6, 2017.

[17] [Online]. Available: www.ppmiinfo.org/data. [Accessed 13 November 2022].

[18] "Parkinsons in men vs women," [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6700650/. [Accessed 20 October 2022].

[19] U. o. York, "What is Machine Learning," 06 September 2021. [Online]. Available: https://online.york.ac.uk/what-is-machine-learning/. [Accessed 11 August 2022].

[20] "Laws of Proximity and Similarity," [Online]. Available: https://isle.hanover.edu/Ch05Object/Ch05ProxSim_evt.html. [Accessed 11 September 2022].

[21] "k optimal value," [Online]. Available: https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb#:~:text=The%20optimal%20K%20value%20usually,be%20aware%20of%20the%20outliers.. [Accessed 11 September 2022].

[22] "Early symptoms signs of PD," [Online]. Available: https://parkinsonsdisease.net/diagnosis/early-symptoms-signs.

11. Appendix

Code Used

```
#importing necessary libraries and configurations
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import ConfusionMatrixDisplay
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
import sklearn.metrics

#reading the Parkinson's patients database and storing according X and y value
data = pd.read_csv('C:/Users/vss19/EE_Parkinsons/Datset_ParkinsonVsControl_PPMI.csv')
X = data.iloc[:, 8:11]
y = data.iloc[:, 0]
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, test_size=0.2)
```

```
#creating, training and testing k-NN and Naive Bayes
classifier_KNN = KNeighborsClassifier(n_neighbors=11, p=2, metric='euclidean')
classifier_NB = GaussianNB()
classifier_NB.fit(X_train, y_train)
classifier_KNN.fit(X_train, y_train)

#performing 10 fold cross-validation and printing accuracy score from each fold and their
mean
scores = cross_val_score(classifier_KNN, X, y, cv=10)
print(scores)
print("%0.2f accuracy of KNN with a standard deviation of %0.2f" % (scores.mean(),
scores.std()))
print()
scores1 = cross_val_score(classifier_NB, X, y, cv=10)
print(scores1)
print("%0.2f accuracy of NB with a standard deviation of %0.2f" % (scores1.mean(),
scores1.std()))
print()

#Plotting all of the results in a single confusion matrix for k-NN
y_pred_KNN = cross_val_predict(classifier_KNN, X, y, cv=10)
cm_KNN_CV = confusion_matrix(y, y_pred_KNN)
cm_display_KNN_CV = ConfusionMatrixDisplay(confusion_matrix = cm_KNN_CV, display_labels =
[False, True])
cm_display_KNN_CV.plot(cmap = plt.cm.Blues)
plt.title("Confusion Matrix of KNN")
plt.show()

#Plotting all of the results in a single confusion matrix for Naive Bayes
y_pred_NB = cross_val_predict(classifier_NB, X, y, cv=10)
cm_NB_CV = confusion_matrix(y, y_pred_NB)
cm_display_NB_CV = ConfusionMatrixDisplay(confusion_matrix = cm_NB_CV, display_labels =
[False, True])
cm_display_NB_CV.plot(cmap = plt.cm.Blues)
plt.title("Confusion Matrix of Naive Bayes")
plt.show()
```

