MACHINE LEARNING (COMP7703)

Assignment - Semester 1, 2025.

Student Name: Volter Entoma

Student ID: 44782711

1. Introduction

There are many different machine learning algorithms, and many metrics to evaluate them. In this assignment, we will focus on the K-Nearest Neighbour (K-NN) algorithm, Random Forest, and Deep Neural Networks, and the accuracy of each model. We will also looks at the different metrics to evaluate the performance of each model, and use these metrics to compare the suitability of each model for the given data.

2. Aims

- To understand the K-Nearest Neighbour (K-NN) algorithm.
- To understand the accuracy of the K-NN classifier.
- To understand the effect of data imbalance on the accuracy of the K-NN classifier.
- To understand how to use cross-validation to improve the accuracy of the K-NN classifier.

3. Data

The data used in this assignment is the `TTSWING` dataset, which is the provided dataset for this assignment. The dataset contains the 33 features, which are the 33 different measuremeents of table tennis swings. The dataset also contains the height of the player, which is the target variable. The dataset is a CSV file, and can be read into Python using the `pandas` library. The dataset is split into three categories: high, medium, and low.

3.1 Data split

The data is split into three height categories: high, medium, and low. The data is split into 60% training data, 20% testing data, and 20% validation data. The training data is used to train the each model, the testing data is used to test the accuracy of each model, and the validation data is used to compare the performance of the models.

3.2 Data imbalance

After splitting the data, there is an imbalance in the amount of training data assigned to each category in height. There are 17364 (29.73%) data points in the high, 40800 (42.03%) data points in the medium, and 27450 (28.23%) in the low category. This is a significant imbalance; this imbalance is problematic because many machine learning algorithms tend to be biased toward the majority class. This can result in poor predictive performance for the minority classes, as the model may learn to ignore them in favor of

achieving higher overall accuracy. In imbalanced datasets, accuracy becomes a misleading metric, since a model can achieve high accuracy by simply predicting the majority class most of the time, while failing to correctly classify minority class instances. This is especially concerning when the minority classes are of particular interest or importance.

The data imbalanced is addressed by the use of Synthetic Minority Over-sampling Technique (SMOTE) to generate synthetic data points for the low category. The SMOTE algorithm generates synthetic data points by interpolating between existing data points in the low category. This is done by selecting a random data point from the low category, and then selecting a random data point from the k nearest neighbours of that data point. The synthetic data point is then generated by taking a weighted average of the two data points.

After applying SMOTE, the training data is balanced, with 24552 data points in each category. All subsequent analysis is performed on the balanced training data. The testing and validation data is not balanced, and is used to test the accuracy of the models.

3.3 Data standardization

4. Principle Component Analysis

To investigate the effect of dimensionality reduction on the accuracy of the models, we will use Principle Component Analysis (PCA) to reduce the dimensionality of the data. PCA is a linear transformation that transforms the data into a new coordinate system, where the first coordinate is the direction of maximum variance, the second coordinate is the direction of maximum variance orthogonal to the first coordinate, and so on.

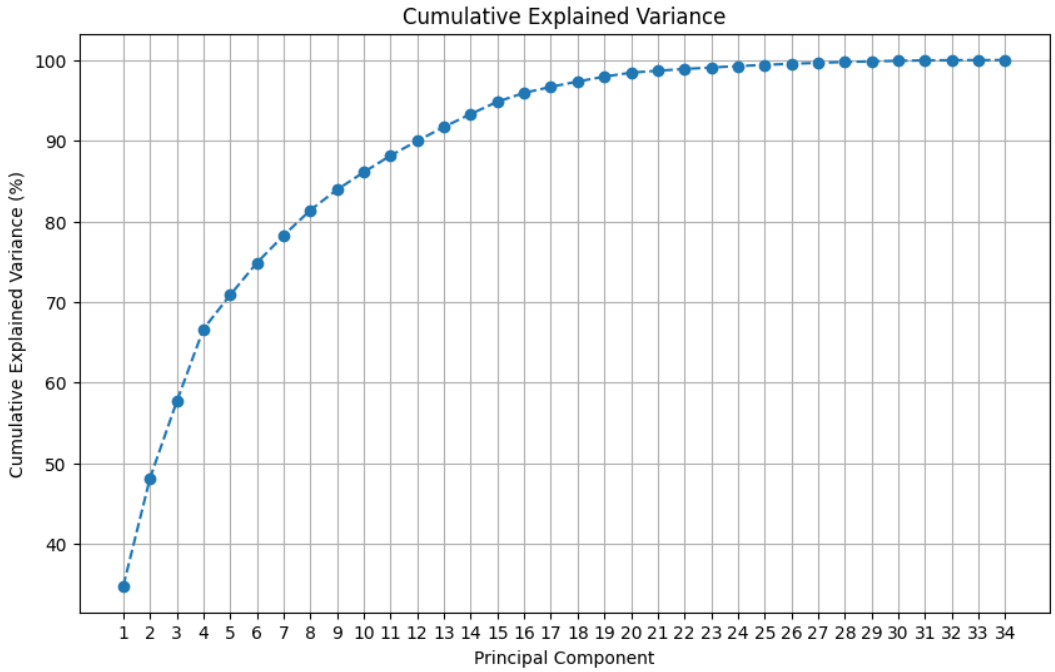

Figure 1: Cumulative variance explained by each principal component.

As shown in Figure 1, approximately 95% of the total variance is explained by the top 16 principal components. This allows us to reduce the dimensionality of our data from 33 dimensions to 16 dimensions while retaining the vast majority of the information content. This reduction offers several benefits:

- Reduced computational complexity in subsequent modeling

- Mitigation of the curse of dimensionality
- Removal of potentially noisy or redundant features
- Improved model interpretability

The effectiveness of this dimensionality reduction will be evaluated by comparing model performance on both the original and PCA-reduced datasets.

5. K-NN classifier

The K-Nearest Neighbour (K-NN) algorithm is a simple and effective machine learning algorithm that can be used for both classification and regression tasks. The K-NN algorithm works by finding the k nearest neighbours of a data point, and then predicting the class of the data point based on the classes of the k nearest neighbours. The K-NN algorithm is a non-parametric algorithm, which means that it does not make any assumptions about the distribution of the data. This makes the K-NN algorithm very flexible, and it can be used for a wide variety of tasks, including for our case of the classifying the height of a table tennis player based on their swing data.
The K-NN algorithm was applied to the data to classify the height of the player based on their swing data.

5.1 Accuracy of K-NN classifier

Using the data split outlined in Section 3.1, the K-NN classifier was trained on the training data, and then tested on the testing data.

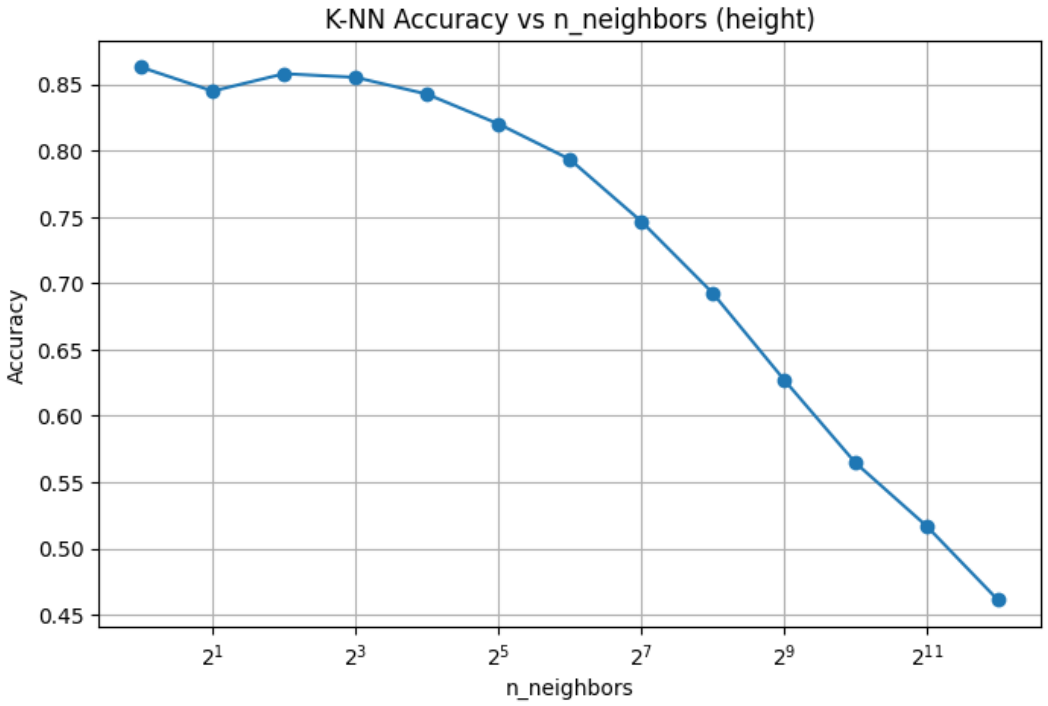

Figure 2: Accuracy of K-NN classifier on testing data.

