Predicting NBA outcomes through uncovering team networks and learning latent positions

Hannah Huh
Princeton University
hannahh@princeton.edu

Hari Raval
Princeton University
hraval@princeton.edu

Isabel Medlock
Princeton University
imedlock@princeton.edu

Tommy Nguyen
Princeton University
tommyn@princeton.edu

Abstract

The NBA (National Basketball Association) is an organization which negotiates the varying interests of players, coaches, owners, and fans. One area of common interest is understanding team and player performance. Teams can obtain insights into their level of play and make decisions on personal changes. Fans can use these insights to engage in the betting market. Analysis of NBA game data from 2004 to 2020 uncovers a complex team network and latent player positions. In this work, we use classification models to predict the outcome of 24,196 games and regression models to predict the impact 2,407 players have on games. We aid these prediction tasks by building digraphs of team networks based on game outcomes and creating new features via clustering on a reduced dimension data set of players. We find that logistic regression achieves an accuracy of 84.89% in predicting game outcome when including network-based features and ridge regression performs best in predicting player impact when using cluster label features.

1 Introduction

The NBA is composed of 30 basketball teams and is one of four professional sports leagues in the United States with abundant viewership. The complex organization produces a vast quantity of game and player statistics throughout each of their seasons. Players, coaches, and owners are especially interested in learning insights from game-by-game and player-by-player data in order to improve their performance relative to other teams. On the other hand, fans are inspired to utilize data-driven predictions in order to enhance winning odds when engaging in the sports betting market, which has recently gained popularity. To address such motivations, we use NBA data from 16 seasons in order to make predictions and uncover trends surrounding team and player performance.

In our work, we use four classifiers on data pertaining to 24,196 games to predict which team wins. After performing hyper parameter tuning and chi-squared feature selection to determine the optimal number of features, we perform network analysis. We create digraphs to represent the interactions between teams in a conference in order to determine the best teams across seasons. Our digraphs are leveraged to build an indegree feature which we use to improve our prediction of game outcomes. We also implement four regressors on player-specific data in order to determine the impact a player has on a game - formally called the "plus-minus" score. We perform hyper parameter tuning and feature selection to improve our predictions. Moreover, we use principal component analysis to uncover latent positions and perform k-means clustering of players by position. Our work concludes by extracting features from our clustering results in order to improve player performance prediction.

2 Related Work and Methods

2.1 Related Work

Sports analytics has been a motivation for numerous work, and the majority of studies have largely to do with the prediction and study of the most important factors related to either game outcomes or individual performance of players (e.g. percentage of free throws made). A limited portion of the literature focuses on the NBA, with most work emphasizing developing models to predict the outcome of the NBA playoffs or specific games [3] [8] [15]. Depending on the model, predictions for the outcome of games have ranged from about 50% to the upper ranges of 80% [15] [18] [14].

Of additional interest, however, is understanding players' impact on a game or uncovering insights about their 'role' on a team. One work examines a slightly adjacent question in attempting to predict players' points in a game relative to their personal average [24]; yet within existing literature, there is little to be said about modelling teams and players as networks or investigating what kind of latent structures exist within a team. Since NBA games inherently involve both intra and inter-team interactions, we believe studying the influence players have on the outcome of the game and the nature of relationships between teams presents an opportunity for additional insight. Using our analysis to create game and player-related features, we also recover baseline results of classification and regression on our data and then improve these tasks with the introduction of our novel features.

2.2 Data Processing

We downloaded 2004-2020 NBA season data for 2,407 players and 24,196 basketball games from Kaggle [12]. The data set contains five CSV files which provide information on game statistics, player statistics, players, and team rankings. To clean the data, we first removed columns that were either redundant or not relevant to our tasks. Namely, we dropped text columns pertaining to the game status, game date, team name abbreviation, team city, and coach comments. We also removed redundant columns regarding number of particular shots attempted and made as percentages are provided. After this, we had to reformat and convert the minutes played per player column into a numeric value. We use KNN imputation to impute missing game and player statistic values [17]. Next, after performing feature selection with a chi-squared test, we hypertuned the feature space to 12 features for game prediction and 15 features for player performance prediction. Finally, we standardized all features by removing the mean and scaling to unit variance.

2.3 Methods

We implement four classification methods from the SciKitLearn Python library to predict NBA game outcomes [17]. First, naive bayes was chosen as a baseline for its simplicity and effectiveness in many settings [23] [5]. Despite the independence assumption among features, naive bayes can adjust posterior probabilities per class and leave the class with the highest posterior probability unchanged to produce correct classifications [22] [5]. Next, a linear support vector machine was chosen due to its proven robustness in adapting to a large feature space, as is the case with our game and player data [11] [5]. Thirdly, logistic regression was included due to its proven consistency in linearly separable classification problems and straightforward classifier optimization through regularization [9] [5]. Finally, a random forest was selected due to its ensemble approach, unlike the other three classifiers. Moreover, prior research has proven that random forests provide robust classification for noisy data like with our player data which varies drastically per game [10] [5].

We use four regression methods to predict players' individual impact on a game [17]. