Basketball Shooting Prediction Using Machine Learning Models and Motion Capture System

by

Chengxu Liang

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved April 2023 by the
Graduate Supervisory Committee:

Todd Ingalls, Co-Chair
Pavan Turaga, Co-Chair
Gennaro De Luca

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

This project explores the potential for the accurate prediction of basketball shooting posture with machine learning (ML) prediction algorithms, using the data collected by an Internet of Things (IoT) based motion capture system. Specifically, this question is addressed in the research - Can I develop an ML model to generalize a decent basketball shot pattern? - by introducing a supervised learning paradigm, where the ML method takes acceleration attributes to predict the basketball shot efficiency. The solution presented in this study considers motion capture devices configuration on the right upper limb with a sole motion sensor made by BNO080 and ESP32 attached on the right wrist, right forearm, and right shoulder, respectively, By observing the rate of speed changing in the shooting movement and comparing their performance, ML models that apply K-Nearest Neighbor, and Decision Tree algorithm, conclude the best range of acceleration that different spots on the arm should implement.

DEDICATION

Sincerely appreciate the people who showed love in this research, I would not get this done without all this incredible support!

Thanks for the supervision by Dr. Todd Ingalls, who proactively provided me with plentiful resources to make this research happen.

Thanks to Dr. Tejaswi Gowda, and his student, Siva Munaganuru, and Pavan Krishna offering hardware support in the motion capture system.

Thanks to Dr. Pavan Turaga, and Dr. Gennaro De Luca for guiding me to tackle issues in the experiment.

Thanks to Ruohan Sun, he offered huge support during data collection.

And people who intellectually inspired me and financially, and mentally supported me in this research.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

This research leverages my personal experience during basketball training and the engineering development skill introduced in the college, in which I aim to define an elementary question - what characteristics do a successful shot pattern needs to be fulfilled? To address this question, I collect raw data on the court and design machine learning models made of two different types of supervised learning algorithms. Specifically, the training data for this model is constructed by the raw data collected on a 60-feet width by 90-feet length outdoor basketball court. The IoT motion capture system is made by an Inertial Measurement Unit (IMU) BNO080 that is soldered on an ESP32 microcontroller, which is used to record the shooting motion of a right-handed basketball player who shot the ball 4 feet away from the center of the rim. Then basketball shot features are refined from the raw data and are encoded into the model to predict shooting outcomes in terms of a vector. Particularly, 36 preprocessed attributes are engineered and sent into model inputs as feature vectors and each input feature vector contains the acceleration reading along with longitudinal, lateral, and vertical axis lay on the right shoulder, right forearm, and right wrist, respectively. The output of the feature vector is simply made by a binary vector in which 0 and 1 are employed to represent the shot that has been missed, and the outcome that shot has been made, respectively. In the last section, K-Nearest Neighbors (KNN) and Decision Tree (DT) are compiled in the model with 14 training observations which contain both field goal shot and missed shot data. I discuss the performance of each algorithm and conclude the pattern that machine learning informs us that could be worked as an effective basketball shots suggestion in practice. In detail, the KNN algorithm has the best performance with the ability to correctly predict over 90% of test cases. However, because of the insufficient size of training data, the comparison of performance evaluation in two algorithms is debatable so it will be looked over again in the future. Additionally, the decision tree algorithm demonstrates the fact that how the acceleration on the shoulder and wrist dominates the efficiency of a basketball shot posture.

CHAPTER 2

DATA COLLECTION METHOD

2.1 Motion Capture System

The motion capture system (MoCap) is a critical method that allows the user to collect primary data in human movement research. By settling the receiver around or on an actor, the sensor continuously provides data to the receivers which simulate the movement figure on a three-dimensional representation. In this study, the sensor is built upon an Inertial Measurement Unit (IMU) BNO080 (Figure 1) that is soldered on a Lilygo TTGO ESP32 development board (Figure 2), powered by a 3.7 Voltage Li-ion polymer battery made by SparkFun Electronics. It is capable of collecting data at a rate of 115,200 bits/s and sends it to the server via the WebSocket.

Figure 1. IMU BNO080

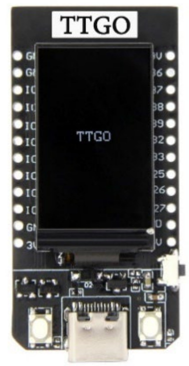

Figure 2. ESP32 Development Board

2.2 Data Processing Method

In order to derive a motion simulation with more accurate data and simplify the future data analysis process, two data preprocessing techniques are inserted into the program.

2.2.1 Acceleration Normalization

Normalization is the basic step of data analysis and plays an essential role to avoid inconsistency when duplicated data appear on different tables. In the primary data of this study, the reading of acceleration is normalized to a pair of 0-1 vectors by dividing an approximate gravity value (9.8 m/s^2).

2.2.2 Low Pass Filter

When the device vibrates at a certain frequency, there is a chance that the vibration data is misinterpreted as a tilt movement. The low pass filter algorithm is taken in to deal with such issues in which the high-frequency data that is produced by sensor vibration, will be filtered out in a way that once the sensor detects a sequence of signal changing quickly, the sensor tends to discount this sort of sequence data and not record it. Instead, the new coming reading is replaced with the integrated value that is computed as the combination of the previous reading and current input. This idea can represent by a mathematics equation as Equation 1 below,

$$\theta_{new} = P_1 * \theta_{measured} + P_2 * \theta_{previous} \qquad \text{(Equation 1)}$$

Where θ represents the degree, P is the normalized possibility and $P_1 + P_2 = 1$

CHAPTER 3

DATA COLLECTION IMPLEMENTATION

3.1 Coordinate System

The coordinate system used in this research follows the representation in Euler angles (Figure 3). Each sensor is a pivot on its own coordinate system in order to simplify the computation.

