Up in the Air: An Analysis of Prediction, Skill, and Luck in Men's College Basketball

Triangle Competition in Math Modeling — 2024

Avantika Chopra, Tanish Kumar, Shambhavi Sinha

November 17th, 2024

1 Abstract

In NCAA Division I Men's Basketball, where skill and luck interplay dramatically shape outcomes, accurately predicting game results presents both a statistical challenge and an opportunity. This study develops a model to predict game outcomes by quantifying the relative contributions of skill and chance, leveraging machine learning techniques to capture this complex dynamic. We use a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel, Random Forest classifiers, and Principal Component Analysis (PCA) for dimensionality reduction, combined with a 10-game moving average for performance smoothing. Our model also employs a luck metricization method, based on the R factor, to assess the balance between consistent skill-based and random luck-driven results.

2 Introduction

2.1 Relevance

The question of human competence and ability has always been muddled with the impact of luck. When we reflect on human activities, a lot is determined by social values, moral boundaries, biological regulations and physical considerations. Coupling this with the notion of chance events leads to a massive increase in the variation of possible events. Given this large variation of possible events, it has been of consistent interest to researchers and lay people alike, to predict and assign causality and distinction in skill and luck.

In trying to assess this critical question of luck versus skill, arises many challenges. Notions of ethics, personal beliefs of success and its factors and profitability are some key challenges that have plagued the sphere of athletic success. We seek to attempt to illuminate one subset of these questions, by predicting the NCAA Division I Men's Basketball performance using a mathematically rigorous and statistical approach.

The NCAA Division I Men's Basketball Tournament featuring 68 teams compete with countless surprises and upsets defying predictions. While some outcomes can be attributed to skillful play, others appear to be driven by luck or even randomness. This paper addresses how to quantitatively assess the roles of skill and luck in shaping the outcomes of games within the tournament. US Bettors spent \$2.72 billion on this year's tournament, around 55% more than what was wagered on the 2024 Super Bowl. With extensive financial investments from individuals, it would be worthwhile to have a model that predicts outcomes more effectively than standard ranking and rating systems and uses only basic input data. In this paper, we present such a model.

2.2 Background

Before diving into the specifics of the model, it is useful to outline the structure of the NCAA Men's Basketball Tournament for those less familiar with the event. Each year, after the regular college basketball season concludes, the NCAA organizes a high-stakes 64-team tournament. This prestigious event includes two main groups of teams: the champions from each of the 31 Division I conferences, and additional teams selected based on overall season performance, which are chosen by the NCAA tournament selection committee.

The selection committee not only picks these remaining teams but also assigns a ranking or "seed" to each of the 64 teams, arranging them within four distinct regions. Each region contains 16 teams ranked from #1 to #16 based on their perceived strength. The top four teams in the country, as judged by the committee, each receive a #1 seed in one of the four regions, while the next four teams receive the #2 seeds, and so on. Within each region, the 16 teams compete in a single-elimination format over four rounds, with matchups organized so that higher-seeded teams face lower-seeded ones initially (e.g., the #1 seed faces the #16 seed, the #2 seed faces the #15 seed, etc.). The last team standing in each region earns a spot in the Final Four. These four teams then advance to a two-round single-elimination tournament to vie for the national championship. Importantly, all games throughout the tournament, including the Final Four, are held at neutral venues, which helps ensure that no team benefits from a home-court advantage.

In the context of tournament pools, which are popular for wagers and casual predictions, participants fill out their entire bracket before the tournament begins, predicting the winners of each game up through the championship. A unique challenge in these pools is that late-round matchups may include teams that participants did not initially predict to advance, which can disrupt bracket predictions as the tournament progresses.

To aid in predicting outcomes, participants often turn to various ranking and rating systems. Popular sources include the Associated Press poll, which reflects the opinions of sportswriters; the ESPN/USA Today poll, based on coach rankings; the Ratings Percentage Index (RPI), which combines team winning percentages with those of their opponents; the Sagarin ratings, published in USA Today; the Massey ratings; Las Vegas betting odds; and the tournament selection committee's official seeding. Each of these systems provides a different perspective on team strength, offering a range of insights for making predictions.

2.3 Our Approach

In this study, we aim to develop a mathematical model that can accurately predict the winning team in NCAA basketball games by capturing a mix of skill and luck factors. Our approach combines various machine learning techniques, including Support Vector Machine (SVM) with a radial basis function (RBF) kernel, random forest classifiers, and dimensionality reduction through Principal Component Analysis (PCA). To further enhance predictive accuracy, we also incorporate moving averages, calculated over the last 10 games, for certain game metrics, which helps smooth out recent performance trends and reveals underlying team strengths.

1. Support Vector Machine (SVM) with RBF Kernel: The SVM model with an RBF kernel is used to capture complex, non-linear relationships between features, allowing us to differentiate teams based on a blend of skill-based attributes (e.g., shooting percentages, assist-to-turnover ratios) and luck-based variables (e.g., turnovers, free throw performance). The RBF kernel is particularly suited for this purpose as it can adapt to variations in data distribution, which is essential when modeling the inherent unpredictability of sports outcomes.

2. Random Forest Classifier: A random forest model is employed to capture interactions between features and provide an ensemble-based perspective on game prediction. This model aggregates predictions from numerous decision trees, each trained on different subsets of data, which helps reduce overfitting and improve robustness. By using random forests, we can identify which skill and luck factors most influence game outcomes, based on feature importance scores, and adjust our predictions accordingly.

3. Principal Component Analysis (PCA): Given the high dimensionality of basketball statistics, PCA is applied to reduce feature space complexity while retaining the most significant variance-driving components. This step allows us to distill key information from numerous factors (e.g., points in the paint, defensive rebounds, fast break points) and helps our models focus on the core skill-based and luck-related components affecting game results.

4. Moving Averages of 10 Games: To smooth out fluctuations and highlight consistent trends, we apply moving averages over the last 10 games for select statistics. This approach helps our model account for short-term form while filtering out extreme, one-off performances, thus striking a balance between recent team momentum and overall season performance.

By combining these methods, our model integrates a range of skill-based and chance-related metrics, allowing us to predict game outcomes more accurately. This blended approach not only provides insights into each game's outcome but also quantifies the relative contributions of skill and luck, offering a comprehensive view of team dynamics.

3 Theoretical Basis

3.1 Support Vector Machine

3.1.1 Introduction to Support Vector Machines (SVMs)

Support Vector Machines (SVM) are a class of supervised machine learning algorithms widely used for classification and regression tasks. Developed by Vapnik and Cortes in the 1990s, SVMs are particularly effective in handling high-dimensional and complex datasets. They operate by constructing an optimal hyperplane in the feature space that maximizes the margin between classes, thus enhancing the model's generalization ability on unseen data. In practice, an SVM aims to find a decision boundary that separates different classes with the greatest possible margin, defined as the distance from the boundary to the nearest data points, known as support vectors. This margin maximization framework not only enables a robust decision function but also enhances the classifier's tolerance to slight variations in the data.

However, real-world datasets often exhibit non-linear relationships, where a simple linear separation is insufficient. To address this limitation, SVMs incorporate the concept of kernel functions, which allow for non-linear decision boundaries by implicitly mapping data into higher-dimensional spaces. In this context, the Radial Basis Function (RBF) kernel, also known as the Gaussian kernel, is particularly prominent due to its flexibility and adaptability to complex data patterns.

