Eigenjumps: analyzing human movement on tactile sensors via singular value decomposition and other linear algebra techniques

Joshua Gao, Owen Xu Li, Skylar Wang, Ian Lim, Annie Gu, Amanda Cao
MATH 232: Advanced Applied Linear Algebra
Professor Ian Adelstein
May, 2025

Abstract

The study of athlete performance has become more precise than ever before with new insights in modern medicine, physiology, and recovery methods. Acknowledging this, we gathered the data of 18 different jumpers performing around 10 counter-movement jumps (CMJs) each. Sensors from Vault Kinetics measure relative force on an 8 x 10 sensor pad, measuring relative force at 60 frames per second. We aim to compare movement patterns between individuals, extracting their "jump signatures" and characteristics unique to that person. Then, we use SVD for dimension reduction and an SVM for jump classification. PCA was used to create "jump profiles" describing the average jump force profile of an individual that express major differences between high performers and dummy data. Teammates with identical training regimens showed highly similar jump profiles. The highest jumpers, although similar in jump height, still showed noticeable differences in jump profile. PC1 and PC2 accounted for more than 60% of jump variance. SVD at rank one was able to successfully dissect distinct pressure behavior of jump tendencies. The developed SVM with a stratified K-fold CV (5-fold) and RBF kernel was accurate in approximately 71% of trials to accurately identify an individual based solely on a single jump signature.

1 Introduction and Background

Over the past decade, advances in physiology and biomechanics have increasingly influenced elite athletics. Of particular interest in the field has been ground force reactions (GRFs), the forces exerted by the ground on the body during movement, defining motion up through the body's kinetic chain. In this paper, we propose a method for analyzing human performance using singular value decomposition (SVD) on ground reaction force (GRF) data.

Just as every individual has a unique fingerprint or signature, we hypothesize that the complexity of human movement gives rise to a unique "jump signature" for every individual. By decomposing high-dimensional GRF data during jumping tasks, we aim to uncover the patterns that define how individuals move. In particular, *classification* of jump signatures is important. Classifying jump signatures by athlete role (e.g. soccer: center, forward, defense) may yield insight in desirable movement mechanics required for different purposes. Classifying by athlete caliber (i.e. professional, recreational, amateur) may yield insight into the developmental patterns that mark the most successful, quickly improving, athletes. Perhaps the most impactful application of jump signature classification: classifying against longitudinal physical health outcomes – extracting key indicators that may preempt an injury or medical condition. The ultimate goal is to predict injuries before they happen.

This paper provides a framework for learning from high-dimensional movement data.

2 Methods

2.1 Sensor Data Collection

Our data was collected using a capacitive tactile sensor system developed at the Yale School of Engineering and Applied Science[2]. The system was configured in an 8×10 matrix of sensors (80 total) embedded in a $2' \times 2'$ rubber surface.

Each sensor is sampled four times per measurement cycle, and the average is used to reduce noise. Sensors are read sequentially, such that a complete scan across all 80 sensors constitutes one frame. These frame scans occur at 60 Hz, resulting in a total of $60 \times 80 \times 4 = 19{,}200$ individual measurements per second. To further smooth the signal with minimal latency, we apply a trailing moving average with a window size of five to each sensor's time series.

The structure of the recorded data closely resembles that of an image, where each sensor resembles a pixel representing localized pressure. This format enables the use of mathematical tools developed for image processing to be used to analyze human performance. In particular, we draw inspiration from facial recognition algorithms (authentication and athlete identification) and image compression techniques (extracting significant features) to analyze jump signatures.

Movement data was collected for 18 individuals, each performing a sequence of standard countermovement jumps (CMJs). Our analysis reveals that each person's force-time profile forms a distinctive jump signature, reflecting how individuals uniquely generate force during the propulsion phase of the jump and absorb impact upon landing.

Figure 1: Jump Testing

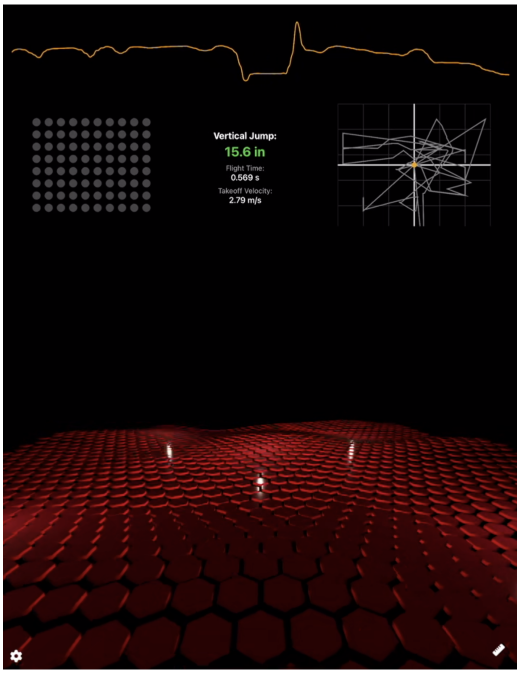

Figure 2: Testing Dashboard Interface

The sensors are calibrated and tared before data collection for each individual . Each individual was instructed to perform a max CMJ jump, trying to jump as high as possible. After the data collection period of 10 jumps, which typically lasted 1-3 minutes, a large dataset of $60 \times 80 \times$time (in seconds) is gathered.

From the force data, jump events can be flagged as moments when the force measurement is 0 (individual is in the air). The duration of a person's time in the air can be used to calculate their vertical jump height, takeoff velocity, or impulse.

Our jumpers are introduced in Table 1. Note that not all jumpers' data was used during PCA or SVM for the purposes of interpretability, since some jumpers did not exactly follow the jump protocol.

Person	**Description**
Owen Xu Li	TD Buttery "Goofy Goober" Enthusiast
Owen Xu Li	Jumping with no knees — dummy data
Collin	Yale Baseball, Infielder
Noah	Whiffenpoof
Connor	Powerlifter
Matthew	Vault!
Annie Gu	Cannot Get Out of Bed Most Days
Skylar Wang	Ultimate Frisbee Enthusiast
Ian Lim	CS Major
Leo	Yale Football, Offensive Lineman
Joshua Gao	Vault!
Garrett	Yale Baseball, Utility
Emma	Yale Track and Field, Throws
Jake	Yale Track and Field, Throws
Chu	Yale Track and Field, Throws
Karoline	Yale Women's Soccer, Midfield
Sylvie	Yale Cross Country
Charles Sun	Fishing and Math Enthusiast

Table 1: Description of subject tests

2.2 Extracting Jump Signature

We adopted a fixed-window approach to extract jump signatures — centering a 180-point window (90 points before and after) around the temporal midpoint between each release and landing. This alignment isolates the key phase of each jump for consistent comparison. To ensure data quality, we applied several filtering steps. First, sensors with near-zero mean values (between -0.2 and 0.2) were classified as inactive and excluded to reduce noise. We then required a minimum aggregated pressure threshold of 20 units across active sensors to filter out minor contacts and misfires, ensuring that only meaningful jumping events were retained. Finally, we restricted flight times to between 0.3 and 0.82 seconds to exclude shifts between jumps and sensor glitches so that our jump data only included the cleaned/controlled jumps.