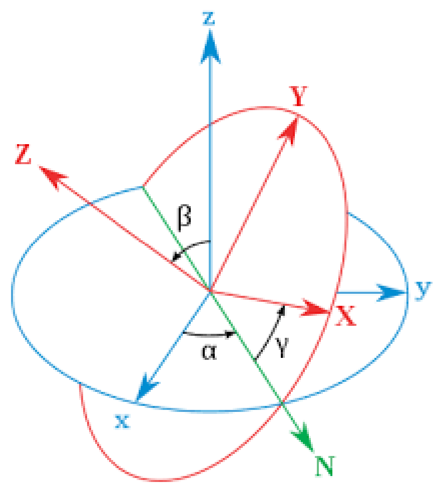

Figure 3. Euler Angles

3.2 Sensor Orientation

This research is interested in partial limb movement when the player shoots the basketball. On the limb of a right-handed basketball player, the sensor is placed around three major joints along the right arm, which are located at the right shoulder, the right forearm, and the right wrist (Figure 4). In order to simplify the process of data calculation, the concept of a local coordinate system is taken in - the X-axis points to the end of fingers along with the coronal plane in terms of the standing body in the natural anatomical position (The green arrow in Figure 4), the Y-axis points along with the tangent of the extension of X-axis (The green arrow in Figure 4), and the movement on the Z-axis goes along with the sagittal plane that perpendicular to the X-Y plane.

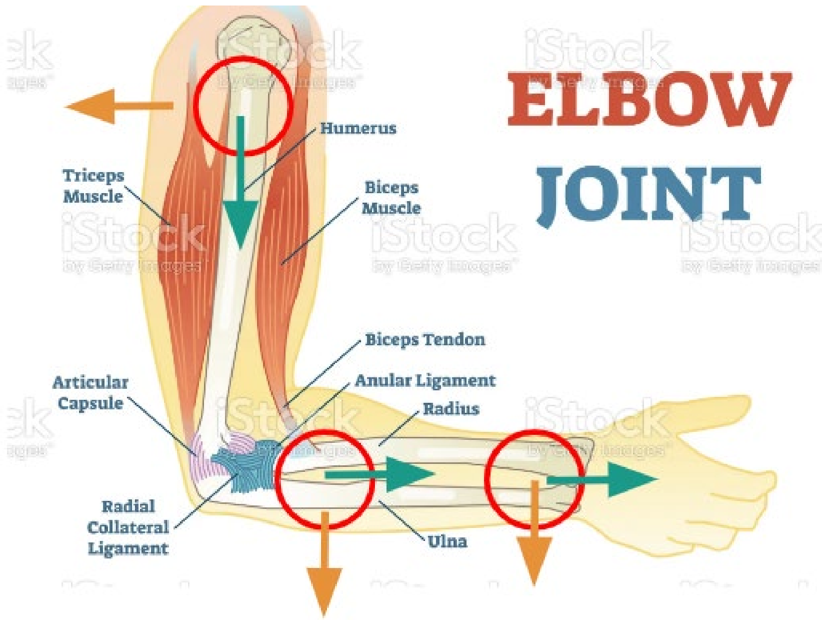

Figure 4. Sensor Orientation on the Arm

3.3 Experiment Process

The data collection experiment is arranged on a standard 60-feet width by 90-feet length outdoor basketball court. The participant is a right-handed basketball player who is capable to make all free-throw shots in ten attempts. The experiment spot is set at 4 feet away from the center of the rim (Figure 5) where the participant attempts to shoot the ball with a constant shooting posture. The motion data is recorded by the same sensor that is separately attached to the right shoulder, the right forearm, and the right wrist during the different shooting sessions. Each shooting session covers all three phases that can be recognized in a natural basketball shot posture, the preparation phase, execution phase, and follow-through phase. In detail, the motion capture system starts tracking the ball from the position in front of the participant's chest, until the follow-through phase is terminated.

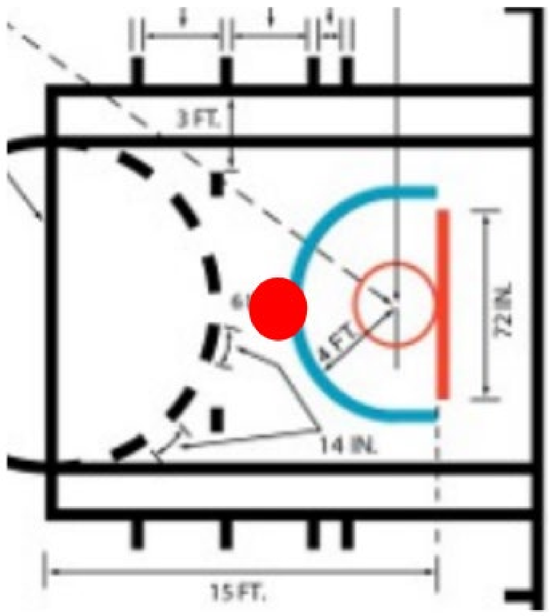

Figure 5. The Location of Shooting Point

3.4 APIs, Libraries, and Raw Data Format

The raw data collected from the sensor is transmitted to the transmission control protocol (TCP) server by running web application programming interfaces (API) on JavaScript code. With help of a Python data analysis library called Pandas that accounts for data manipulation like converting numerical tables and time series into a normalized format and provides enrich operations for data analysis, the raw data is formatted in a Comma Separated Value (CSV) file that contains a sequence of attributes following by the time-series (Table 1),