3.1.2 The Role of the Radial Basis Function (RBF) Kernel in SVMs

The RBF kernel is a popular choice for non-linear SVMs because it transforms data into a higher-dimensional space where linear separation becomes feasible. Mathematically, the RBF kernel can be defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

where $\mathbf{x}$ and $\mathbf{x}'$ are two data points in the input space, $\|\mathbf{x} - \mathbf{x}'\|$ denotes the Euclidean distance between these points, and σ is a hyperparameter that determines the kernel's width. An alternative formulation uses the parameter γ, which is inversely proportional to the variance (σ^2), defined as:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$$

This kernel function computes the similarity between two points, with similarity decaying exponentially as the distance between points increases. The RBF kernel is especially useful because of its adaptability; it provides a localized response, where points close to each other in the feature space are assigned higher similarity scores than those further apart. This property is advantageous in distinguishing complex, non-linear patterns within the data, making it ideal for tasks with complex class boundaries.

3.1.3 Properties of the RBF Kernel

The RBF kernel function reaches its maximum value of 1 when $\|\mathbf{x} - \mathbf{x}'\| = 0$, implying that identical points have the highest similarity. As the distance $\|\mathbf{x} - \mathbf{x}'\|$ grows, the similarity measure $K(\mathbf{x}, \mathbf{x}')$ approaches zero, signifying dissimilarity between points. The variance parameter σ (or, equivalently, γ) plays a critical role in determining the "width" of the region of similarity:

1. When σ is large (small γ), the kernel function decreases slowly, and points at greater distances from each other are still considered relatively similar. This broadens the influence of each data point, enabling the decision boundary to be smooth and less sensitive to local data variations.

2. When σ is small (large γ), the RBF kernel function decays rapidly, causing only nearby points to be considered similar. This setting creates a highly localized decision boundary that can capture intricate structures in the data but may also risk overfitting if γ is too high.

The RBF kernel's behavior is mathematically akin to the Gaussian distribution, where the distance between data points effectively represents dissimilarity. In an SVM context, this allows the model to adaptively fit to non-linear boundaries by learning from highly similar points, enhancing both accuracy and robustness. This similarity-based transformation creates a decision boundary that is non-linear in the original space but linear in the transformed feature space.

Since the feature space induced by the RBF kernel is of infinite dimensionality, it is theoretically capable of representing any non-linear function. This property makes the RBF kernel highly expressive and suitable for datasets that exhibit non-linearity, as it can adjust to the local complexity of the data.

3.1.4 Hyperparameter Selection and Model Tuning

To leverage the RBF kernel effectively, it is crucial to optimize its associated hyperparameters, γ (or σ) and the regularization parameter C of the SVM. The parameter C controls the trade-off between maximizing the margin and minimizing classification errors, thereby adjusting the complexity of the model:

- Low values of C allow for a wider margin and greater tolerance for misclassified points, leading to smoother decision boundaries that may generalize better.
- High values of C penalize misclassifications more heavily, which can lead to a tighter fit to the training data and potentially overfit if not managed carefully.

Selecting an appropriate γ is essential to achieve a balanced model. A high γ value (small σ) results in narrower regions of influence, making the decision boundary sensitive to individual points and possibly leading to overfitting. Conversely, a low γ value (large σ) creates a broader region of influence that smooths out local variations and can improve generalization at the risk of underfitting.

3.2 Principal Component Analysis

3.2.1 Introduction to Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a statistical technique commonly used for dimensionality reduction in high-dimensional datasets. By transforming a large set of correlated variables into a smaller set of uncorrelated variables, PCA simplifies the dataset while preserving as much of its original variability as possible. This reduced set of variables, called principal components, allows for more efficient data exploration and modeling, particularly in cases where visualizing high-dimensional data or mitigating computational complexity is necessary.

The core objective of PCA is to capture the maximum variance in the data within a reduced number of dimensions, while minimizing information loss. This is achieved by identifying the principal directions—orthogonal axes—along which the data variance is maximized. In effect, PCA represents the data in terms of these principal components, which form a new coordinate system aligned with the directions of greatest variance.

3.2.2 Mathematical Foundation of PCA

PCA can be broken down into a series of mathematical steps that leverage concepts from linear algebra, particularly eigenvalues and eigenvectors, to determine the principal components of the data.

Step 1: Standardization

Given a dataset $\mathbf{X}$ with n samples and p variables, PCA first standardizes the data to ensure that all variables contribute equally to the analysis. This is necessary because PCA is sensitive to the scale of the variables; variables with larger ranges would otherwise dominate the principal components. Standardization is performed by centering each variable (subtracting its mean) and scaling by its standard deviation:

$$Z_{ij} = \frac{X_{ij} - \mu_j}{\sigma_j}$$

where Z_{ij} is the standardized value of variable j for sample i, μ_j is the mean of variable j, and σ_j is its standard deviation. The result is a transformed dataset $\mathbf{Z}$ in which each variable has a mean of zero and a standard deviation of one.

Step 2: Covariance Matrix Computation

To identify correlations between variables, PCA computes the covariance matrix of the standardized dataset $\mathbf{Z}$. The covariance matrix $\mathbf{C}$ is a $p \times p$ symmetric matrix, where each entry C_{jk} represents the covariance between variables j and k:

$$C_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (Z_{ij} - \bar{Z}_j)(Z_{ik} - \bar{Z}_k)$$

where $\bar{Z}_j$ is the mean of the standardized values for variable j. The diagonal elements of $\mathbf{C}$ represent the variances of each variable, while the off-diagonal elements indicate the covariance between pairs of variables. Variables with high covariance are highly correlated, implying redundancy in the data.

Step 3: Eigenvalues and Eigenvectors of the Covariance Matrix

The covariance matrix $\mathbf{C}$ captures the spread and orientation of the data. To identify the principal components, PCA computes the eigenvalues and eigenvectors of $\mathbf{C}$. An eigenvector defines a direction in the original feature space, and its associated eigenvalue indicates the amount of variance along that direction. Mathematically, the eigenvalues and eigenvectors are derived from the characteristic equation:

$$\mathbf{C}\mathbf{v} = \lambda\mathbf{v}$$

where λ represents an eigenvalue and $\mathbf{v}$ an eigenvector. For a p-dimensional dataset, this results in p eigenvalue-eigenvector pairs, which define the directions (principal components) and the magnitudes of variance along these directions. By ordering the eigenvalues in descending order, we prioritize components that capture the most variance.

Step 4: Feature Vector Construction

Once the eigenvalues are ordered from highest to lowest, we select the top k eigenvectors (associated with the largest eigenvalues) to form a feature vector $\mathbf{P}$. This matrix $\mathbf{P}$, with dimensions $p \times k$, represents the k principal components that capture the majority of the variance in the data. Retaining only the eigenvectors associated with the largest eigenvalues allows us to reduce the dimensionality while retaining the most informative components.

Step 5: Recasting the Data in the Principal Component Space

The final step in PCA is to project the original data onto the new k-dimensional principal component space. This is achieved by multiplying the standardized data matrix $\mathbf{Z}$ by the feature vector $\mathbf{P}$:

$$\mathbf{Y} = \mathbf{Z}\mathbf{P}$$

where $\mathbf{Y}$ is the transformed dataset in the principal component space. Each row of $\mathbf{Y}$ represents the projection of a sample along the principal components, providing a compressed representation of the data in k-dimensions.

3.2.3 Interpretation of Principal Components

The principal components are linear combinations of the original variables, oriented in the directions of maximum variance. Importantly, these components are uncorrelated, meaning they capture distinct aspects of the data's variability. The proportion of total variance explained by each principal component is given by the ratio of its eigenvalue to the sum of all eigenvalues:

$$\texttt{Variance Explained by}\mathrm{PC}_j = \frac{\lambda_j}{\sum_{i=1}^{p} \lambda_i}$$

By plotting the variance explained by each principal component (often visualized as a scree plot), we can identify how many components are needed to capture a desired level of information, typically allowing us to discard components with low eigenvalues and thereby reduce dimensionality.

3.2.4 Practical Considerations and Applications

PCA is widely used in machine learning, especially for preprocessing high-dimensional data before applying further analysis or machine learning algorithms. By reducing the dimensionality, PCA minimizes noise and computational costs, often improving model performance and interpretability. However, it is important to note that the principal components lack direct interpretability in terms of the original variables, as they are synthetic axes that optimize variance capture rather than representing specific measured features.

4 Random Forests Classifier

4.0.1 Introduction to Random Forests

Random Forests, introduced by Breiman (2001), are an ensemble learning technique that leverages the power of multiple decision trees to improve classification and regression accuracy. The approach combines the principles of bagging (bootstrap aggregation) and random feature selection, which together reduce overfitting and enhance generalizability. Each tree in a random forest is trained on a bootstrapped sample of the data and splits nodes using a random subset of features, resulting in a "forest" of diverse, decorrelated trees.

4.0.2 Mathematical Foundation of Random Forests

Random Forests operate by constructing multiple decision trees and averaging their predictions. For classification tasks, the final output is based on a majority vote across the trees, while for regression tasks, it is the average prediction. The following steps outline the mathematical process:

Step 1: Bootstrapping and Building Individual Trees

Given a dataset $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ with N samples and p features, Random Forests create B trees by: 1. Bootstrapping: Generating B bootstrap samples $\mathbf{D}_b^*$ (with replacement) from the original dataset. 2. Random Feature Selection: For each node split in tree b, a random subset of m features (where $m \ll p$) is selected as candidates for the split, reducing correlation among trees.

Each decision tree T_b is constructed by recursively selecting the best feature and split point from the m candidate features to maximize an impurity reduction criterion, such as the Gini index for classification or mean squared error (MSE) for regression.

Step 2: Aggregating Tree Predictions

For a new input $\mathbf{x}$, the Random Forest prediction is derived by aggregating the predictions from all B trees.

- For Classification: Each tree T_b provides a class prediction $C_b(\mathbf{x})$. The final prediction $\hat{C}(\mathbf{x})$ is the majority vote:

$$\hat{C}(\mathbf{x}) = \text{mode}\{C_b(\mathbf{x})\}_{b=1}^B$$

- For Regression: Each tree T_b outputs a predicted value $T_b(\mathbf{x})$. The final prediction $\hat{f}(\mathbf{x})$ is the average of all tree predictions:

$$\hat{f}(\mathbf{x}) = \frac{1}{B}\sum_{b=1}^{B} T_b(\mathbf{x})$$

Step 3: Variance Reduction through Averaging

The primary advantage of Random Forests is the reduction in variance achieved by averaging the predictions of many decorrelated trees. Given a single tree's variance σ^2 and a pairwise correlation ρ among trees, the ensemble variance for a Random Forest with B trees is given by:

$$\text{Var}(\hat{f}_{\text{rf}}(\mathbf{x})) = \rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

As B increases, the second term diminishes, but the first remains. Thus, decorrelating trees (reducing ρ) is crucial for effective variance reduction, which is achieved by random feature selection at each node split.

4.0.3 Important Features of Random Forests

Out-of-Bag (OOB) Error Estimation

Random Forests naturally provide an out-of-bag (OOB) error estimate, which approximates the model's performance on unseen data. Since each tree is trained on a bootstrap sample, approximately one-third of the data is left out of each sample. These OOB samples serve as a validation set for that tree, and averaging the OOB error across all trees provides an unbiased estimate of the Random Forest's test error.

Variable Importance

Random Forests estimate the importance of each feature by calculating the decrease in prediction accuracy or impurity when that feature is removed or permuted. For a feature j, its importance score is computed by the average decrease in the Gini index (for classification) or in MSE (for regression) when used in splits across all trees in the forest.

4.1 Moving Averages

4.1.1 Introduction to Moving Averages

A moving average is a commonly used method for smoothing time series data. Moving averages are particularly valuable in highlighting long-term trends by filtering out short-term fluctuations. This technique is applicable across various fields, including economics, finance, and signal processing, where it is often employed to examine temporal data and eliminate high-frequency noise. Moving averages come in several forms, including the simple moving average (SMA), weighted moving average, and exponential moving average, each differing in how data points within the subset are weighted.

4.1.2 Simple Moving Average (SMA)

The Simple Moving Average (SMA) is the most basic form of moving average and calculates an unweighted mean over a fixed number of past observations, referred to as the "window" or "subset size." This average is continuously recalculated as the window moves forward over time, effectively creating a "running" or "rolling" average that responds to the latest data while discarding older points. The result is a smoother time series that reveals trends by minimizing random fluctuations.

The SMA for a time series $\{p_i\}_{i=1}^{n}$ with a window size k is defined as follows:

$$\mathrm{SMA}_k = \frac{1}{k} \sum_{i=n-k+1}^{n} p_i$$

where: - p_i represents the individual data points in the series,
- k is the fixed subset size or window length,
- n is the index of the last data point in the current window.

The value of k determines the period over which the average is computed. A smaller k results in a more sensitive SMA that closely follows recent data points, whereas a larger k produces a smoother SMA that captures long-term trends.

4.1.3 Mathematical Properties of SMA

The SMA can be viewed as a convolution operation, where the time series is convolved with a rectangular window function (boxcar filter) of equal weights, each equal to $\frac{1}{k}$. This has implications for its behavior in frequency domain analysis, as the SMA acts as a low-pass filter, attenuating high-frequency components and passing lower frequencies.

4.1.4 Recursive Calculation of SMA

To efficiently compute the SMA in a real-time context, the next SMA value can be derived from the previous SMA value by accounting for the addition of a new data point and the removal of the oldest data point in the window. Given a current SMA value $\text{SMA}_{k,\text{prev}}$, the next SMA value $\text{SMA}_{k,\text{next}}$ can be updated using:

$$\text{SMA}_{k,\text{next}} = \text{SMA}_{k,\text{prev}} + \frac{1}{k}\left(p_{n+1} - p_{n-k+1}\right)$$

where: - p_{n+1} is the new data point entering the window, - p_{n-k+1} is the data point exiting the window.

This recursive formulation allows for computational efficiency, requiring only three operations per update, rather than recalculating the sum from scratch.

4.1.5 Interpretation and Applications of SMA

The SMA serves as a smoothing tool, especially valuable in time series analysis to reveal trends over chosen periods. The selection of k, the window size, depends on the objective and the type of trends of interest. For instance:

- Short-term trends are observed with smaller k, providing higher sensitivity but potentially more noise.