This standardization process serves a similar role to cropping and aligning faces in facial recognition: by cropping force profiles around comparable points in the jump cycle, we make it possible to meaningfully compare different events. Without this alignment, variations unrelated to the jump itself, such as when the jump started or how sensors responded off-baseline, would introduce noise and obscure underlying patterns. The final processed dataset was exported to CSV format.

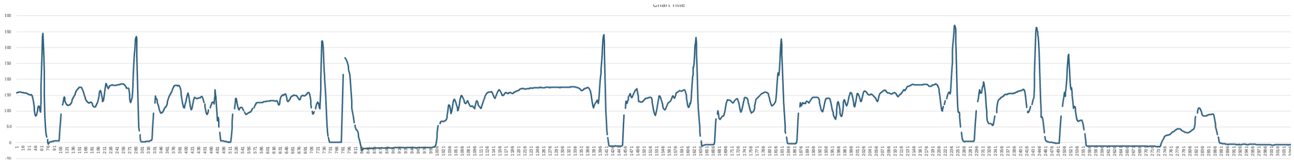

Figure 3: Raw Sensor Reading Collin

2.3 Performing PCA

For the purposes of interpretation, we selected 12 subjects' jumps to perform PCA. We then selected the best five jumps per subject by removing jumps with unusually high noise, or clear outliers — for instance, some verticals were measured to be over 30 inches high when the rest were around 15 inches.

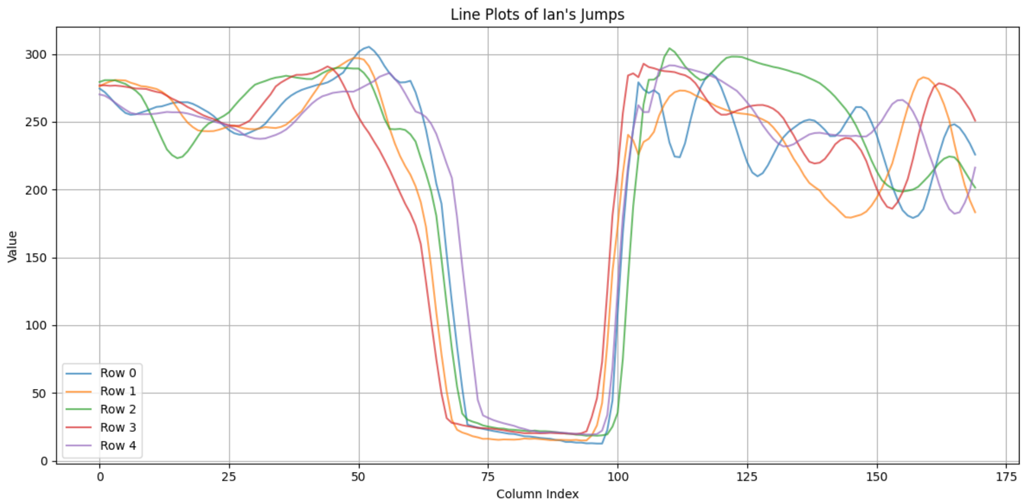

Figure 4: Plot of Ian's selected five jumps

The figure above displays an example plot of the resulting cleaned data.

We then concatenated the selected jumps of all selected subjects into one dataset and mean-centered the data. Moreover, we divided each column by its standard deviation. Hence, each column has $\mu = 0$ and $\sigma = 1$.

$$B = \begin{bmatrix} | & \cdots & | \\ \frac{\boldsymbol{x}_1 - \bar{x}}{\sigma_1} & \cdots & \frac{\boldsymbol{x}_n - \bar{x}}{\sigma_n} \\ | & \cdots & | \end{bmatrix}.$$

Finally, we utilized `np.linalg.svd()` to decompose the standardized matrix into $B = U\Sigma V^T$.

2.3.1 Teammate Comparisons

Beyond athlete identification, we were also interested in analyzing jump profiles between athletes of similar sports. Further PCA was performed on cleaned data that resulted in $m \times 170$ data matrices that were further simplified into 170×1 vectors that would encompass movement.

The following was performed on these $m \times 170$ matrices:

1. **Transpose:** The matrix is transposed to $X^\top \in \mathbb{R}^{170 \times m}$ so that each of the original columns becomes an observation and each row a variable.

2. **Centering and Scaling:** PCA is applied to the transposed matrix. Internally, this standardizes the matrix:
$$Z = \text{scale}(X^\top) \in \mathbb{R}^{170 \times m}$$
where each column is centered to have mean 0 and scaled to unit variance.

3. **Covariance Matrix and PCA:** PCA computes the eigenvectors of the covariance matrix:
$$\text{Cov}(Z) = \frac{1}{m-1} ZZ^\top$$
Let $v_1 \in \mathbb{R}^m$ be the first principal component direction (i.e., the eigenvector corresponding to the largest eigenvalue). The data are projected onto v_1 to obtain the first principal component scores:
$$\mathbf{z}_1 = Zv_1 \in \mathbb{R}^{170}$$

4. **Return:** The function returns $\mathbf{z}_1$ as a column vector (i.e., an 170×1 matrix), which is a one-dimensional summary of the original m-dimensional feature space.

2.4 Performing SVM

In addition to using SVD and PCA techniques to decompose the data, we sought to understand whether supervised learning techniques could be applied to distinguish between athletes. We decided upon using support vector machines (SVMs) for this classification task. Although SVMs are optimized for solving binary classification problems, we applied this method to a higher-dimensional problem space. SVMs aim to find the optimal decision boundary (in the form of a line or hyperplane) that separates the data points into different classes. This boundary not only separates the data, but also attempts to maximize the distance between the plane and the closest data points of each cluster. The steps utilized to perform SVM are as such:

```
procedure PERFORM SVM
    Create feature dataset X using force values at each time step
    Create label dataset y corresponding to jumper identity
    Split X and y into training and test sets using cross-validation
    Train the SVM on the training set
    Evaluate the model's predictions on the test set
end procedure
```

Our data was formatted in the following way: Each jumper had a separate CSV, labeled with their name. Each row represented one jump, with force measured across all 80 sensors at each timestep (of which there were 171). Each jump was labelled with a unique jump number. In order to process the data for SVM, each individual jumper's data was concatenated into a large dataframe. A column called `user_id` was added to distinguish between participants – this column contained a string displaying each jumper's name. Initially, the jumpers used were as follows: Annie, Caroline, Charles, Chu, Colin, Connor, Emma, Garrett, IanLim, Jake, Joshua, Matthew, Noah, Owen, Skylar, Sylvie, Leo.