Dataset Used

Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness
0	23.98	39.16	27.38	0	29.56	39.31	30.03	0	26.11	42.98	31.80	0	17.65	30.03	23.85	0	14.58	30.41	19.43	0	24.77	37.18	29.15	1	8.28	8.31	7.07	1	19.25	20.10	23.85
0	28.69	34.83	28.27	0	21.63	33.64	29.15	0	17.59	26.11	22.97	0	17.02	21.38	17.67	0	27.90	38.53	27.38	0	22.38	41.88	30.03	1	24.83	25.46	16.78	1	13.07	11.79	19.43
0	23.23	36.40	24.73	0	26.30	36.05	24.73	0	21.63	35.52	29.15	0	28.06	39.97	30.92	0	20.82	34.86	26.50	0	19.22	29.66	23.85	1	16.11	16.62	21.20	1	0.00	0.00	0.00
0	23.17	35.96	29.15	0	14.51	25.64	19.43	0	26.21	38.53	27.38	0	27.12	44.80	27.38	0	25.46	33.58	27.38	0	27.84	38.40	29.15	1	16.93	15.83	17.67	1	0.00	0.00	0.00
0	27.93	35.24	28.27	0	27.93	36.96	27.38	0	24.01	31.35	28.27	0	21.57	33.29	27.38	0	23.17	38.34	27.38	0	25.52	32.10	28.27	1	19.34	16.74	18.55	1	22.29	27.65	28.27
0	23.14	35.61	25.62	0	23.17	33.98	30.03	0	16.90	31.98	20.32	0	20.78	35.99	30.03	0	23.23	35.99	30.03	0	20.91	40.72	29.15	1	11.44	11.07	11.48	1	0.00	0.00	0.00
0	19.25	31.60	22.08	0	29.44	42.35	35.34	0	22.35	35.55	26.50	0	28.69	42.67	29.15	0	19.94	31.29	22.97	0	19.94	32.38	25.62	1	12.16	13.54	19.43	1	9.84	9.91	15.02
0	30.22	33.42	29.15	0	30.32	42.67	36.22	0	26.37	33.17	29.15	0	28.69	42.82	28.27	0	21.60	35.17	26.50	0	19.97	31.63	23.85	1	21.66	18.40	22.08	1	0.00	0.00	0.00
0	20.82	29.63	23.85	0	28.72	38.53	27.38	0	18.43	25.33	21.20	0	16.11	34.30	22.08	0	27.09	38.84	30.03	0	19.22	31.26	21.20	1	0.00	0.00	0.00	1	20.82	24.45	24.73
0	30.16	38.12	28.27	0	19.97	30.10	23.85	0	19.31	29.69	26.50	0	25.55	38.56	24.73	0	23.86	38.34	27.38	0	20.69	30.44	22.97	1	12.60	13.10	16.78	1	9.81	11.54	12.37
0	19.97	33.17	22.08	0	30.35	44.86	30.03	0	19.22	30.88	24.73	0	27.09	39.56	29.15	0	23.86	37.15	30.03	0	23.79	41.13	31.80	1	19.28	15.49	22.08	1	21.69	19.12	18.55
0	23.01	33.29	24.73	0	28.72	39.56	30.03	0	26.21	37.56	23.85	0	23.95	36.71	22.08	0	28.72	42.07	28.27	0	27.90	39.50	25.62	1	0.00	0.00	0.00	1	0.00	0.00	0.00
0	18.59	26.52	23.85	0	30.35	43.58	34.45	0	21.63	32.76	23.85	0	26.40	35.58	27.38	0	23.23	36.84	27.38	0	15.33	21.73	19.43	1	0.00	0.00	0.00	1	18.15	11.07	16.78
0	25.46	33.64	27.38	0	27.87	36.08	27.38	0	19.28	27.71	19.43	0	27.09	41.54	26.50	0	22.98	34.74	22.08	0	25.64	40.16	24.73	1	14.55	11.47	14.13	1	6.68	10.31	8.83
0	20.78	30.88	22.08	0	17.65	31.60	17.67	0	23.10	34.86	25.62	0	26.21	29.88	25.62	0	22.10	42.67	27.38	0	21.63	33.20	22.08	1	18.53	15.99	19.43	1	19.28	17.09	25.62
0	22.38	33.26	23.85	0	20.82	34.39	22.97	0	26.33	36.11	22.97	0	18.62	27.65	25.62	0	22.45	33.23	21.20	0	23.29	35.21	24.73	1	0.00	0.00	0.00	1	18.31	17.71	21.20
0	27.09	33.70	25.62	0	24.74	38.37	26.50	0	21.47	31.26	22.08	0	25.55	31.44	30.03	0	20.82	30.47	25.62	0	31.79	39.97	30.03	1	0.00	0.00	0.00	1	11.44	12.26	11.48
0	25.58	36.93	22.08	0	19.15	31.69	16.78	0	22.29	34.77	24.73	0	19.09	32.85	22.08	0	22.38	29.28	22.97	0	17.46	31.95	22.08	1	14.58	11.44	15.02	1	17.68	18.18	22.97
0	23.04	33.26	26.50	0	19.34	32.82	24.73	0	21.57	33.67	23.85	0	23.10	34.01	28.27	0	20.06	34.77	25.62	0	24.01	37.56	30.03	1	19.31	21.38	19.43	1	0.00	0.00	0.00