Table 1 Attributes in The Raw Data

Time Stamp	MAC ID	X	Y	Z	accX	accY	accZ	gyroX	gyroY	gyroZ

Where “Time Stamp” records the timeslot the data comes in, “MAC ID” is a unique, 12-character alphanumeric attribute that is used to identify the individual sensor on the network. “X”, “Y”, and ”Z” are the coefficients of three basic vectors in the quaternion that applied mechanics in three-dimensional space. “accX”, “accY”, and “accZ” stand for the acceleration reading along with three axes in the Euler angles system, respectively. Similarly, the angular velocity on these three axes is represented as “gyroX”, “gyroY”, and “gyroZ”. Figure 6 shows an example of partial raw data,

Time Stamp	ID	X	Y	Z	accX	accY	accZ	gyroX	gyroY	gyroZ
10:51:43:336	78:21:84:88:B9:F4	-0.46	-0.89	-0.03	0.01	0	0.09	0.01	0	0

Figure 6. An Example of Partial Raw Data

Once the collection of raw data for each spot on the limb get done, the irrelevant phase signal will be chopped out and the remaining series of the signal will be wrapped up in a folder named "Court_Data" in which all partial raw data are tag with the format "[Body Part]_[YES/NO]_[Session ID]". For instance, if the sensor on the shoulder detected a successful shot signal that was caught on the third attempt, its figure will be labeled as "Shoulder_YES_3", and so on.

CHAPTER 4

MACHINE LEARNING MODEL DESIGN

4.1 Overview

Training data and machine learning algorithms are the most significant portions that deserve a careful arrangement in the machine learning model design. Specifically, the training data is the initial dataset work for training machine learning models, and algorithms account for refining the rules by implementing the collection of instances. The quality of these two parts will explicitly impact the model's availability and accuracy. This section will start with the process of filtering features, which is a critical step to build a robust training set. Then, it will go through candidate machine learning algorithms with detailed analysis.

4.2 Feature Extraction

Feature extraction is a reduction process in which a large volume of raw data will be condensed based on the most representative data characteristics. In order to find out such information for a successful basketball shot, I break down the raw data I collected from three distinct aspects by which machine learning algorithms are allowed to evaluate the action from different perspectives.

4.2.1 Discrete Point Analysis

The discrete point analysis is often easy and cost-effective because this technique usually does not require the support of a large amount of input data to make inferences. In this study, the maximum and minimum values of acceleration on the individual axis are referenced, by which the model can have a sense of the restriction of the speed changing in the basketball shot. These two values detected on one of the spots on the limb explicitly show the range of acceleration that a successfully shot is supposed to be. However, the discrete point analysis is not enough to reflect the overall picture of a basketball shot. In fact, solely sticking to the discrete point omits a vast amount of motion information that digital signals provide. Because of this disadvantage in the discrete point analysis, the project considers another value to complement the lost figures.

4.2.2 Curve Analysis

By condensing the entire curve into one amount, the average value plays another role in feature mining. This whole-curve character offers a measurement that indicates the central limb motion tendency in a successful shot. On the other hand, this tendency reflected from the average value is super sensitive to extreme values. In other words, if there is a basketball player who shoots with an uncommon posture which causes weird extreme variation in the acceleration, the prediction accuracy in this model that used average value as a main feature would be significantly affected.

4.2.3 Cross-Correlation Analysis

Instead of interpreting the characteristics of one single axis by transforming a couple of figures into one curve, cross-correlation manifests the degree of similarity between two sets of data. The idea is based on the fact that if the differential plot of two series of data shares a point-by-point relationship on a discrete point, the sum of their products will be reflected as a quantification as Equation 2 shows,

$$r_{xy}(l) = \sum_{i=0}^{N-1} x_i y_i \qquad \text{(Equation 2)}$$

Where N is the number of time stamps in two series of data, x_i is the i^{th} data set of the first data series, y_i is the i^{th} data set of the second data series, and r_{xy} indicates the cross-correlation.

This statistical method considers a higher cross-correlation value as the indicator of a high degree of similarity between two series of data, on the opposite side, the motion of two signals that varied without association returned a lower correlation value in the end. In this study, cross-correlation is computed for each couple between each axis, which ends up with a ternary set - the value contained in that shows how the rate of speed changes over each axis would associate with the same attribute on the other two. There are two potential issues that need to be addressed here with this form of the cross-correlation function (Equation 2). Firstly, two series of data can be unmatched if to shift one of them toward the end of the time axis, some point-by-point multiplications that occurred in the unshifted set might be ignored after one of the datasets is manipulated. For instance, consider a case in which an unshifted array [1,3,5,7] relative to the second array [4,6,8,10] is shifted by one unit, and new match pair can be demonstrated as 1-6, 3-8, 5-10. Though, this mapping would not contain the information reflected on the 7-4 relationship. Instead, the computation left with a shortened series. Since the movement on the arm during basketball shooting is not circular motion, it is okay to leave this issue aside and still have valid information on relationship quantification for three axes. The second issue arises because the equation used in the program (Equation 2) is not unitless. This issue matters when two tables contain two sets of data with the varied unit, which results in high difficulty in comparing cross-correlation, and even worse that may end up with a misleading comparison. In order to prevent this issue, the cross-correlation derived from the function above can be normalized by an additional operation, which can express as Equation 3 shows,

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0)}\sqrt{r_{yy}(0)}} \qquad \text{(Equation 3)}$$

which by dividing by the product of the square root of the autocorrelation of x at zero lag and the square root of the autocorrelation of y at zero lag. Again, fortunately, I don't have to worry about applying Equation 2 directly in this case since the contrast of acceleration on each axis shares the same type of unit.

4.3 Training Dataset

Overall, the training data construction in this research combines all feature analysis bullet-point above. Table 2 demonstrates 12 features that are selected to trace the trend for each spot on the shooting arm, where Acc., Avg, and CC stand for the acceleration, the average value, and the cross-correlation value, respectively.