Once Garrett's data had been removed, the new smallest class contained 5 samples, hence, we moved to 5-fold CV. For the last two trials, we decided to incorporate Owen's dummy data (i.e. the jumps without bending knees). This data appeared to be sufficiently different from the rest of the jumps, hence, our SVM should be able to classify it reliably.

2.5 Performing Low-Rank Approximation on Aggregated Jump Data

The jumps for each jumper are aggregated in a matrix where each row corresponds to an instance of a jump and the columns correspond to the aggregated sensor reading values at a particular timestamp in a matrix B. Jumps (i.e., rows) are also sorted in increasing jump height. We then normalize the matrix.

Then, SVD is performed to obtain the reduced SVD form

$$B_r = U_r \Sigma_r V_r^T$$

From there, we extracted the k-th rank 1 component of the reduced SVD and studied each of them by plotting a heatmap corresponding to the matrix. In other words, for

$$B_r = \sum_i \sigma_i u_i v_i^T$$

We looked at each $u_i v_i^T$. We conjecture that rank reducing and observing each of the terms in the SVD will allow us to observe particular features of the jump (example: the liftoff period, landing, takeoff patterns etc).

3 Results

3.1 Athlete Identification via Jump Signature Profiles. PCA

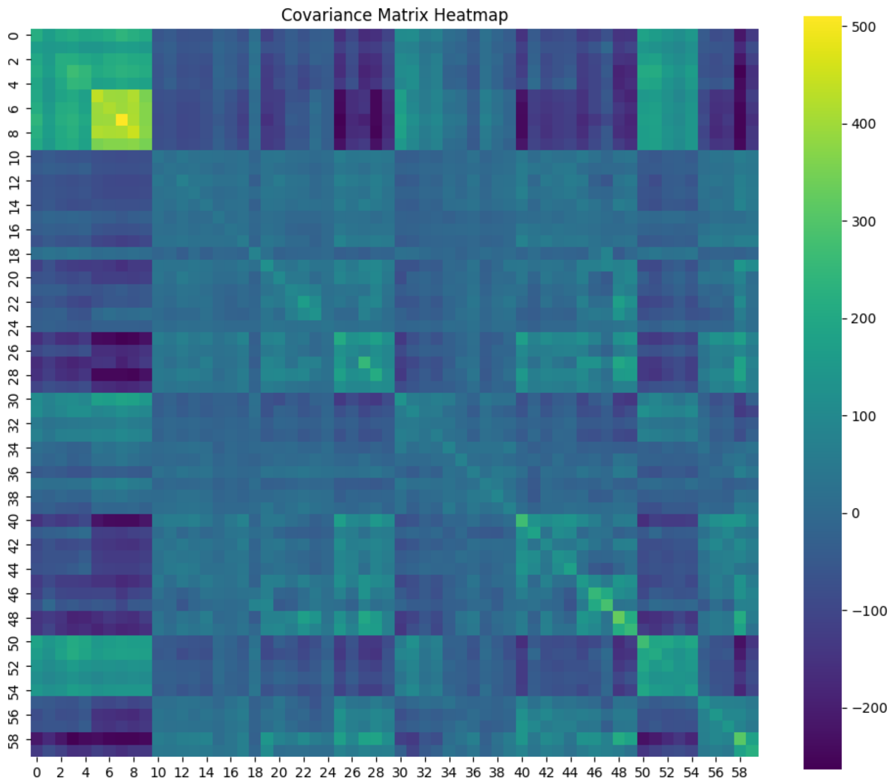

Figure 5: Covariance Matrix Heatmap

Figure 5 displays the covariance matrix BB^T heatmap, which represents the pairwise similarity between individual jumps, where each point corresponds to a single jump described by the aggregated force measurement of the 80 sensors.

The main diagonal is bright due to each jump being perfectly correlated with itself, while off-diagonal values show how similar different jumps are. Since we selected five jumps for each jumper, then distinct 5×5 blocks along the diagonal correspond to intra-jumper covariance — highlighting consistency or variability within each jumper's attempts. Bright blocks indicate consistent performance across a jumper's five jumps, while darker or more chaotic blocks suggest inconsistency. Patterns across blocks reveal similarity in performance across different individuals, while darker regions signal divergent jumping styles.

Specifically, the heatmap shows that most people jump similarly, as the blocks are mostly following the same gradient. This is to be expected, as many of our subjects were not athletes, and even amongst athletes the jumping vertical height did not differ substantially. However, we do observe that some blocks are much brighter than the others, implying those jumps are more correlated to themselves than to other jumps.

For instance, if we only use data for Owen's dummy jumps without using his knees, and Connor's jumps, then we observe the block structure in the covariance matrix becomes clear. The figure below displays this structure.

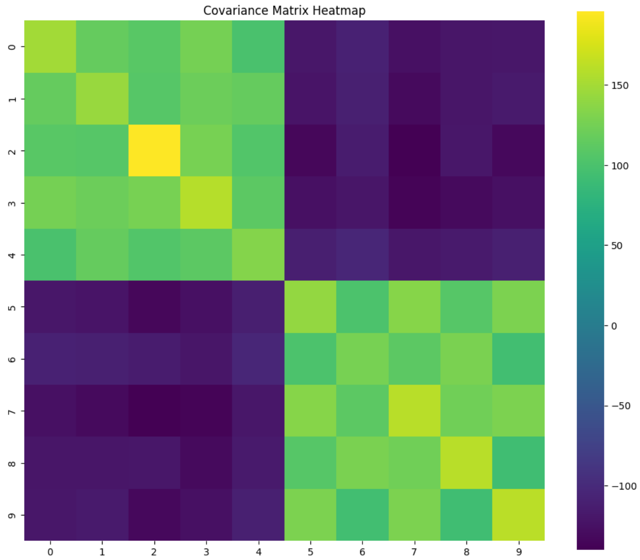

Figure 6: Covariance Matrix — Owen's dummy against Connor

3.1.1 PCA Plot:

To get the PCs, we extract the right singular vectors of B (the eigenvectors of $B^T B$), which are contained in the columns of V. To achieve this, we project the data onto the PCs through 'scores $= BV$. The result is a matrix scores that is of dimension 60×60. The (i, j) entry of scores gives the score of the i-th jump on the j-th PC.

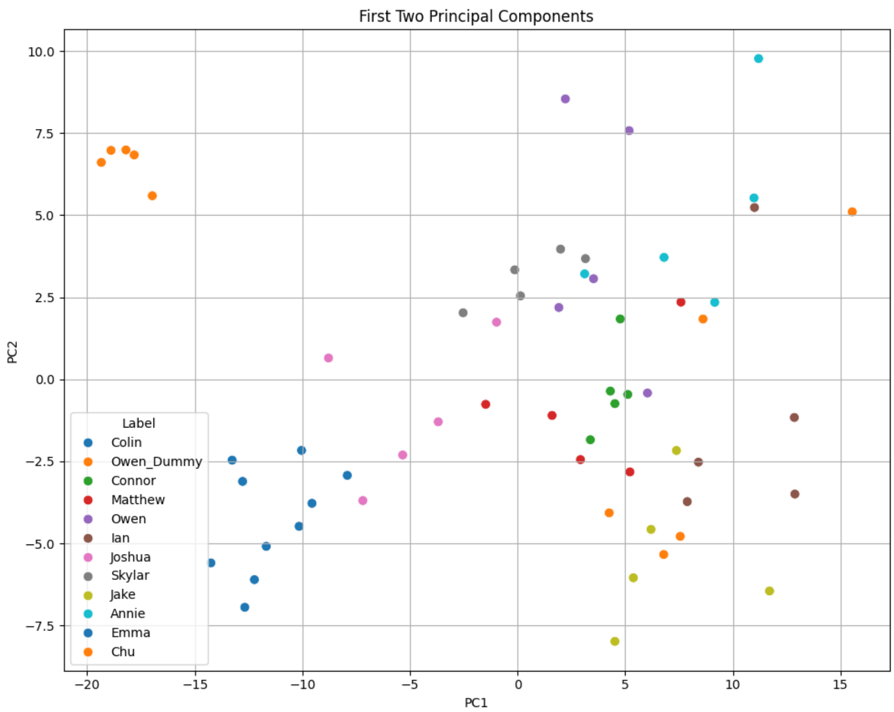

Figure 7: Plot of PC1 and PC2

Each PC represents a specific mode of variation across all jumps, with PC1 capturing the strongest pattern of difference and PC2 capturing the next most significant orthogonal variation.

Since each jump is a high-dimensional time series, these PCs serve as basis "jump patterns" that real jumps approximate to varying degrees. The score of a jump on a given PC reflects how strongly that jump expresses the corresponding pattern. Because PC1 explains over 50% of the variance, most of the variability between jumps can be attributed to a single dominant difference, while PC2 adds further nuance.

From the figure above, some jumpers (e.g. Colin with Emma, Owen_Dummy, and Jake) form tight, distinct clusters, indicating very consistent jump patterns across their jumps, and strong separation from others in the main modes of variation. This means their jumps follow unique, identifiable force-time profiles. For jumpers like Connor, Skylar, or Chu, the points are more spread out or overlap with other jumpers. This could reflect intra-jumper variability (inconsistent jumping technique) or similarity to other jumpers' styles.

The strong spread along the PC1 axis suggests that this component encodes the primary difference in jump types. Since Owen_Dummy is at the left-most side, and Jake is at the right-most side, PC1 could represent whether a jump has an early or late force peak. Hence, the PCA is successful because it captures a substantial portion of the total variance in just the first two components while also revealing clear structure in the data — indicating that PCA effectively distinguishes between significantly different jump styles. The separation along PC1 suggests a dominant timing-based difference across jumps, while PC2 adds a second, meaningful layer of variation, like jumping symmetry.

3.2 Teammate Comparisons with PCA

A specific group of interest were the three Field athletes who have similar training regimens, schedules, and lifestyles: Chu, Emma, and Jake. The following displays their movement vectors:

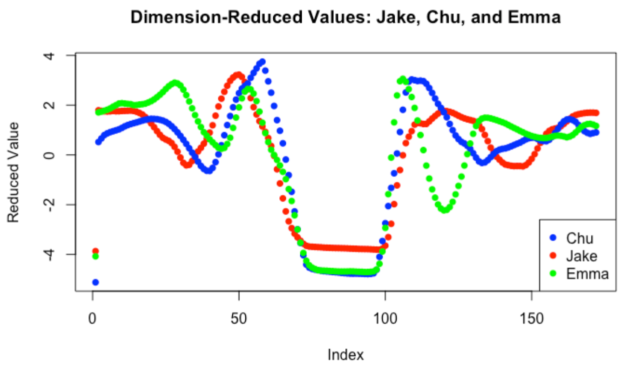

Figure 8: Display of the Throwers' and Connor's vectorized movements

We can easily see similarities present between these three individuals' jump patterns, especially on takeoff and ascent. PCA here has dimension reduced our jump data into a "one jump sample" that allows us to compare with other individuals much more easily.

It is extremely interesting to analyze how these teammates have very similar jump signatures when analyzing the shape and activity present here (especially in takeoff and flight).

When analyzing these three teammates compared to individuals of other athletic specialties, the differences are quite noticeable. Jake had the highest average vertical of the jumper group;

the next highest was Connor (powerlifter). Despite both of them having high verticals, there are still differences between Connor and the Throwers' jump profiles.

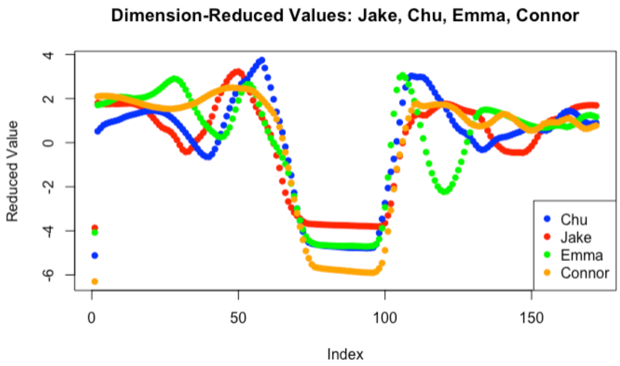

Figure 9: Display of Throwers' vs. Connor's vectorized movements

Especially when we look at takeoff, the sinusoidal action in the Throwers' activity compared to Connor's smoother jump up displays differences in usage of force.

There are also stark differences between the jump profiles of the Throwers to Owen's dummy data.