- Long-term trends are observed with larger k, yielding a smoother curve with reduced short-term variability.

4.1.6 Limitations and Frequency Response of SMA

While the SMA is straightforward to implement, it has certain limitations. As a low-pass filter, the SMA attenuates high-frequency variations but does so imperfectly, sometimes allowing unwanted frequencies and introducing artifacts. Notably, the SMA has a sinc-shaped frequency response, which can introduce oscillations (known as the Gibbs phenomenon) when filtering periodic components that match the window length k. This effect can lead to unexpected artifacts, such as inversion of peaks and troughs in the smoothed data.

Another limitation is the lag effect, especially when applied to non-centered data. Since the SMA uses past data points up to the current time, it inherently lags behind real-time changes. In situations where real-time responsiveness is critical, alternative moving average methods, such as exponential moving averages (which place more weight on recent data), may be preferable.

5 Methods

5.1 Analysing Luck Metricisation

To quantify the role of luck in NCAA basketball outcomes, we apply a metrization process that leverages the R^* factor, which has been used effectively to balance skill and chance in various professional sports, as outlined in the SIAM paper by Hosoi et al. This metric relies on examining game outcomes through a mathematical framework that distinguishes consistent skill-driven outcomes from those influenced by luck.

In our approach, we measure the consistency of teams' win fractions over different segments of the season. This concept is based on comparing first and second halves of team performance to assess "persistence." A high correlation between halves would indicate skill dominance, as skill is expected to persist; a low correlation or spread suggests a larger influence of luck.

The R^* metric is derived by rotating and adjusting win-loss data into new coordinates, defined as S and T, where:

- S: Captures the degree to which win rates differ from an average of 0.5 (indicating more skill if they deviate significantly from chance). - T: Reflects variance between first and second half win rates, where high variance in T suggests the influence of luck.

The variance along these axes helps distinguish skill from luck:

$$R^* = 1 - \frac{B}{A}$$

where A is the variance along S (indicating skill), and B is the variance along T (indicating luck).

This calculation can be applied to the 2024 NCAA data, providing an insight into whether win rates correlate strongly across the season, pointing to a skill-based outcome, or if results fluctuate, suggesting a substantial luck component. High R^* values imply a skill-driven league, whereas values near zero denote high randomness in outcomes, essential for predicting and interpreting tournament performance.

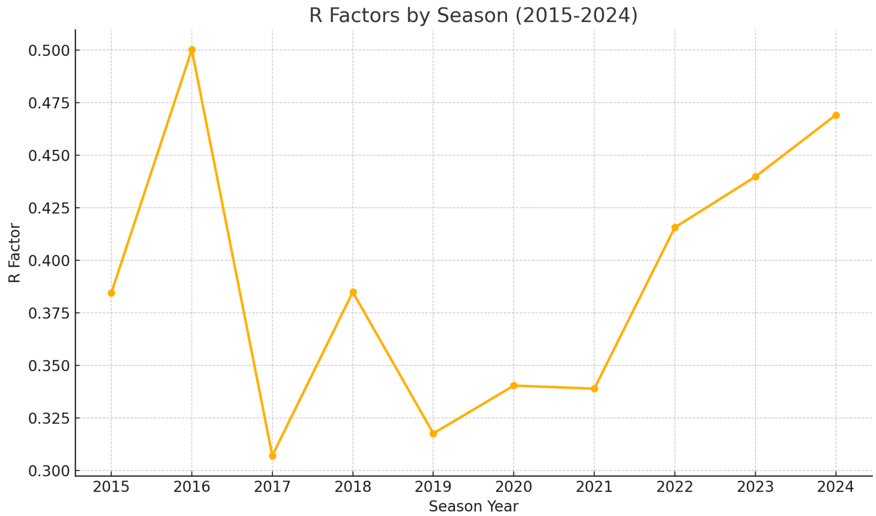

Figure 1: R factor for NCAA Seasons For The Past 10 Years

In our model, we sought to quantify the role of luck in determining game outcomes by examining specific in-game statistics that could indicate random influences on results. We identified several metrics that could serve as proxies for luck, based on their susceptibility to unpredictable events or outcomes that deviate from skill-based consistency. These metrics allow us to assess how much of a game's outcome may have been determined by chance rather than purely by the relative skills of the teams. The indicators of luck we used, along with others we identified as potential additions for future models, are as follows:

1. Close Game Indicators

    - Largest Lead: Smallest leads throughout the game suggest a tightly contested matchup, where minor events (like a single turnover or unexpected foul) could significantly impact the outcome. Games with narrow leads are likely influenced by chance, as no team maintains a dominant performance.

- Points Scored vs. Opponent's Points Scored: When games end with a small point difference (e.g., within five points), minor factors—such as a single lucky shot or foul—can easily swing the outcome. Close game results are therefore treated as potentially luck-influenced.

2. Turnovers and Points from Turnovers

   - Turnovers: Turnovers can result from both skill and unpredictable events (e.g., a bad pass or an unexpected loose ball). High turnover rates may indicate a chaotic or luck-driven game, as turnovers can be less predictable.
   - Turnover Points: Points scored off turnovers add an opportunistic element that might be influenced by luck, especially if many points stemmed from turnovers. These points often depend on specific, isolated plays rather than consistent team performance.

3. Free Throws and Foul Statistics

   - Flagrant and Technical Fouls: Such fouls are often situational and disrupt game flow in unexpected ways. If one team accumulates many flagrant or technical fouls, the outcome may be more luck-influenced due to increased free throw opportunities and possession changes.
   - Free Throw Percentage: Deviations in free throw accuracy, especially large discrepancies from a team's typical performance, can introduce randomness. In games where a team shoots significantly better or worse than usual, luck may play a larger role.

4. Shooting Efficiency Metrics

   - Field Goal Percentage (FG%): Variability in shooting efficiency, particularly for three-point attempts, can reflect luck. Uncharacteristically high or low field goal percentages indicate a game outcome potentially influenced by unexpected shooting success or failure.
   - Three-Point Field Goal Percentage: Three-point shooting tends to be more volatile than closer shots, so games with unusual three-point performance (either very high or low) might be influenced by luck-driven shooting.

5. Steals and Blocks

- Steals: While steals require skill, they can also result from random events (e.g., an opponent's lapse). High steals in a game, especially if uncharacteristic, could introduce luck factors due to increased turnover chances.
- Blocks: Blocked shots might indicate defensive strength but could also signal missed offensive opportunities. Games with numerous blocks may be slightly luck-driven if opponents struggle to avoid defenders.

6. Fast Break Points

   Fast breaks are often initiated by turnovers or defensive rebounds, leading to potentially luck-driven scoring opportunities. A game with high fast break points may be affected by favorable (or unfavorable) fast transitions.

7. Rebounds

   Rebounding success, especially for offensive rebounds, can generate additional scoring opportunities. Games where one team has significantly more rebounds than usual might involve elements of randomness, as they benefit from advantageous positioning or fortunate bounces.

8. Home/Away Factor

   Home games provide a slight performance edge due to familiarity and crowd support. Away wins, especially in close games, may suggest that the winning team benefitted from luck or exceeded expectations due to less controllable factors.

9. Points in Paint vs. Three-Point Field Goals

   - Points in Paint: Higher points in the paint indicate reliance on high-percentage, skill-driven shots, whereas three-point shots are more variable.
   - Three-Point Field Goals: Teams relying heavily on three-pointers are often more subject to luck-based outcomes due to the increased variability in shot success.