0	26.99	34.52	25.62	0	27.12	35.21	27.38	0	27.09	42.01	25.62	0	2.76	2.76	2.65	0	19.22	30.06	23.85	1	30.35	25.74	29.15	1	23.32	21.44	22.97	1	9.78	10.75	15.02
0	28.72	39.22	26.50	0	16.83	27.27	22.08	0	25.61	36.40	25.62	0	23.17	35.17	23.85	0	20.78	34.74	23.85	1	13.01	15.02	19.43	1	20.91	19.19	15.90	1	0.00	0.00	0.00
0	22.35	28.50	20.32	0	16.87	28.47	21.20	0	20.82	32.38	20.32	0	21.60	31.26	22.97	0	18.34	30.47	23.85	1	14.55	11.54	12.37	1	20.88	19.47	22.08	1	20.91	17.84	20.32
0	16.83	21.04	16.78	0	23.98	33.23	23.85	0	18.59	28.53	19.43	0	29.44	43.20	34.45	0	20.06	27.65	21.20	1	0.00	0.00	0.00	1	6.71	8.28	15.02	1	24.89	20.22	25.62
0	27.09	33.67	24.73	0	29.50	36.46	30.03	0	16.93	24.89	20.32	0	21.54	37.15	29.15	0	24.67	39.94	27.38	1	17.68	21.00	22.97	1	10.56	12.73	16.78	1	16.99	18.94	27.38
0	19.22	26.11	21.20	0	23.92	35.55	27.38	0	9.78	15.77	11.48	0	20.69	36.80	34.45	0	19.25	31.66	22.97	1	2.35	2.41	2.65	1	18.43	16.77	19.43	1	23.23	19.00	28.27
0	26.40	36.74	31.80	0	27.05	35.17	31.80	0	18.40	24.14	22.08	0	26.08	39.03	31.80	0	20.72	34.77	23.85	1	16.93	15.93	24.73	1	17.71	18.25	21.20	1	16.96	12.67	20.32
0	19.81	31.26	26.50	0	28.69	40.79	30.03	0	20.82	33.29	22.08	0	22.51	32.42	32.68	0	17.68	28.84	21.20	1	0.00	0.00	0.00	1	24.58	26.71	28.27	1	23.92	36.33	27.38
0	28.72	38.00	27.38	0	19.19	23.39	22.97	0	22.32	31.60	25.62	0	25.46	35.96	32.68	0	19.31	25.36	18.55	1	11.35	11.54	9.72	1	0.00	0.00	0.00	1	20.85	18.37	22.97
0	26.40	38.81	29.15	0	23.17	37.90	27.38	0	20.06	29.31	29.15	0	20.72	32.38	26.50	0	19.94	32.73	25.62	1	16.77	15.49	22.08	1	24.01	20.38	21.20	1	13.76	13.10	12.37
0	29.41	41.63	26.50	0	30.32	45.39	27.38	0	17.71	28.40	19.43	0	22.35	36.37	37.10	0	24.80	36.77	24.73	1	9.84	14.30	18.55	1	16.27	33.92	23.85	1	20.06	16.21	22.08
0	22.35	34.05	25.62	0	24.77	43.07	25.62	0	23.14	34.83	30.03	0	17.49	27.31	30.92	0	28.72	38.40	28.27	1	16.18	15.80	19.43	1	17.59	17.12	16.78	1	17.74	23.32	26.50
0	29.53	41.26	30.03	0	23.95	38.37	30.03	0	23.95	37.09	24.73	0	28.00	35.55	37.10	0	24.77	34.05	26.50	1	20.75	21.00	22.97	1	11.41	13.07	22.08	1	17.62	21.35	21.20
0	22.51	32.89	26.50	0	20.75	31.60	24.73	0	31.00	40.75	31.80	0	21.63	34.77	25.62	0	20.82	33.26	22.08	1	20.63	18.06	25.62	1	14.58	18.97	23.85	1	16.96	16.68	22.08
0	23.98	34.45	27.38	0	28.72	36.93	29.15	0	25.52	26.15	22.08	0	17.74	30.06	21.20	0	22.48	30.88	22.97	1	13.83	12.26	15.90	1	0.00	0.00	0.00	1	17.78	23.36	25.62
0	31.10	45.21	30.92	0	24.70	39.88	30.92	0	21.44	34.77	26.50	0	21.60	35.61	20.32	0	31.07	43.11	30.03	1	23.26	20.28	28.27	1	10.60	11.07	8.83	1	0.00	0.00	0.00
0	29.47	42.42	32.68	0	27.68	41.19	31.80	0	23.14	33.58	29.15	0	18.37	23.01	16.78	0	22.35	34.77	33.57	1	3.54	3.98	4.42	1	23.10	25.11	22.97	1	25.68	19.81	24.73
0	31.13	40.10	28.27	0	31.95	38.75	33.57	0	18.47	31.22	30.03	0	16.05	26.93	20.32	0	14.51	30.88	20.32	1	9.84	10.69	9.72	1	21.76	22.98	25.62	1	17.59	32.35	24.73
0	31.16	41.51	32.68	0	28.03	38.40	30.92	0	20.82	32.85	22.97	0	15.39	27.24	22.08	0	23.73	38.75	30.03	1	11.41	9.15	9.72	1	12.20	14.64	22.97	1	13.86	10.69	15.90

Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness
1	13.73	10.00	13.25	1	22.45	18.68	24.73	1	6.65	7.12	4.42	1	16.08	19.47	14.13	1	21.69	21.00	24.73	1	0.00	0.00	0.00	1	22.45	22.92	30.03	1	0.00	0.00	0.00
1	12.23	12.26	16.78	1	14.55	11.13	10.60	1	9.81	10.69	14.13	1	8.28	5.55	7.07	1	5.11	3.98	5.30	1	10.66	7.59	10.60	1	14.55	17.43	21.20	1	0.00	0.00	0.00
1	0.00	0.00	0.00	1	18.47	18.65	22.08	1	14.61	18.50	25.62	1	3.54	2.76	4.42	1	20.10	21.79	24.73	1	2.76	2.38	1.77	1	28.03	26.27	30.03	1	16.99	18.97	19.43
1	14.51	16.15	19.43	1	21.63	18.65	22.08	1	0.00	0.00	0.00	1	0.00	0.00	0.00	1	0.78	0.78	0.88	1	16.83	14.33	17.67	1	5.93	7.52	6.18	1	18.09	15.11	16.78
1	0.00	0.00	0.00	1	0.00	0.00	0.88	1	22.82	19.50	22.97	1	9.84	7.56	8.83	1	24.05	24.52	24.73	1	11.79	13.51	16.78	1	16.11	14.01	15.02	1	12.20	14.26	15.02
1	11.38	12.26	18.55	1	20.94	20.10	20.32	1	8.28	9.09	11.48	1	10.60	9.56	7.95	1	23.29	24.99	27.38	1	7.46	8.72	7.95	1	8.28	10.25	8.83	1	0.00	0.00	0.00
1	0.00	0.00	0.00	1	27.18	24.30	22.97	1	13.76	12.63	16.78	1	0.00	0.00	0.88	1	0.00	0.00	0.00	1	22.35	23.42	26.50	1	16.93	13.51	15.90	1	20.03	20.97	22.97
1	5.08	5.58	8.83	1	14.55	11.57	15.02	1	12.98	10.38	13.25	1	5.08	5.58	5.30	1	14.55	15.83	22.08	1	9.88	6.30	9.72	1	17.68	16.96	22.08	1	9.88	10.69	11.48
1	12.20	12.73	19.43	1	0.00	0.00	0.00	1	5.89	3.57	8.83	1	25.58	25.52	25.62	1	25.64	25.93	19.43	1	12.13	14.70	16.78	1	6.30	6.74	7.07	1	17.68	17.46	18.55
1	16.65	16.99	22.97	1	14.55	14.30	19.43	1	0.00	0.00	0.00	1	1.98	1.57	0.88	1	16.02	18.78	14.13	1	15.36	14.30	12.37	1	20.78	19.09	22.08	1	14.55	12.70	15.02
1	11.44	10.31	12.37	1	13.01	11.47	20.32	1	16.11	17.84	18.55	1	16.08	18.18	19.43	1	0.00	0.00	0.00	1	15.86	21.22	25.62	1	0.00	0.00	0.00	1	11.35	16.65	14.13
1	15.33	15.05	11.48	1	16.83	20.94	25.62	1	7.05	9.15	7.07	1	9.84	9.84	15.02	1	18.47	17.46	16.78	1	12.60	15.05	21.20	1	27.15	21.98	29.15	1	20.06	21.00	18.55
1	12.26	11.94	10.60	1	17.71	14.30	15.90	1	12.23	14.67	16.78	1	18.15	21.79	21.20	1	14.55	13.95	13.25	1	14.48	15.52	12.37	1	16.08	18.97	15.90	1	0.00	0.00	0.00
1	16.11	14.61	15.90	1	16.18	16.21	19.43	1	13.01	13.45	16.78	1	20.00	17.62	13.25	1	21.69	20.69	19.43	1	15.33	14.30	14.13	1	16.27	15.77	19.43	1	14.61	13.51	12.37
1	0.00	0.00	0.00	1	17.68	20.19	23.85	1	10.66	10.66	9.72	1	19.91	20.94	24.73	1	13.01	13.45	17.67	1	16.93	19.84	22.97	1	5.93	8.31	6.18	1	12.95	14.64	11.48
1	17.74	15.86	16.78	1	10.97	12.38	8.83	1	16.93	16.74	15.90	1	14.48	15.49	16.78	1	21.60	21.35	27.38	1	26.43	29.16	28.27	1	0.78	1.19	2.65	1	10.63	10.31	12.37
1	12.20	11.94	14.13	1	21.66	19.94	16.78	1	0.00	0.00	0.00	1	0.00	0.00	0.00	1	20.06	19.00	22.97	1	13.67	16.30	16.78	1	12.16	13.86	15.02	1	8.24	7.93	7.07
1	16.08	17.46	15.02	1	0.00	0.00	0.00	1	16.08	15.08	15.90	1	15.30	14.70	16.78	1	8.31	7.49	10.60	1	16.90	18.56	23.85	1	8.28	7.12	7.95	1	20.72	27.34	26.50
1	8.28	8.72	12.37	1	16.87	17.90	24.73	1	9.88	12.26	12.37	1	5.11	6.74	8.83	1	12.60	8.31	15.02	1	16.11	18.28	18.55	1	0.00	0.00	0.00	1	0.00	0.00	0.00