Table 2 Attributes for the Movement on Single Spot

Max Acc. on X	Min Acc. on X	Avg Acc. on X	Max Acc. on Y	Min Acc. on Y	Avg Acc. on Y	Max Acc. on Z	Min Acc. on Z	Avg Acc. on Z	CC X/Y	CC X/Z	CC Y/Z

In total, there are 36 features employed to reflect the holistic right-arm movement when the participant shoots the ball (Table 3).

Table 3 Attributes in the Training Data

SH Max Acc. on X	SH Min Acc. on X	SH Avg Acc. on X	SH Max Acc. on Y	SH Min Acc. on Y	SH Avg Acc. on Y	SH Max Acc. on Z	SH Min Acc. on Z	SH Avg Acc. on Z	SH CC X/Y	SH CC X/Z	SH CC Y/Z
FA Max Acc. on X	FA Min Acc. on X	FA Avg Acc. on X	FA Max Acc. on Y	FA Min Acc. on Y	FA Avg Acc. on Y	FA Max Acc. on Z	FA Min Acc. on Z	FA Avg Acc. on Z	FA CC X/Y	FA CC X/Z	FA CC Y/Z
WR Max Acc. on X	WR Min Acc. on X	WR Avg Acc. on X	WR Max Acc. on Y	WR Min Acc. on Y	WR Avg Acc. on Y	WR Max Acc. on Z	WR Min Acc. on Z	WR Avg Acc. on Z	WR CC X/Y	WR CC X/Z	WR CC Y/Z

Where SH, FA, and WR indicate the shoulder, the forearm, and the wrist, respectively.

4.4 Machine Learning Strategies

4.4.1 Classification vs Regression

Both classification and regression algorithms are used to forecast outcomes in supervised machine learning by labeled datasets. The key difference between them is that the regression algorithm is used to predict the continuous result such as the price, the energy, etc., though the classification algorithm is applied for distinct value prediction, For instance, in the True/False or Male/Female Prediction. In the case of this research, the consequence of a basketball shot is expected to be forecasted in the model, which is considered a type of distinct value as there are only two types of outcomes for a basketball shot. Thus, the classification algorithm is employed by identifying new basketball shot observations in the training data that is consisted of multiple feature vectors that are marked with two types of shot outcomes.

4.4.2 Parametric vs Nonparametric

In the ML model design, it is important to have a sense of the kind of data I inhibited and choose the proper algorithm category to fit research demands. In the last subsection, I differentiate a variety of machine learning strategies based on the need of the model output, and now I aim to optimize my algorithm selection by looking over the characteristics of the existing dataset. Considering the small volume of my training data and it made by constant parameters, I choose to start my model design with the parametric machine learning model. By the definition, a parametric model "summarizes data with a set of parameters of fixed size (independent of the number of training examples), No matter how much data you throw at a parametric model, it won't change its mind about how many parameters it needs." In addition, the parametric model features with simple-use and fewer data required. As a trade-off, the type of model is not reliable if the research problem has complexity with a variety of uncertain mapping forms. To resolve the poor fit in such situations, it is necessary to introduce the nonparametric machine learning algorithm to complement the flaw in the application applied parametric method. In contrast, the nonparametric machine learning algorithm seeks fit by generating an independent mapping function in the training data, meanwhile, it maintains the generalization for the unseen data. As such, assembling nonparametric algorithms is precedent when the data comes up with a large volume of training information. "Nonparametric methods are good when you have a lot of data and no prior knowledge, and when you don't want to worry too much about choosing just the right features."

4.5 Machine Learning Algorithms

According to the analysis above, it is getting clear that the category of the nonparametric classification algorithm is a good way to approach the goal of this research. Particularly, K-Nearest Neighbors (KNN) and Decision Tree are selected. Each algorithm will be clarified in its own subsection, which includes the algorithm principles, the implementation, and the analysis of its merits and flaws.

4.5.1 K-Nearest Neighbors (KNN)

KNN is an instance-based algorithm that provides powerful functionality for both supervised learning and unsupervised learning task. As a foundation of many other learning algorithms, KNN is either able to classify data with distinct labels or handle continuous labels in the regression computation. The principle behind the KNN algorithm is to find a predefined point of the stored instance in the training dataset in distance to the new point in the test data. By reading through all instances in the training data, the algorithm will return the case that is closest to the unseen data point as the outcome of the prediction (Figure 7).

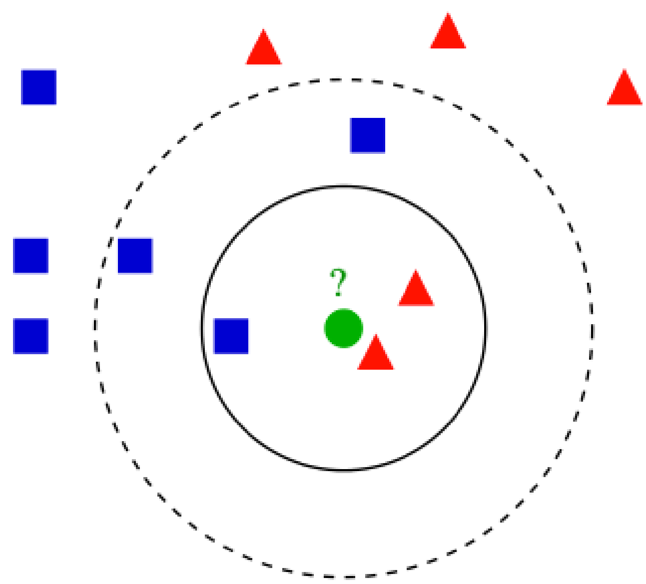

Figure 7. A Diagram of KNN

On the other hand, this type of workflow weakens the capability of resolving larger amounts of data because it takes time to iterate each piece of data in the training set. Another feature that differentiates KNN from other machine learning algorithms is KNN does not require a list of hyperparameters being set up ahead. Instead, the only hyperparameter it needs is the K value, which indicates the number of neighbors that the algorithm should look over.