Some Additional Factors for Future Model Improvement

While our model captures several indicators of luck, there are additional factors that could improve its predictive power by accounting for other sources of randomness. Due to limitations in our data, we were unable to incorporate these elements, but they are worth considering for a more comprehensive analysis:

- Injury/Player Availability: The absence of key players due to injury or other reasons introduces an element of unpredictability, as team performance often shifts significantly in response to player availability.

- Rest Days: Fatigue from a packed game schedule could affect performance, adding randomness if a team has had fewer rest days than usual before a game.

- Opponent Strength: Games played against particularly strong or weak opponents are less random, as skill discrepancies usually determine the outcome. Upsets in such games could be flagged as luck-driven.

- Last-Minute Shots: Games decided by last-second plays often reflect high levels of luck, as the outcome can hinge on a single successful shot or defensive stop in the final moments.

These additional factors could improve the robustness of our luck metric by capturing more subtle elements that influence game variability. Incorporating these into future models may yield a deeper understanding of how skill and luck shape game outcomes and enhance predictive capabilities for analyzing season-long performance.

5.2 Data Structuring

When deciding how to structure our data, our primary concern was compiling a list of a predictive factors (both luck-based and skill-based) for whether or not a team would win the game. We collected data on each game played in the 2023 NCAA season, the teams playing and their 10 point moving averages of skill and luck based factors prior to the game. We took this data and completed a traditional 80/20 test/train data split, and trained the relevant models using the randomly selected training data.

Statistic	**Statistic**
Team Score	Assists
Blocks	Defensive Rebounds
Fast Break Points	Field Goal Percentage
Field Goals Made	Field Goals Attempted
Flagrant Fouls	Fouls
Free Throw Percentage	Free Throws Made
Free Throws Attempted	Largest Lead
Offensive Rebounds	Points in Paint

Table 1: Statistics Table 1

Statistic	Statistic
Steals	Team Turnovers
Technical Fouls	Three-Point Field Goal Percentage
Three-Point Field Goals Made	Three-Point Field Goals Attempted
Total Rebounds	Total Technical Fouls
Total Turnovers	Turnover Points
Turnovers	

Table 2: Statistics Table 2

5.3 10 Point Moving Average

Within the NCAA, team performance varies in-season, depending on a variety of factors, including player fatigue/injury, morale, and other soft metrics. This means that games further in the past are less relevant for predicting future games, as they don't as accurately capture these soft metrics in the present. To account for this, we implemented moving averages instead of all-season averages for our predictive factors. We chose a 10-game moving average to balance the fact that older games aren't relevant with the game-to-game variance. In other words, using too few games in the moving average exposes us to the high variance that comes with game performance. Choosing 10 games us both to ignore out-of-date games while also average enough games to largely eliminate high game-to-game variance. We chose to disregard games where there weren't 10 games in the past to compute the pre-game average metrics.

5.4 Principal Component Analysis

To identify the highest contributing factors to whether a team lost or won, we ran a principal component analysis. We chose 10 components to fit our data. We selected 10 components because they explained a significant portion of the variance in the dataset and avoided the inclusion of components that would contribute minimal new information. The importance of each factor was calculated using the PCA itself and its logistic regression coefficients.

The most influential factors for both teams include Team 2's Average Team Score (0.082) and Team 1's Average Field Goal Percentage (0.081), suggesting that scoring efficiency and overall team performance are crucial for determining the outcome. Additionally, Team 2's Average Field Goals Made (0.074) and Team 1's Average Three-Point Field Goal Percentage (0.073) also rank highly, highlighting the importance of shooting accuracy from both

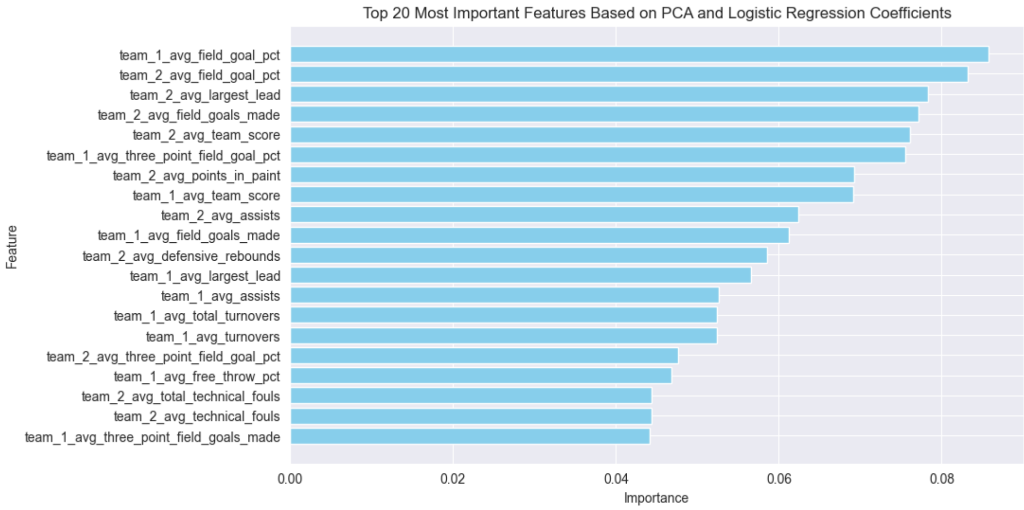

Figure 2: PCA Output

inside and beyond the arc. Other significant factors include Team 1's Average Turnovers (0.058) and Team 2's Average Largest Lead (0.076), indicating that minimizing turnovers and maintaining a commanding lead are key contributors to winning. The relative importance of these features demonstrates that efficient scoring, strong shooting percentages, and the ability to control the game by reducing mistakes and maintaining leads are essential factors in determining the outcome of a match.

5.5 Support Vector Machine

We split our data 80:20 to train and test, set shuffle to true for randomization in the order before splitting and set the random state to 104 in order to ensure reproducibility in the splitting process.

We used a Radial Basis Function (RBF) Kernel in our Support Vector Machine with $nfeatures = 55$ where

$$\gamma = \frac{1}{nfeatures \times X.var()}$$

$$C = 1.0$$

5.6 Random Forest

The Random Forest Classifier is useful in this context because it can handle a large number of input features, like the statistical averages for multiple performance metrics, and it is effective for predicting categorical outcomes (win/loss). We instantiated our Classifier with a maximum depth of 20 for each tree, preventing over fitting by limiting how complex each tree can get.