The accuracy of the K-NN classifier is heavily dependent on the number of neighbours (k) used in the algorithm. Figure 2 shows the accuracy of the K-NN classifier reaches a maximum at 1 neighbour, and

then decreases as the number of neighbours increases. This could be related to the high dimensionality of the data.
The K-NN classifier is particularily sensitive to the curse of dimensionality; as the number of dimensions increases, point within the hyperspace are more likely to be equidistant from each other. This means that the K-NN algorithm is less able to distinguish between points, and hence the accuracy of the K-NN classifier decreases. This is a common problem with high dimensional data, and is one of the reasons why dimensionality reduction techniques such as PCA are used.

We can confirm this by looking at the accuracy of the K-NN classifier on the PCA-reduced data.

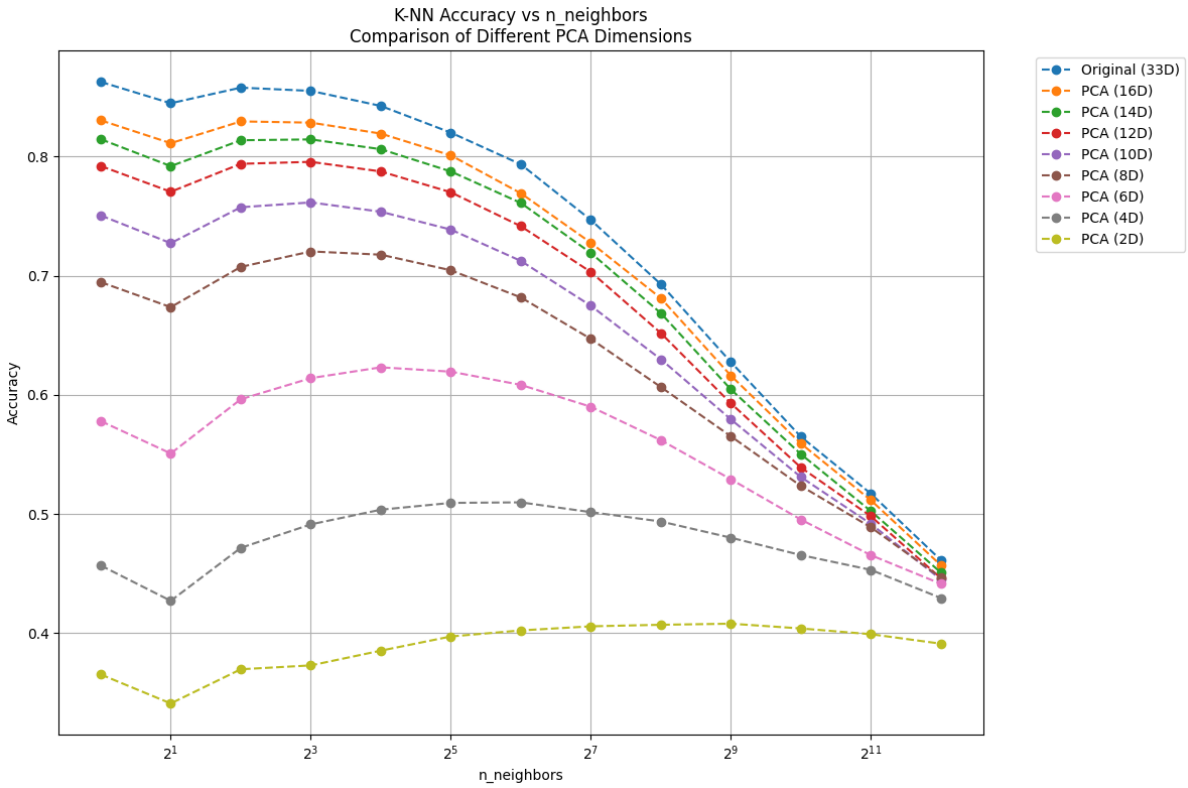

Figure 3: Accuracy of K-NN classifier on PCA-reduced data.

The accuracy of the K-NN classifier on the PCA-reduced data is shown in Figure 3. As we can see, despite the dimensionality reduction, the accuracy of the K-NN classifier is still highest at 1 neighbour, and then decreases as the number of neighbours increases. This suggests that the decrease in accuracy as neighbours increase is not solely due to the high dimensionality of the data, but also due to the nature of the K-NN algorithm on this particular dataset. This idea is supported by the fact that the curve for each PCA-reduced data (16 to 2 dimensions) is similar to the curve for the original data.
Additionally, the accuracy of the K-NN classifier decreases as the number of dimensions decreases. If the curse of dimensionality was a significant problem for this dataset, we would expect the accuracy to increase. This suggests that the curse of dimensionality is not a significant problem for this dataset, and that decreasing the number dimensions is not beneficial to the model, as it loses too much information.

5.1.1 Computational time after dimensionality reduction

The computational time of the K-NN classifier is also significantly reduced after dimensionality reduction. The computational time of the K-NN classifier is largest