A perfect example that could help to understand the mechanism of the KNN algorithm is to recall the scenario that how would a person be impacted by surrounding people. In reality, individual personality, mentality, standards, etc., are likely to be shaped by the individuals close to them. For instance, the behavior pattern of an adolescent usually is the reflection of the pattern of his parents assuming he lives with them most of the time. The prediction in the KNN algorithm follows the same logic – in which the outcome of unseen data is determined by selecting the majority as the result of the target data point. The hyperparameter K makes things interesting because it can be adjusted to modify the range of the context. There is one aspect worth noting here – the KNN regression works slightly differently than what the KNN classifier does. Instead, the point value is determined by the mean value of K closest points in the KNN regression.

4.5.2 Decision Tree

The decision tree is another nonparametric supervised machine learning algorithm that is used in both classification and regression. This algorithm has a tree-type, hierarchical structure made of the root node, internal nodes, and leaf nodes (Figure 8).

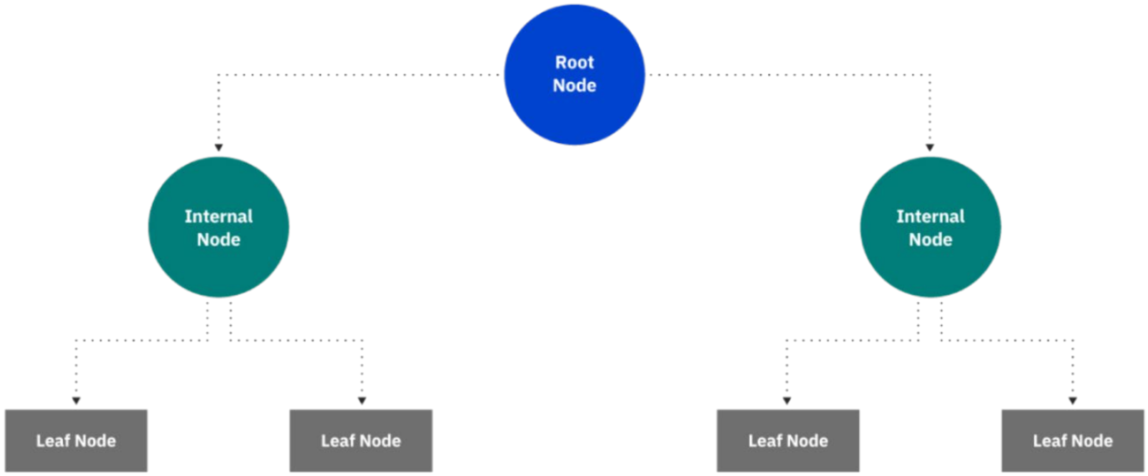

Figure 8. The Structure of Decision Tree

The decision tree learns data in a divide-conquer strategy by applying a greedy search approach. The algorithm pulls out features that are most representative in training motion interpretation and splits them within the tree. This process will recursively run in a top-down manner until most of the data are classified as homogenous sets. In the practice of this study, each node contains at most 5 sets of information as an example in Figure 9 shows,

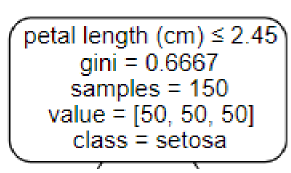

Figure 9. An Example of a Decision Tree Node

The node above is shown as a part of the decision tree in an example that is utilized to predict the category of flowers. It is easier to interpret the internal information backward – "class" tells us the prediction the node will make, and the number in the "samples" shows the number of examples that fall into this prediction. Additionally, a list in the "value" gives further detail about the number of examples in each prediction. The information on the top indicates the feature the algorithm chooses to split - it classifies data based on the petal length in this case. If the petal length in the data is no larger than 2.45 cm, the data will be classified into the node on the left, and the rest of the samples will be sent into the right node. There is an important index called the Gini index, which is a measurement that quantifies the purification of the node. A Gini index that is greater than 0 indicates the sample contained within the class might belong to other categories. In other words, a zero Gini index implies that all samples in the current are explicitly classified in the same category. Compare to other machine learning algorithms, these nodes' information provides a better-visualized understanding of machine learning workflow which most algorithms are not able to approach.

CHAPTER 5

RESULT AND DISCUSSION

5.1 Overview

The content in this section includes the machine learning model performance analysis and the information that the machine learning model guided in the practiced basketball shot training. In the discussion, I go through this project as a whole and list the potential issues that could be improved in the future. In the end, I will talk about the future application of this research because I hope this project can serve as a feature in my other project – which is a cross-platform web application employed to record and visualize athletic data.

5.2 Machine Learning Model with KNN

Since the value of neighbors K is the only hyperparameter in the algorithm, I first make a chart to highlight the performance of each K value by inputting both original training and test data (Figure 10). According to the information in the diagram, this model makes decent predictions with 14 training examples because the fact that the training set consists of 9 positive and 5 negative examples guarantees the accuracy of this KNN classifier would not below the portion of the positive cases (approximately 64.3%). The diagram also shows that the model has the best performance when the K value has been set up to 2, 4, 5, and 6. Under this circumstance, I will choose a smaller K value to run the algorithm because a larger K value would result in more data points involving.