6 Model Outcomes

6.1 Principal Component Analysis

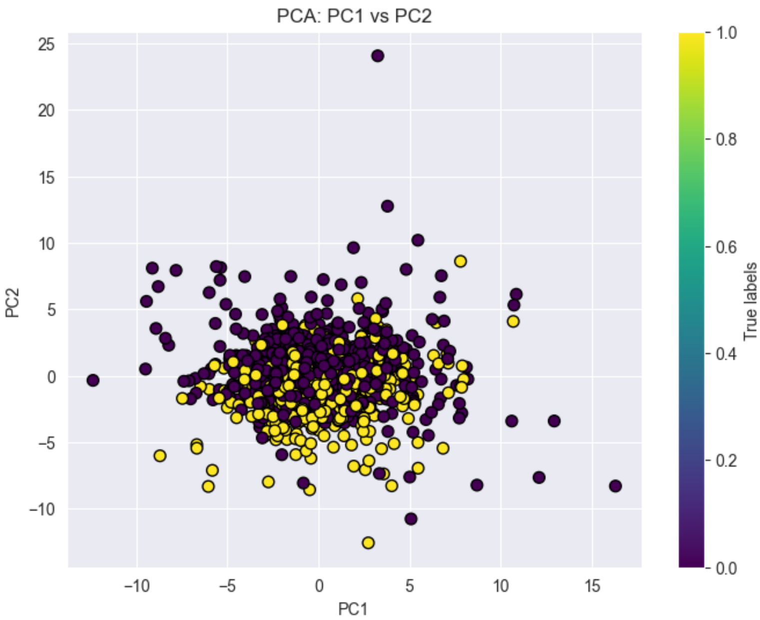

Figure 3: Principal Component Analysis Output

The purple and yellow clusters can be distinguished from each other since they capture the most variance.

6.2 Support Vector Machine

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.659$$

$$Precision = \frac{TP}{TP + FP} = 0.733$$

$$Recall = \frac{TP}{TP + FN} = 0.216$$

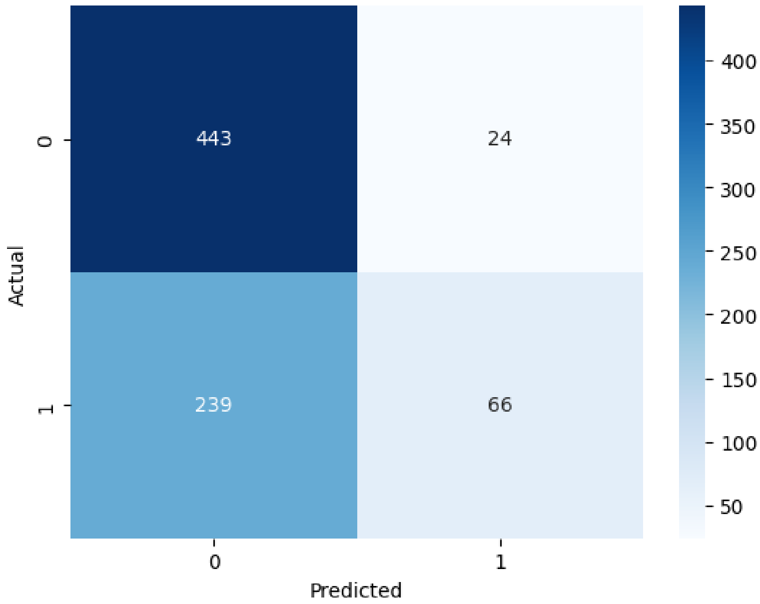

Figure 4: SVM Confusion Matrix

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0.333$$

$$Specificity = \frac{TN}{TN + FP} = 0.948$$

- High Specificity: The model is very good at identifying the negative class (class 0). With a specificity of around 95%, it is reliably predicting when an instance belongs to the negative class.

- Imbalanced Classes: Given the relatively low number of true positives (66) compared to true negatives (443), there is a potential class imbalance in the data, which might be influencing the model's behavior.

- Accuracy vs. Precision/Recall: The overall accuracy of 65.9% may appear acceptable at first glance, but the model has poor recall for the positive class and a low F1-score, indicating that the accuracy is largely driven by its ability to correctly predict the negative class (class 0).

6.3 Random Forest

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.659$$

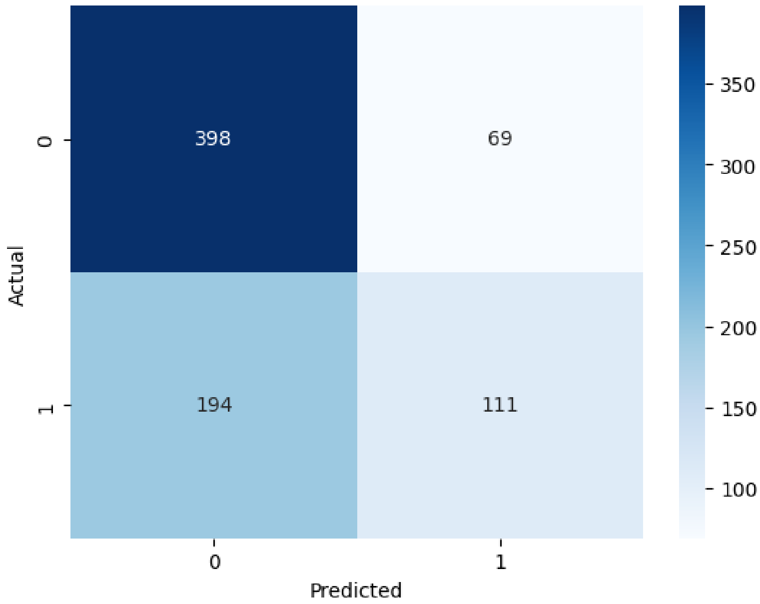

Figure 5: RF Confusion Matrix

$$Precision = \frac{TP}{TP + FP} = 0.617$$

$$Recall = \frac{TP}{TP + FN} = 0.364$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 0.459$$

$$Specificity = \frac{TN}{TN + FP} = 0.852$$

- Low Recall for Class 1: The model's recall for class 1 is quite low (36.4%), meaning it misses a significant number of actual positive cases. Although this is an improvement compared to the earlier confusion matrix (21.6% recall), it still suggests that the model has a bias towards predicting the negative class.

- Moderate Precision for Class 1: The precision for class 1 is 61.7%, which is better than in the previous confusion matrix (73.3%), but still not ideal. It shows that while the model is relatively accurate when it predicts class 1, it still makes a fair number of false positives.

- Imbalanced Classes: Similar to the previous case, there seems to be some class imbalance, with more true negatives (398) than true positives (111). The model might be focusing more on predicting class 0 due to this imbalance, leading to lower recall for the positive class.

- F1-Score: The F1-score of 0.459 is an improvement over the previous 0.333 but still reflects a suboptimal performance for the positive class. It shows the model's struggle to balance precision and recall, especially for class 1.

6.4 Key Results

Traditionally, the NCAA has an upset rate of between 35-40%, meaning that the "better" team wins 60-65% of the time. Because of this, it is very challenging to create a model with an accuracy higher than this range. Given this context, we believe our models perform well, given that we don't factor any sort of upset probability in.

When looking at the PCA, we see some group differentials, indicating that modelling a categorical response variable with SVMs and Random Forests is a reasonable approach. When fitted, both the SVM and Random Forest models exhibit a tendency to favor the negative class, resulting in low recall for the positive class. In the case of NCAA tournament predictions, the frequency of upsets means that the model is tasked with correctly identifying a less frequent outcome - when the "underdog" (lower-seeded team) wins. Due to the high upset date, this is a difficult task, especially when the majority of historical outcomes favor the higher-seeded team. This imbalance limits the model's performance.

Furthermore, when looking at luck versus skill based factors, we see that the most powerful predictive factors are skill-based, including Team Points Scores and Average Field Goal Percentage. More luck-based factors, like rebounds and steals generally have very low predictive power. We argue that the results indicate that Men's College Basketball is still a skill-based sport, but not without elements of luck that can compound into upsets, especially in closely-matched games.