1	11.41	10.31	10.60	1	20.82	20.94	22.08	1	13.01	9.56	10.60	1	12.16	12.73	15.02	1	15.42	14.26	17.67	1	21.44	25.58	27.38	1	0.00	0.00	0.00	1	15.33	16.65	18.55
1	12.20	10.72	9.72	1	15.39	18.21	21.20	1	10.66	9.53	11.48	1	20.00	35.96	30.92	1	16.87	13.51	15.02	1	17.71	16.99	25.62	1	19.22	13.92	15.90	1	19.19	23.39	22.97
1	21.69	19.81	22.97	1	13.79	14.61	14.13	1	13.79	13.45	15.90	1	20.03	21.82	22.08	1	20.06	24.92	25.62	1	20.91	19.03	22.08	1	21.35	25.30	22.97	1	10.69	11.10	10.60
1	14.61	14.70	20.32	1	0.00	0.00	0.00	1	18.43	15.02	15.90	1	15.36	12.73	20.32	1	17.65	14.26	13.25	1	23.92	32.92	29.15	1	9.81	14.26	16.78	1	17.71	17.02	19.43
1	21.69	37.12	33.57	1	16.87	19.37	17.67	1	9.84	9.53	8.83	1	10.63	11.88	11.48	1	16.15	15.11	22.08	1	18.62	16.58	18.55	1	20.88	25.33	24.73	1	16.15	17.40	22.08
1	14.58	16.99	21.20	1	13.76	13.92	15.90	1	8.28	9.88	15.90	1	15.33	16.65	16.78	1	17.68	18.59	17.67	1	15.77	15.52	13.25	1	9.78	11.85	11.48	1	0.00	0.00	0.00
1	15.36	15.86	13.25	1	20.06	18.68	27.38	1	8.24	9.12	10.60	1	21.60	21.79	20.32	1	15.36	13.04	18.55	1	19.94	28.87	22.08	1	0.00	0.00	0.00	1	8.24	7.59	8.83
1	24.83	20.72	29.15	1	13.01	14.23	18.55	1	0.00	0.00	0.00	1	0.00	0.00	0.00	1	12.20	10.66	15.90	1	7.46	7.56	7.95	1	17.74	19.06	16.78	1	16.90	17.43	23.85
1	12.20	12.70	17.67	1	15.39	18.15	20.32	1	17.71	16.68	17.67	1	0.00	0.00	0.00	1	20.91	21.04	23.85	1	0.00	0.00	0.00	1	16.93	17.56	16.78	1	17.46	20.19	22.97
1	0.00	0.00	0.00	1	22.48	17.96	19.43	1	20.03	22.95	24.73	1	16.11	17.81	18.55	1	16.96	15.86	20.32	1	10.60	12.70	8.83	1	5.89	7.90	8.83	1	15.39	17.02	20.32
1	22.42	24.48	28.27	1	11.04	9.53	12.37	1	0.00	0.00	0.00	1	0.00	0.00	0.00	1	21.60	25.33	26.50	1	14.51	14.67	16.78	1	11.47	17.78	17.67	1	18.43	21.79	22.97
1	18.50	18.65	15.90	1	23.23	19.53	15.02	1	0.00	0.00	0.00	1	0.00	0.00	0.00	1	16.90	18.21	22.08	1	11.44	9.91	12.37	1	16.11	16.65	21.20	1	5.14	5.14	7.07
1	24.80	26.93	30.03	1	19.19	29.22	24.73	1	12.98	13.48	15.02	1	0.00	0.00	0.00	1	12.98	16.24	18.55	1	8.28	8.75	10.60	1	9.84	7.18	9.72	1	29.56	27.68	33.57
1	18.53	18.56	19.43	1	9.44	9.53	8.83	1	12.98	15.77	17.67	1	4.73	4.39	5.30	1	15.33	15.83	19.43	1	13.79	11.88	19.43	1	13.04	13.86	14.13	1	17.74	19.72	25.62
1	19.31	19.00	21.20	1	13.76	17.81	15.90	1	16.99	13.48	17.67	1	14.51	14.70	20.32	1	9.81	13.45	17.67	1	7.46	8.28	13.25	1	13.83	13.86	15.90	1	20.91	17.84	25.62
1	17.84	19.41	27.38	1	16.87	19.37	15.90	1	19.25	24.14	24.73	1	15.33	17.46	18.55	1	20.91	19.53	26.50	1	0.00	0.00	0.00	1	9.84	10.35	9.72	1	11.38	15.42	13.25
1	17.78	17.40	25.62	1	8.28	8.72	11.48	1	11.04	8.34	7.07	1	16.99	18.62	20.32	1	18.50	18.68	20.32	1	16.11	13.89	15.90	1	0.00	0.00	0.00	1	16.18	15.83	14.13
1	17.62	17.15	22.97	1	0.00	0.00	0.00	1	16.15	19.37	18.55	1	13.67	15.17	8.83	1	7.09	5.17	10.60	1	5.96	5.93	7.95	1	13.79	16.24	16.78	1	20.82	16.36	19.43
1	21.73	16.99	18.55	1	18.56	18.53	19.43	1	12.16	7.15	11.48	1	22.38	21.10	28.27	1	0.00	0.00	0.00	1	15.30	17.84	19.43	1	18.47	18.59	21.20	1	17.65	17.15	15.02