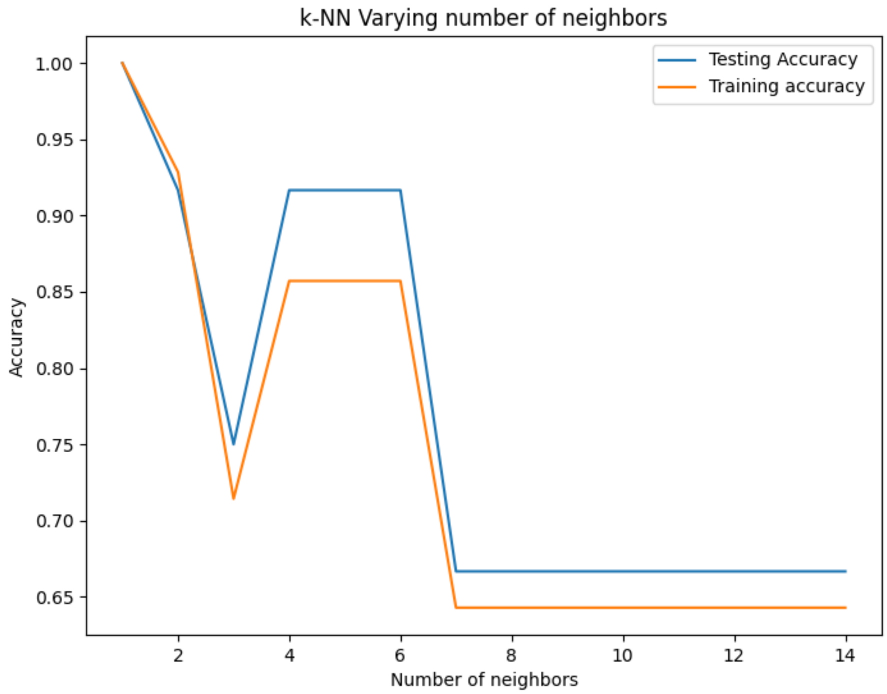

Figure 10. The Change of Accuracy with Varying Number of Neighbors

However, we can observe that the accuracy of the classification model is significantly unstable when the K value is in the range of 3 to 7, then the tendency of accuracy stays around 65% gently no matter what kind of data set we used to evaluate the model. I deduced that is because the feature in the successful basketball shot is highly overlapped with the feature in the failed shot. It is unrealistic to visualize the distribution for all 36 features so I reduce the number of features to 2 which can offer a touch of sense of how the classification is being placed in the prediction.

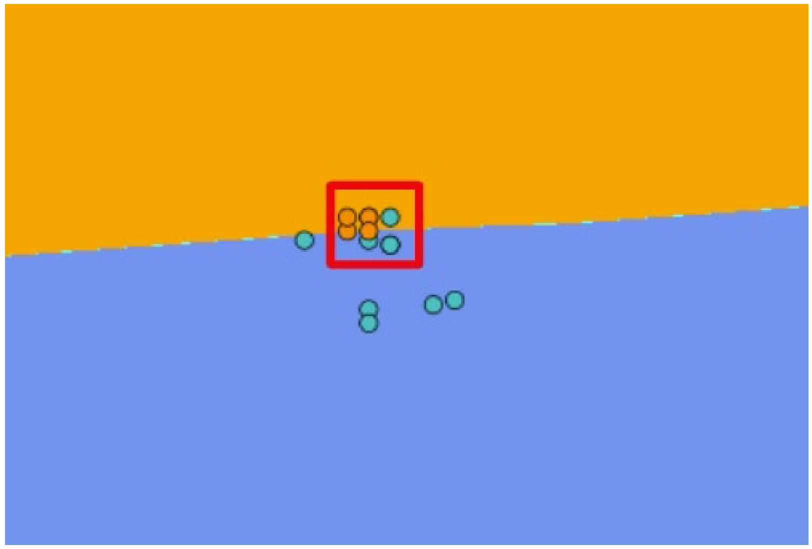

Figure 11 Data Classification with Max_Z_W and Min_Z_W as Features

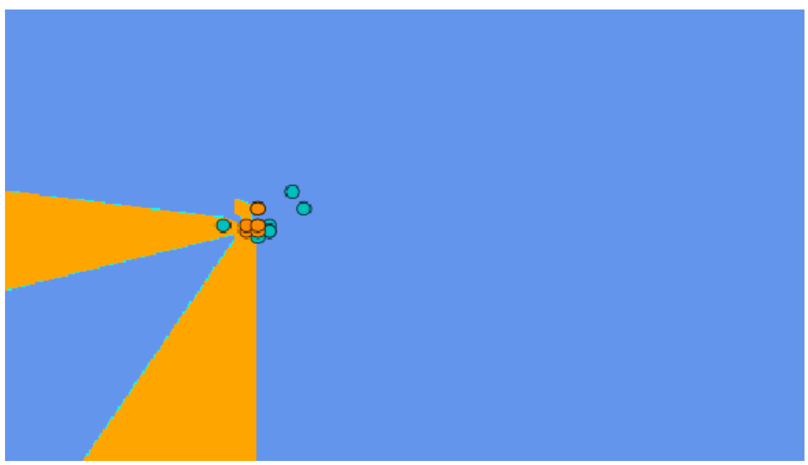

Figure 12 Data Classification with Max_Z_W and Min_Y_S as Features

Both Figure 11 and Figure 12 demonstrate the distribution of two shooting outcomes – specifically, figure x employed the maximum acceleration on the Z axis that collected on the wrist, and the minimum acceleration on the Y axis that collected on the shoulder as the primary attribute to classify basketball shot, and figure y employed same type of feature on the wrist but it is paired with the minimum acceleration on the Z axis. These two charts inform us of an elementary phenomenon in the distribution that the category of failed shot (orange dots) tends to gather inside of a compact region most of the time where it also contains some examples that belong to the group of a successful shot, which may result in the incorrect classification even evaluating model with the identical training data. In other words, the target data point that lands close to the packed area always gets a chance of being incorrectly predicted, which is demonstrated more explicitly by the combined information in Figure 11 and Figure 13.