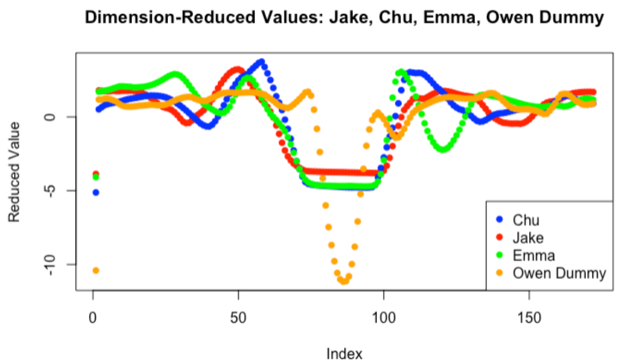

Figure 10: Display of Throwers' vs. Owen's Dummy Data vectorized movements

The differences are pretty apparent with all forms of metrics here - making sense as there should be ways for the data to represent itself well even with dimension reduction.

3.3 Athlete Classification via SVM

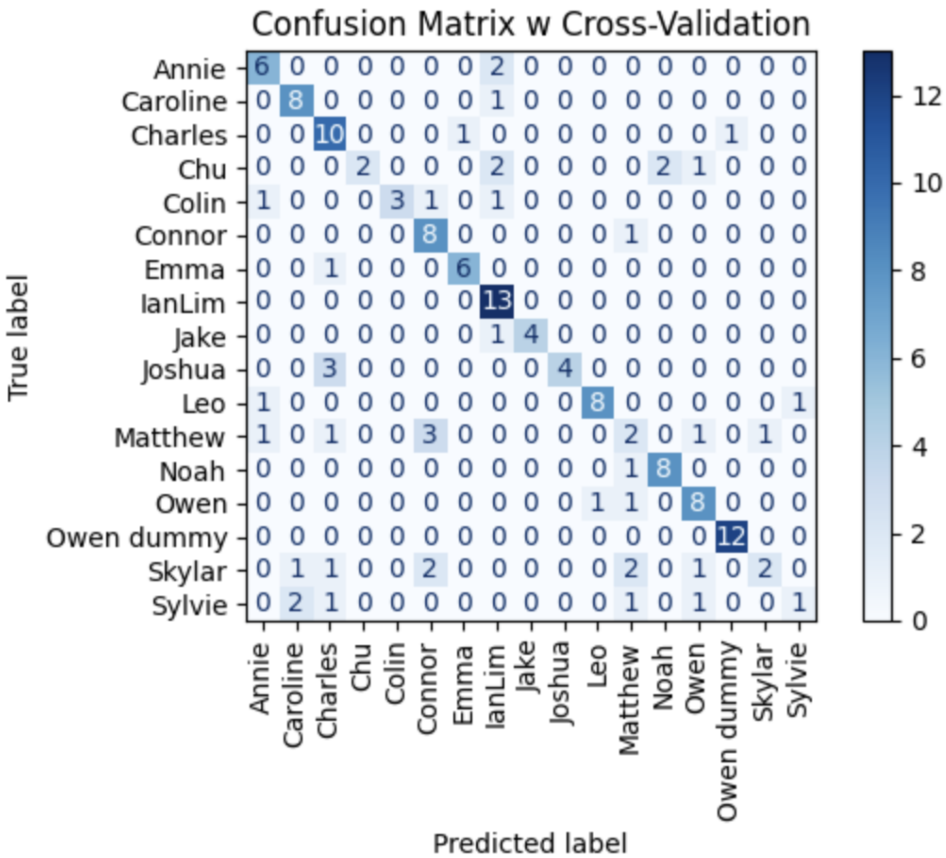

Figure 11: SVM Confusion Matrix

The figure above displays the confusion matrix from our best attempt, with an accuracy of $\approx 70.9\%$. Refer to the Appendix for an in-depth description.

As illustrated in the figure, the SVM classifier demonstrates strong overall discriminative power, with the vast majority of instances correctly assigned to their true classes (e.g. Charles: 10/12, IanLim: 13/13, Owen_Dummy: 12/12). Diagonal entries dominate the matrix.

3.4 Interpretations of SVD

In this section, we aim to understand the jumps by performing SVD and analyzing low-rank approximations of the full data matrix. We store our data by aggregating each jump into one matrix to extract common patterns/properties underlying each jump.

3.4.1 SVD on Aggregated Jump Data

For the first term, $u_1 v_1^T$ of B_r, the heatmap is as follows:

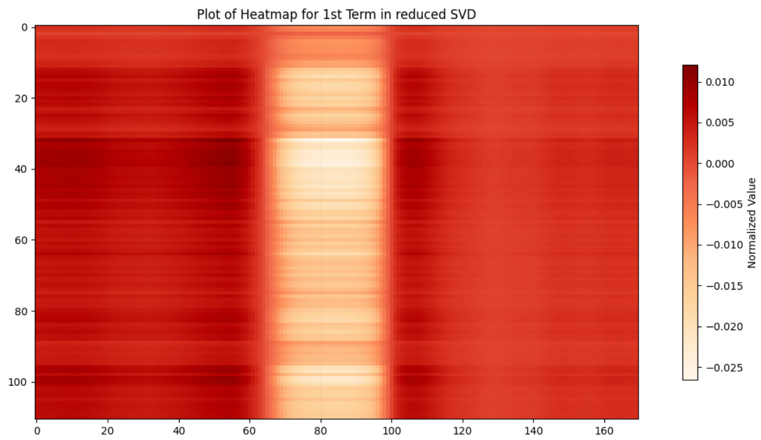

Figure 12: Heatmap for $u_1 v_1^T$

The most prominent feature of the jump is the liftoff as the sensor readings change the most before and after liftoff. Thus, we would expect the best rank-1 approximation of our jumps to be a matrix that prominently includes information about the liftoffs. Notice the dominantly yellow vertical band in the center of the heat map. This corresponds with the aligned extracted data centering at the liftoff region.

In the first term of the SVD, we can clearly see the captured liftoff phases where areas of lighter color correspond to lower sensor readings which indicate the jumper was in the air.

In addition, notice the deep red vertical banding immediately before and after the liftoff event. This may suggest particular similarity among the last moments of the propulsion phase and the first shock absorption instance of the landing phase.

Looking at the second term, $u_2 v_2^T$ yields the following heatmap:

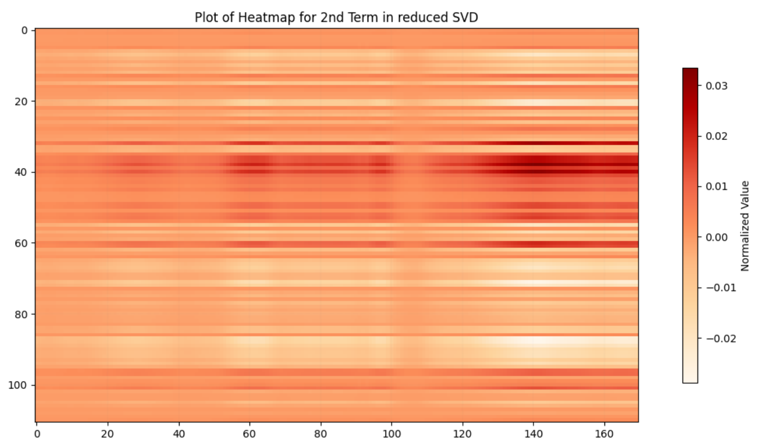

Figure 13: Heatmap for $u_2 v_2^T$

This second term captures the next prominent feature of all jumps. Here, notice the more prominent horizontal banding. This rank 2 approximation seems to adjust for particular abnormalities among individual jumpers. Notice in the PCA plots above, a few particular outliers in the PC2 direction. The strong row banding of this heat map seems suggest that $u_2 v_2^T$ is an adjustment for individual jump signatures.