7 Discussion

7.0.1 Model Strengths

1. Comprehensive Feature Set: By integrating a wide array of game metrics (e.g., shooting percentages, turnovers, free throw performance, and points in the paint), the model provides a well-rounded representation of factors impacting game outcomes. This range enables the model

to account for both skill and chance elements, enhancing its ability to capture complex dynamics.

2. Advanced Machine Learning Techniques: Utilizing Support Vector Machines (SVM) with an RBF kernel, random forests, and PCA for dimensionality reduction, our model benefits from advanced approaches that are capable of handling non-linear relationships and high-dimensional data. The random forest classifier, for instance, captures interactions between features, while SVM with RBF kernel adapts to variations in data, essential for modeling unpredictable sports outcomes.

3. Moving Averages for Recent Performance Trends: By implementing a 10-game moving average, the model smooths out recent performance metrics, thus balancing the impact of long-term team strengths and short-term form. This approach helps mitigate the effects of game-to-game variance, ensuring that predictions are not overly influenced by outlier performances.

4. Luck Metricization with R^* Factor: The model's luck metric, based on the R^* factor, allows us to quantify the role of chance in game outcomes by assessing consistency in win rates. This factor captures how much of each team's season can be attributed to persistent skill versus random, luck-driven fluctuations, providing a unique insight into the skill-luck balance in NCAA games.

7.0.2 Model Weaknesses

1. Imbalanced Class Performance: Analysis of the model's confusion matrices and F1-scores indicates that it performs better in predicting the negative class (games where the "expected" team won) than the positive class (upsets or unexpected wins). This imbalance can limit the model's ability to capture upset scenarios, which are often of particular interest in NCAA tournament predictions.

2. Sensitivity to Variability in Game Factors: Certain metrics, such as turnovers or free throw percentage, may introduce noise due to variability from game to game. While the moving average reduces some of this noise, it may not entirely account for sudden shifts in player performance or strategy, which could impact model reliability in high-stakes, high-variability games.

3. Lack of Contextual Variables: While our model includes many in-game metrics, it currently lacks certain contextual factors that can influence

outcomes, such as injury status, rest days, and strength of opponent. These factors are known to impact performance but were not available in our dataset, limiting the model's ability to capture nuances in team conditions.

4. Potential Overfitting with Random Forest: The random forest classifier, while powerful, may risk overfitting if too many trees or a high maximum depth is used. Although we mitigated this by setting a maximum depth, the model's performance could still be sensitive to tuning, requiring further adjustments to achieve optimal balance.

7.0.3 Future Considerations

1. Incorporate Additional Contextual Factors: Adding features such as player injury status, rest days between games, and opponent strength could enhance the model's predictive capabilities. These variables would allow the model to account for game-specific conditions that significantly affect outcomes but are not directly captured by in-game metrics alone.

2. Refine Luck Metricization Approach: Expanding on the R^* factor by including metrics such as last-minute shots or outcomes of high-stakes possessions could provide a more detailed view of luck's impact on game outcomes. Additionally, using alternate forms of metricization that capture different luck-driven scenarios could improve the robustness of the model's skill-luck balance assessment.

3. Enhance Upset Prediction with Synthetic Data: Given the model's imbalance in class prediction, future work could incorporate synthetic data generation or resampling techniques to better balance the positive and negative classes. This could improve the model's recall for upsets and unexpected outcomes, which are challenging yet critical in sports predictions.

4. Apply Cross-Validation and Hyperparameter Tuning: While initial tuning was applied, further hyperparameter optimization, especially for SVM (e.g., adjusting gamma and C) and random forest (e.g., varying maximum depth and number of trees), could improve performance. Cross-validation would also help verify the model's generalizability across different subsets of data, mitigating overfitting risks.

5. Integrate Additional Time-Series Techniques: Although moving averages smooth recent data, other time-series techniques, such as Expo-

nential Moving Average (EMA) or Autoregressive Integrated Moving Average (ARIMA) models, could better capture short-term trends and autocorrelations in team performance.

6. Consider Ensemble Methods: Future iterations could explore combining multiple machine learning models, such as a stacking ensemble of SVM, random forest, and neural networks. An ensemble approach could leverage the strengths of each method, potentially enhancing the model's ability to predict complex and uncertain outcomes in a high-variance tournament setting.

In summary, while our model effectively integrates a range of skill-based and chance-related features, enhancing it with additional contextual factors, refined metricization techniques, and advanced tuning and ensemble strategies could significantly improve its predictive power and robustness.

8 Letter

Chief Editor
Newspaper Name
Newspaper Address

Subject: NCAA Basketball – Skill vs. Luck: An Analysis for Fan Engagement

Dear [Editor's Name],

I am pleased to share the findings of our recent study on NCAA basketball and the roles of skill and luck in determining game outcomes. Our goal was to create a model capable of predicting which team might win a game by analyzing a mix of skill-based factors (e.g., shooting accuracy) and luck-based factors (e.g., turnovers). Using data from past seasons, including scores, win/loss records, and more detailed game statistics, we were able to capture both the reliable patterns and the unpredictable nature that fans love in basketball.

Key Findings and Insights for Fans:

- **Skill vs. Luck Balance:** Our analysis confirmed that while skill is the primary driver of wins, luck plays a meaningful role in close games. For example, when a team wins by a narrow margin, small events, like a single free throw miss or a lucky steal, can change the outcome. Fans might enjoy this "unpredictable edge" of close games, which adds suspense to each tournament round.

- **High Impact Factors:** Certain metrics stood out as reliable predictors of winning. Shooting accuracy, especially for three-point shots, and limiting turnovers were consistently associated with successful outcomes. When teams control these skill-driven factors, they are more likely to win, and fans can watch for these stats to gauge how a game might unfold.

- **Upsets and Randomness:** Our model faced challenges when predicting upsets, reflecting the high "luck factor" in surprising wins. For fans, this is a reminder that no matter the seeding, every game can bring surprises. This unpredictability keeps March Madness engaging and is part of what makes underdog victories so memorable.

Future Considerations: Enhancing predictions further could involve adding new data, such as player injuries and rest days, to account for the physical demands of a packed tournament schedule. Additionally, measuring factors like last-minute shots would capture even more of the luck-driven drama that fans experience in real-time.

Our model captures some of what makes basketball captivating, and these findings can help your readership appreciate both the statistical underpinnings of team success and the thrilling uncertainty that keeps them watching. Please let us know if you'd like to discuss these findings further or if we can support the presentation of these insights in your publication.

Best regards,
Avantika Chopra
Tanish Kumar
Shambhavi Sinha

9 Works Referred to

References

[1] Mark Brown and Joel Sokol. "An Improved LRMC Method for NCAA Basketball Prediction". In: *Journal of Quantitative Analysis in Sports* 6.3 (July 19, 2010). Publisher: De Gruyter. ISSN: 1559-0410. DOI: 10.2202/1559-0410.1202. URL: https://www.degruyter.com/document/doi/10.2202/1559-0410.1202/html (visited on 11/17/2024).

[2] chang_trenton. *Answer to "Mathematics of Random Forest Classifier"*. Cross Validated. Oct. 7, 2023. URL: https://stats.stackexchange.com/a/628207 (visited on 11/17/2024).