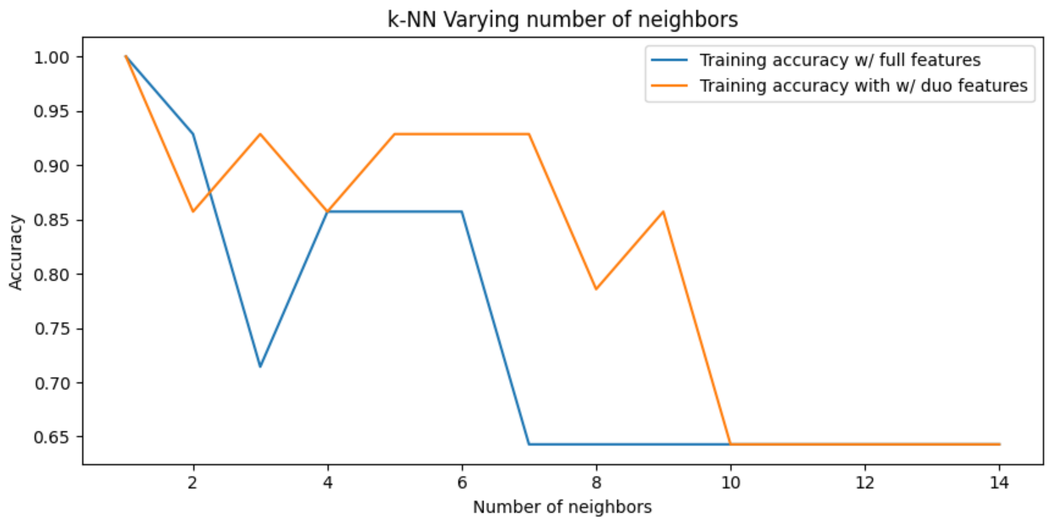

Figure 13 The Trend of Varying Accuracy

Figure 13 shows the performance of two different sizes of features training with varying K values – therein, the feature that utilizes in the duo feature is as same as the feature used in Figure x, and it reflects the pendulum in the accuracy starting from k = 2 until k = 10. We can find a hint about why the wave ends up at k = 10 by looking over the pattern in Figure x, in which there are 5 cyan data points distanced away from the highlighted area. We can consider this as the area would not get confused by the chaos in the highlighted region. The outcome of the total number of examples (14) subtracted by the value above (5) supports this idea because it is exactly the k value that the flat tendency starts from. By which we can get informed about the importance of the space among different dots mean to the stability of accurate classification.

Beyond that, there are three essential metrics are utilized to generally evaluate the machine learning model performance in this project. The value of accuracy identifies the predicted correctness over holistic results (Equation 4).

$$Accuracy = \frac{True\ Positive + True\ Negative}{All\ Ouputs} \qquad \text{(Equation 4)}$$

The precision gives a sense of accuracy type of information regarding how well the model correctly predicts positive outcomes (Equation 5),

$$Precision\ = \frac{True\ Positive}{True\ Positive + False\ Positive} \qquad \text{(Equation 5)}$$

The recall is a value that indicates the completeness of positive cases predication (Equation 6),

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \qquad \text{(Equation 6)}$$

Figure 11 demonstrates the KNN model evaluation when K = 2,

```
Accuracy: 0.9166666666666666
Precision: 1.0
Recall: 0.875
```

Figure 14. The Evaluation of KNN Model

In this case, the model is able to correctly predict 91.7% of basketball shot motion, including correctly predicting all basketball shot that has been successfully made, and capable of differentiating 87.5% of field goal postures.

5.3 Machine Learning Model with Decision Tree

We can always be aware of the process of the decision tree algorithm by visualizing the node tree, which is super helpful to correlate the view of the machine learning model with the practical operation. For instance, by running the same training set, we have a decision tree report like this (Figure 12),