4 Conclusion

In this study, we applied linear algebraic techniques—specifically singular value decomposition (SVD), principal component analysis (PCA), and support vector machines (SVM)—to analyze high-dimensional ground reaction force (GRF) data collected during countermovement jumps on a tactile sensor array. By representing each jump as a structured matrix of spatial-temporal force data, we demonstrated that individuals exhibit consistent and distinguishable "jump signatures," which can be extracted, visualized, and classified with high fidelity.

Our results show that dimensionality reduction techniques can uncover the dominant modes of variation in jump mechanics across individuals, and that these reduced representations retain sufficient discriminatory power for effective athlete identification. Additionally, we found that SVD-based noise filtering and anatomical interpretations of sensor activations can reveal biomechanical insights such as takeoff asymmetry, peak force location, and foot-loading strategy. Classification with SVM further demonstrated that these jump profiles are not only unique, but also learnable with high accuracy.

Beyond the results of this study, our work proposes a generalizable analytical framework for quantifying human movement through high-dimensional sensor data. As sensor-integrated environments become increasingly embedded in athletic training contexts, this framework provides a foundation for continuous monitoring of movement patterns over time. By capturing an athlete's evolving jump signature longitudinally, future studies may address open questions such as: What differentiates the movement profiles of elite soccer midfielders and basketball point guards? How do movement signatures change with development, fatigue, or injury? What latent characteristics of movement are predictive of future performance or risk?

This work lays the groundwork for such inquiries by formalizing a data-driven, mathematically principled approach to characterizing and comparing athletic motion.

5 Appendix

5.1 Github Repository

A link to the repository can be found at: https://github.com/Math-232-Project

5.2 Data Processing

Below we provide a general description of the data processing and cleaning procedure for the jump data:

To view the code, including that of the PCA, SVM, and the rest of the project, please refer to the submitted Python files in the repository.

5.3 PCA

Screeplot:

We plotted a screeplot to visualize the cumulative explained variance of each PC — since we had 60 jumps (5 for each of the 12 subjects), there are 60 PCs.

procedure ConvertAndPlot(filePath)
    Read log file and extract timestamps and sensor readings
    Identify release and landing events, compute flight times
    Filter valid jumps based on thresholds
    Mark in-flight periods in data
    Aggregate sensor values (sum, mean, etc.)
    Drop low-mean sensors, clear invalid jump sets
    Extract time windows centered on jump midpoints
    Transpose and label each jump cycle
    Save jump cycles to CSV
    Compute cosine similarity between jumps
    Filter and plot high-similarity cycles
end procedure

$\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$ is the diagonal matrix of singular values from the SVD $B = U\Sigma V^T$, where B is the mean-centered data matrix we calculated previously. Since the singular values represent the standard deviations associated with each Principal Component (PC), the total variance in the data is given by the sum of the squared singular values:

$$\text{Total Variance} = \sum_{j=1}^{r} \sigma_j^2.$$

The variance explained by the j-th principal component is σ_j^2, so the **explained variance ratio** for the j-th component is:

$$\text{Explained Variance Ratio}_j = \frac{\sigma_j^2}{\sum_{k=1}^{r} \sigma_k^2}.$$

This ratio quantifies the proportion of total variance in the data captured by each principal component, and is useful for determining how many components to retain in PCA.

From the figure, the first two PCs explain over 60% of the total variance. After PC2, the explained variance drops rapidly, forming a classic "elbow" shape. This suggests that PC1 and PC2 together capture the vast majority of the structure in the data, and most remaining components contribute little individually — supporting dimensionality reduction down to 2 PCs for visualization is appropriate.

5.4 SVM

At first, SVM was performed in the traditional method, using an 80/20 train/test split. However, we quickly realized that because our data contained only ≈ 10 jumps per person, splitting the data this way would only allow for ≈ 2 testing points. This could lead to unstable performance that depends largely on the partitioning within the data. Additionally, a rigid split such as this might accidentally underrepresent some classes, especially because our data was not originally balanced. Thus, we decided to use Stratified K-Fold cross validation. This method trains and validates the model on different subsets of the data, while preserving the distribution of classes at each fold.

When utilizing Stratified K-Fold cross validation, we wanted to ensure that each class (i.e. each unique jumper) is represented in every fold. Hence, this means that the maximum number of