[3] Sahel Eskandar. *Introduction to RBF SVM: A Powerful Machine Learning Algorithm for Non-Linear Data*. Medium. Mar. 26, 2023. URL: https://medium.com/@eskandar.sahel/introduction-to-rbf-svm-a-powerful-machine-learning-algorithm-for-non-linear-data-1d1cfb55a1a (visited on 11/17/2024).

[4] *Feeling Lucky? The Relative Roles of Skill and Chance in Sports — SIAM*. Society for Industrial and Applied Mathematics. Oct. 1, 2024. URL: https://www.siam.org/publications/siam-news/articles/feeling-lucky-the-relative-roles-of-skill-and-chance-in-sports/ (visited on 11/17/2024).

[5] Daniel Getty et al. "Luck and the Law: Quantifying Chance in Fantasy Sports and Other Contests". In: *SIAM Review* 60.4 (Jan. 2018), pp. 869–887. ISSN: 0036-1445, 1095-7200. DOI: 10.1137/16M1102094. URL: https://epubs.siam.org/doi/10.1137/16M1102094 (visited on 11/17/2024).

[6] J. R. Kreiger. *Evaluating a Random Forest model*. Analytics Vidhya. Sept. 10, 2021. URL: https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56 (visited on 11/17/2024).

[7] Paul Kvam and Joel S. Sokol. "A logistic regression/Markov chain model for NCAA basketball". In: *Naval Research Logistics (NRL)* 53.8 (Dec. 2006), pp. 788–803. ISSN: 0894-069X, 1520-6750. DOI: 10.1002/nav.20170. URL: https://onlinelibrary.wiley.com/doi/10.1002/nav.20170 (visited on 11/17/2024).

[8] Paul Kvam and Joel S. Sokol. "A logistic regression/Markov chain model for NCAA basketball". In: *Naval Research Logistics (NRL)* 53.8 (2006). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20170, pp. 788–803. ISSN: 1520-6750. DOI: 10.1002/nav.20170. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20170 (visited on 11/17/2024).

[9] *Moving average*. In: *Wikipedia*. Page Version ID: 1252528298. Oct. 21, 2024. URL: https://en.wikipedia.org/w/index.php?title=Moving_average&oldid=1252528298 (visited on 11/17/2024).

[10] *NBA-Prediction-Modeling/NBA-Prediction-Modeling.ipynb at main · luke-lite/NBA-Prediction-Modeling*. GitHub. URL: https://github.com/luke-lite/NBA-Prediction-Modeling/blob/main/NBA-Prediction-Modeling.ipynb (visited on 11/17/2024).

[11] Henry Palattella. *College Basketball's March Madness Betting By The Numbers*. Forbes Betting. Section: College-Basketball. Sept. 21, 2024. URL: https://www.forbes.com/betting/basketball/college-basketball/march-madness-by-the-numbers/ (visited on 11/17/2024).

[12] *Principal Component Analysis (PCA) Explained*. Built In. URL: https://builtin.com/data-science/step-step-explanation-principal-component-analysis (visited on 11/17/2024).

[13] *Random Forest Classifier using Scikit-learn*. GeeksforGeeks. Section: AI-ML-DS. Sept. 4, 2020. URL: https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/ (visited on 11/17/2024).

[14] *Reflections on March Mathness — SIAM*. Society for Industrial and Applied Mathematics. Apr. 3, 2023. URL: https://www.siam.org/publications/siam-news/articles/reflections-on-march-mathness/ (visited on 11/17/2024).

[15] *Sabermetric Research: The Tango method of regression to the mean – a proof*. URL: http://blog.philbirnbaum.com/2011/08/tango-method-of-regression-to-mean-kind.html (visited on 11/17/2024).

[16] Madison Schott. *Random Forest Algorithm for Machine Learning*. Capital One Tech. Feb. 27, 2020. URL: https://medium.com/capital-one-tech/random-forest-algorithm-for-machine-learning-c4b2c8cc9feb (visited on 11/17/2024).

[17] Sushanth Sreenivasa. *Radial Basis Function (RBF) Kernel: The Go-To Kernel*. Medium. Oct. 12, 2020. URL: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a (visited on 11/17/2024).

[18] *TrueSkill™ Ranking System*. Microsoft Research. URL: https://www.microsoft.com/en-us/research/project/trueskill-ranking-system/ (visited on 11/17/2024).

[19] *What Is Random Forest? — IBM*. Oct. 20, 2021. URL: https://www.ibm.com/topics/random-forest (visited on 11/17/2024).

10 Appendix

10.1 Code

Link to Github!
SportsDataVerse Packages

10.2 Data

Table 3: Outputs of PCA

Feature	Importance
Team Home or Away	0.000
Team 1 Avg Team Score	0.070
Team 1 Avg Assists	0.049
Team 1 Avg Blocks	0.003
Team 1 Avg Defensive Rebounds	0.027
Team 1 Avg Fast Break Points	0.019
Team 1 Avg Field Goal Pct	0.081
Team 1 Avg Field Goals Made	0.058
Team 1 Avg Field Goals Attempted	0.012
Team 1 Avg Flagrant Fouls	0.000
Team 1 Avg Fouls	0.037
Team 1 Avg Free Throw Pct	0.052
Team 1 Avg Free Throws Made	0.041
Team 1 Avg Free Throws Attempted	0.026
Team 1 Avg Largest Lead	0.056
Team 1 Avg Offensive Rebounds	0.035
Team 1 Avg Points In Paint	0.040
Team 1 Avg Steals	0.000
Team 1 Avg Team Turnovers	0.042
Team 1 Avg Technical Fouls	0.003
Team 1 Avg Three Point Field Goal Pct	0.073
Team 1 Avg Three Point Field Goals Made	0.041
Team 1 Avg Three Point Field Goals Attempted	0.006
Team 1 Avg Total Rebounds	0.000
Team 1 Avg Total Technical Fouls	0.003
Team 1 Avg Total Turnovers	0.058
Team 1 Avg Turnover Points	0.033
Team 1 Avg Turnovers	0.058

Feature	Importance
Team 2 Avg Team Score	0.082
Team 2 Avg Assists	0.069
Team 2 Avg Blocks	0.028
Team 2 Avg Defensive Rebounds	0.062
Team 2 Avg Fast Break Points	0.041
Team 2 Avg Field Goal Pct	0.084
Team 2 Avg Field Goals Made	0.074
Team 2 Avg Field Goals Attempted	0.018
Team 2 Avg Flagrant Fouls	0.000
Team 2 Avg Fouls	0.018
Team 2 Avg Free Throw Pct	0.026
Team 2 Avg Free Throws Made	0.031
Team 2 Avg Free Throws Attempted	0.023
Team 2 Avg Largest Lead	0.076
Team 2 Avg Offensive Rebounds	0.010
Team 2 Avg Points In Paint	0.051
Team 2 Avg Steals	0.009
Team 2 Avg Team Turnovers	0.003
Team 2 Avg Technical Fouls	0.027
Team 2 Avg Three Point Field Goal Pct	0.070
Team 2 Avg Three Point Field Goals Made	0.061
Team 2 Avg Three Point Field Goals Attempted	0.029
Team 2 Avg Total Rebounds	0.037
Team 2 Avg Total Technical Fouls	0.027
Team 2 Avg Total Turnovers	0.001
Team 2 Avg Turnover Points	0.033
Team 2 Avg Turnovers	0.001