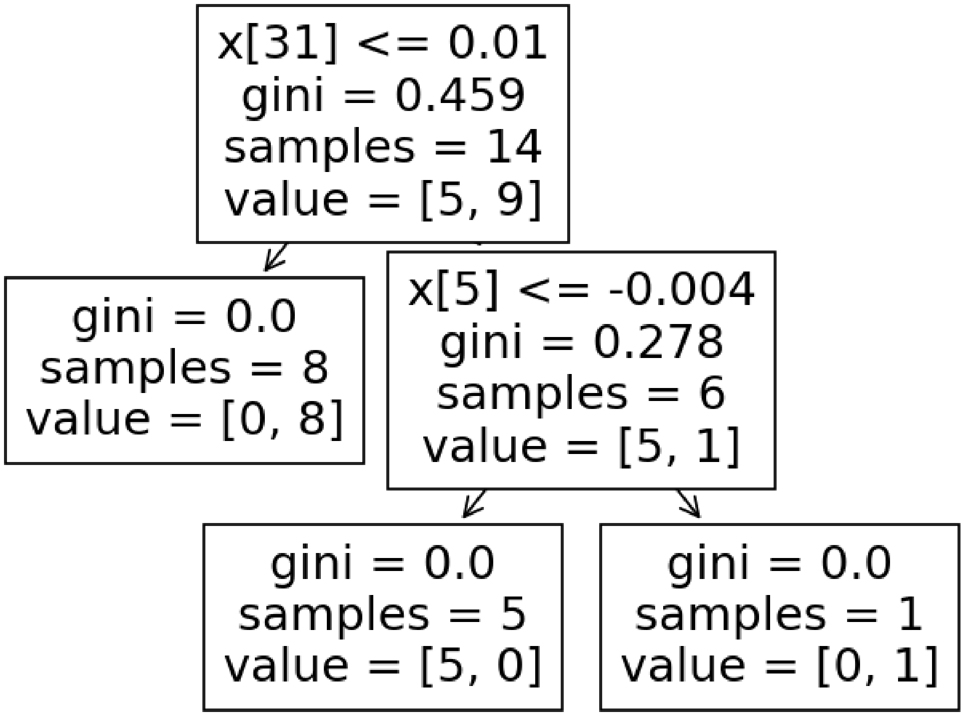

Figure 15. Decision Tree Visualization

We noticed that x[31] is constantly a feature that is selected in the root node of the algorithm, which is an attribute that indicates the maximum acceleration reading on the Z axis collected on the right wrist. The diagram informs us it is always a case that the player can make a shot when this acceleration reading is no larger than 0.01 m/s^2. It can be proved in reality that the shooter should never ever exert extra force along with the perpendicular direction on the wrist because it is the main reason that results in the ball being off the shooting path. Another feature that the algorithm used to distinguish shot posture is the minimum acceleration reading on the Y-axis collected on the shoulder (x[5]). According to the information inside of the node, the algorithm believes that it is impossible to make a shot under the circumstance if this acceleration value is no larger than -0.004 m/s^2. This phenomenon can be translated in practice as the shoulder is not properly extended during shooting basketball, which usually occurs when the basketball player lacks strength on the shoulder and the back. Overall, the decision tree algorithm genuinely reflects the elementary shooting quality that a professional shooter should hold his shoulder tightly and evenly with proper ball-pressing on the wrist and keeps towards the rim. In the end, Figure 12 demonstrates the Decision Tree model evaluation,

```
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
```

Figure 16. The Evaluation of Decision Tree Model

As we observe, all values go to 1 because of the insufficient training data, that exactly is the point I need to improve in the next phase of this research.

5.4 Discussion and Future

5.4.1 Aspect of Motion Capture System

Enhance the accuracy and endurance of motion capture sensors - The motion capture system used in this research is folded in a rough 3d-printed box with an unstable physical structure which not only results in the vibration that generates misleading readings but also is not qualified to engage in the intense exercise because of its fragility.

5.4.2 Aspect of Machine Learning Model

a. Training data extension - As I mentioned above, the volume of training data significantly impacts the machine learning model's accuracy and quality. In order to take this project to approach practical application, I would prioritize extending the size of dataset in the future. Besides, I only collect the motion data on the shooting arm and discard the impact from other body parts. The reliability of the machine learning model gets hurt by solely concentrating on partial movement since shooting a basketball is a systemic motion. In addition, in order to help the model generalize the shooting posture pattern accurately, it is necessary to acquire data from different shooting distances.

b. Explore other proper machine learning algorithms - Sticking on the current model that is employed to predict the outcome of a basketball shot, I can utilize a method like the neural network algorithm to automatically learn the field goals' features from the training examples instead of labeling the output for each feature vector. Moving forward to another approach in which I attempt to discover the change of energy that exerts on the shoulder by adjusting shooting posture, the machine learning model turns out to be a regression model instead of a model solely able to be tackling classification problems.

5.4.3 Aspect of Experiment Methods

Multiple Sensors Synchronization - Currently, the data in each session of the basketball shot posture is separately collected by the same sensor. On the good side, this patchwork type of data collection method gives a brief pattern of limb movements in the basketball shot by breaking down an integral movement as pieces which helps the researchers generate the targeted solution to improve basketball shot skill. However, this method lacks data correlation on each joint which might impede future data analysis on other body parts.

5.4.4 Aspect of Entire Project

I am working on a cross-platform web application development that features data collection by a motion capture system, data management, and data visualization. I aim to integrate the part of data analysis in this research into this ongoing program in order to land the idea in the practice that helps athletes enhance their performance and reduce the chance of getting injured.

.

REFERENCES

Stergiou, N. (2004). Time Series Analysis: The Cross-Correlation Function. In *Innovative analyses of human movement*. essay, Human kinetics.

Russell, S. J., & Norvig, P. (2015). *Artificial Intelligence: A Modern Approach*.

Kinerd, N. I. (2012b). Motion capture study of human movement recognition. *Clemson Universeity Tiger Prints*.

Arithmetic Average Advantages and Disadvantages - Macroption. (n.d.). https://www.macroption.com/arithmetic-average-advantages-disadvantages/

Writing WebSocket servers - Web APIs | MDN. (2023b, February 19). https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API/Writing_WebSocket_servers

Hamilton, W. (1853). Lectures on quaternions. *PREFACE*.

Huilgol, P. (2023). Precision and Recall | Essential Metrics for Data Analysis (Updated 2023). *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/#:~:text=Precision%20and%20recall%20are%20two,the%20completeness%20of%20positive%20predictions.

Shung, K. P. (2020, April 10). Accuracy, Precision, Recall or F1? - Towards Data Science. *Medium*. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Navlani, A. (2018). *Decision Tree Classification in Python Tutorial*. https://www.datacamp.com/tutorial/decision-tree-classification-python

Shung, K. P. (2020b, April 10). Accuracy, Precision, Recall or F1? - Towards Data Science. *Medium*. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

Carnegie Mellon University. (n.d.). *How to Find Your Device MAC Address - Computing Services - Office of the CIO - Carnegie Mellon University*. https://www.cmu.edu/computing/services/endpoint/network-access/mac-address.html#:~:text=A%20MAC%20(Media%20Access%20Control,electronic%20devices%20on%20a%20network.