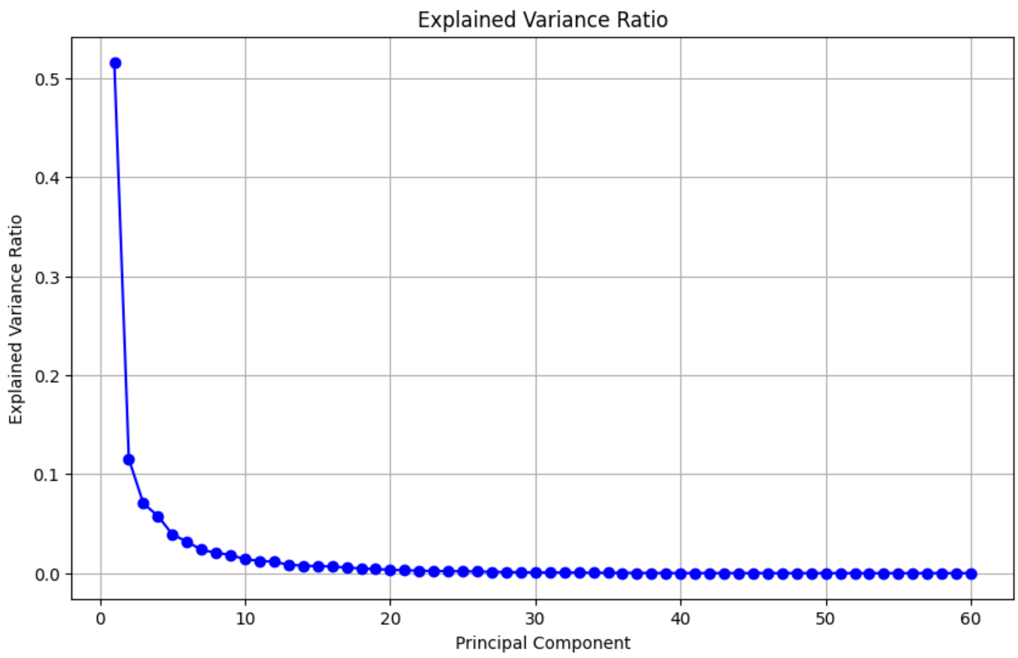

Figure 14: Principal Component's Screeplot

folds cannot exceed the number of samples in the smallest class. Let's say the smallest class — Ian, for example — contained only 2 samples. If we were to select $k > 2$ folds, then Ian would not be represented in all of the folds, which would lead to invalid training and evaluation. Thus, we identified that the number of samples in the smallest class was 3, corresponding to Garrett's jump data. However, in such a small dataset, we want to maximize the number of folds. With more folds, a larger proportion of the data is utilized for training in each iteration, as each data point is used for both training and validation (in separate folds). Hence, we decided to exclude Garrett's data to use the number of samples in the next-smallest class (5 samples). Below is a table of some of the split type and feature combinations, along with the resulting accuracy.

Split	**Features**	**Accuracy**
80/20 train/test split	All jumpers	0.46
3-fold SKF CV	All jumpers	0.6115
3-fold SKF CV	All jumpers + engineered features	0.6402
5-fold SKF CV	Jumpers - Garrett	0.6617
5-fold SKF CV	Jumpers - Garrett + engineered features	0.6691
5-fold SKF CV	Jumpers - Garrett + Owen Dummy	0.7027
5-fold SKF CV	Jumpers - Garrett + Owen Dummy + eng features	0.7094

Table 2: SVM Attempts

In order to incorporate cross validation, we wrote a pipeline that first normalized the data using `StandardScaler()`, then passed it through an SVM using the Radial Basis Function (RBF) kernel. This kernel is similar to the Gaussian kernel, and helps increase the performance of the SVM on non-linear classification tasks.

Our initial attempt with cross-validation saw a large increase in accuracy as compared to a normal 80/20 train/test split. Then, we experimented with adding engineered features in order to boost the predictive power of the SVM. The engineered features in question were a jump's maximum force (calculated as the maximum across each row) and total force (calculated as the sum of each row). We assumed that we would observe differences amongst athletes in these variables. With 3-fold Stratified CV, we observed the largest improvement in predictive accuracy when these engineered features were added.

5.5 Temporal Noise Reduction and Signal Smoothing through SVD

It has been shown that noisy sensor signals could also be smoothed via singular value decomposition[1]. To demonstrate this, a noisy signal was generated by taking capacitive touch measurements of a free hanging electrode, periodically interrupted by a touch event. The resulting data output was a vector R in t where t is the number of reading taken over the time horizon. To perform SVD on this data, we mapped the $1 \times t$ vector into an $m \times n$ matrix using a bijective transformation. The resulting matrix A_{ij} can be calculated as follows

$$(\mathbf{A})_{ij} = x_k : \begin{cases} i = 1 + (k-1) \bmod m \\ j = 1 + (k-1) \div m \end{cases}$$

After the matrix decomposition is complete, it can be then unraveled back into the 1 x t vector via the following reverse operation

$$x_{p,k} = (\mathbf{A}_p)_{ij} \quad : \quad k = i + m \cdot (j-1).$$

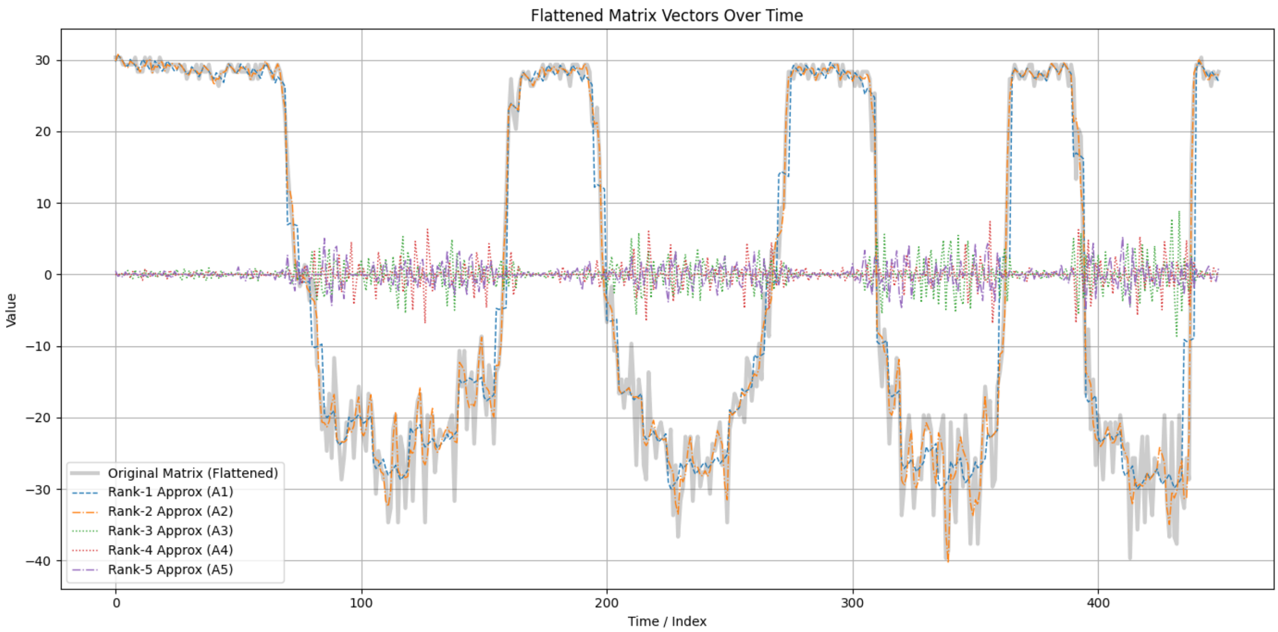

Figure 15: SVD Signal Processing

As shown in the plot, the rank one approximation of this time series data provides a smoothed version of the raw signal, while still preserving the meaningful fluctuations caused by the touch event.

It was observed that varying dimensions of the bijectively mapped A matrix affect the signal approximation. For the plot below, $t = 450$, and the matrix A was chosen to have dimensions 5×90, where $m = 5$ was chosen to be the smallest possible factorization of 450 other than 1×450. These oblong dimensions provided the best rank one approximation. Perhaps this is because these dimensions yielded the smallest $\dim(\Sigma)$. Fewer singular values perhaps force most of the variation in the data to be stored in the $u \cdot v^T$ (column $\times$ row) vectors, and, ultimately, $u_1 \cdot v_1^T$ is what is used as the approximation. The lower rank approximations extract much of the noise in the raw signal.

In practice, this noise reduction technique may be used in data pre-processing to support touch and release event flagging without any time delay. Moving averages introduce some lag for high acceleration events; this event may avoid this.

5.5.1 Spatial Noise Reduction through SVD

We want to study one specific instance of a jump using SVD. In this example, we are using one specific jump from Collin's data. We do so by first identifying critical timestamps of the jump, and loading the sensor reading values for each critical timestamp into a 8×10 matrix corresponding to the actual orientation of the sensors. We then perform SVD.

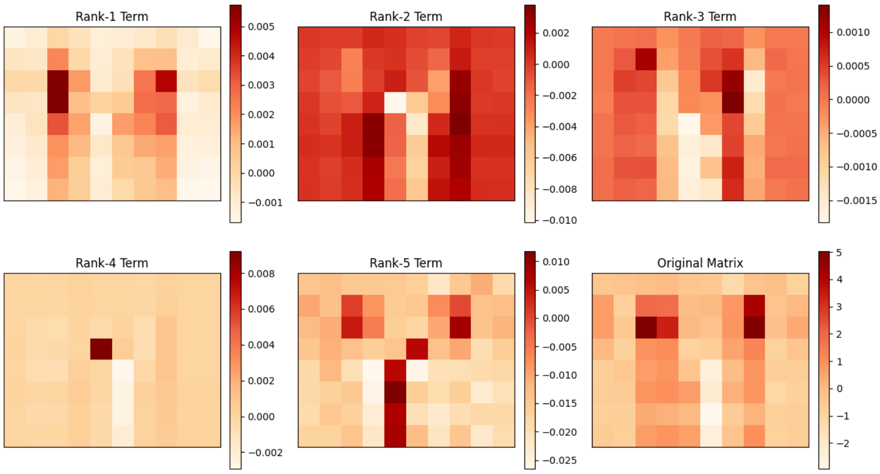

Figure 16: Sensor Readings for Liftoff

From the original matrix in the figure above, we notice clearly that this jumper is used to using their toes for lifting off the ground as seen from stronger sensor activations at the sensors near the toes. This observation is much clearer if we look at the rank-1 approximation for this matrix. In this case, SVD acts as noise filtering to allow us to identify important characteristics of the jump that might not be apparent just by observing the original matrix.

5.6 Linear Regression on Assorted Metrics

We were also interested in any possible linear relationships between individual's jump height and their physiology. Every jumper had their calf length and overall leg length measured to compare calf:leg ratio and calf length with average jump height.

From class, we know that calculating linear regression is a fundamental statistical technique used to model the relationship between a dependent variable y and one or more independent variables x. In the case of simple linear regression, the model assumes a linear relationship of the form:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

where β_0 is the intercept, β_1 is the slope coefficient, and ε is a random error term assumed to be normally distributed with mean zero and constant variance.

The objective of linear regression is to estimate the parameters β_0 and β_1 by minimizing the sum of squared residuals:

$$\min_{\beta_0,\beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

The resulting line of best fit can then be used for prediction or to understand the strength and direction of the relationship between the variables. The Pearson correlation coefficient r is often used alongside linear regression to quantify the linear association between x and y.

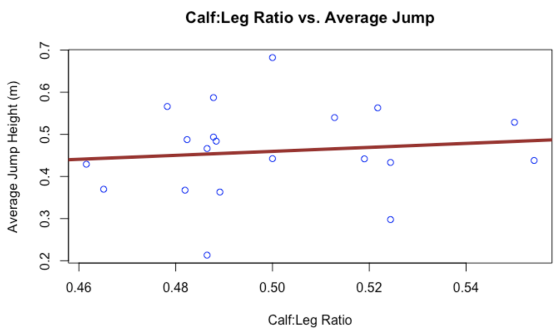

Figure 17: Calf:Leg Ratio vs. Average Jump, r = 0.1129

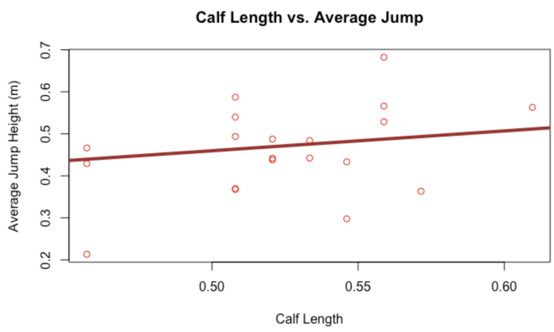

Figure 18: Calf Length vs. Average Jump, r = 0.3920

We can see that the data doesn't show any staggering findings. Although there is a slightly positive trend, it is not an overwhelmingly obvious one. There are high jumpers in both plots around the middle of the pack that deviate from possible linear trends. This suggests that our data is not linearly related. Upon further inspection, however, we can see the trend that these variables follow:

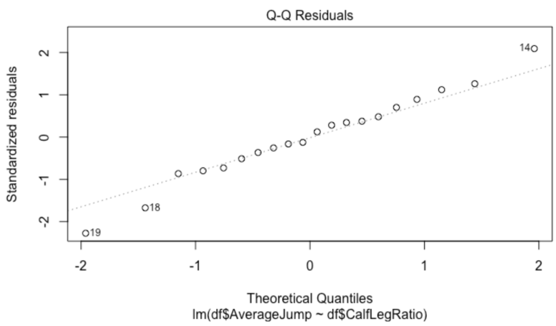

Figure 19: Calf:Leg Ratio vs. Average Jump QQ Plot,

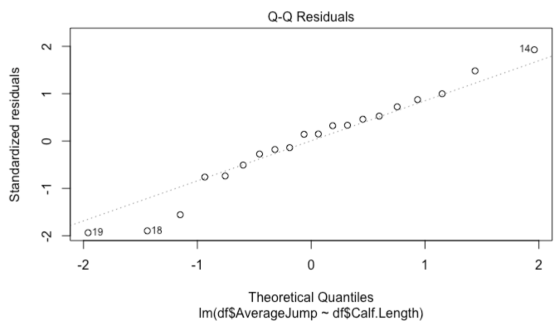

Figure 20: Calf Length vs. Average Jump QQ Plot

As both QQ plots appear to be approximately linear, this suggests that our data are actually approximately normally distributed.

References

[1] Thomas Schanze. Removing noise in biomedical signal recordings by singular value decomposition. *Current Directions in Biomedical Engineering*, 3(2):253–256, 2017.

[2] Vault Kinetics. Vault kinetics: Ai-powered athletic performance analytics. https://www.vaultkinetics.com/, 2025.