Diagnosis	Width	Length	Thickness	Diagnosis	Width	Length	Thickness
1	16.87	15.89	17.67	1	16.93	13.89	12.37
1	12.95	12.32	7.95	1	17.68	20.19	21.20
1	9.12	7.46	8.83	1	18.62	12.26	19.43
1	7.46	6.02	13.25	1	20.16	32.48	26.50
1	0.00	0.00	0.00	1	0.00	0.00	0.00
1	18.50	15.89	18.55	1	14.51	20.63	17.67
1	0.00	0.00	0.00				
1	16.90	15.42	15.02				
1	0.00	0.00	0.00				
1	23.98	19.94	24.73				
1	15.36	14.67	18.55				
1	14.51	17.05	16.78				
1	16.11	12.32	17.67				
1	16.15	14.64	15.02				
1	24.80	20.66	25.62				
1	16.15	12.26	11.48				
1	14.61	14.26	15.90				
1	0.00	0.00	0.00				
1	0.00	0.00	0.00				
1	20.78	19.00	19.43				
1	7.09	7.96	5.30				
1	10.66	13.42	10.60				
1	12.95	13.07	15.90				
1	20.10	16.30	18.55				
1	13.01	13.07	15.02				
1	0.00	0.00	0.00				
1	13.79	10.72	14.13				
1	19.34	17.31	22.97				
1	16.93	16.30	19.43				
1	0.00	0.00	0.00				
1	16.96	14.26	19.43				
1	17.34	18.53	21.20				
1	6.74	9.09	8.83				
1	18.56	17.02	21.20				
1	24.74	21.47	22.97				
1	11.04	12.29	15.02				
1	2.76	4.36	4.42				
1	13.79	10.75	10.60				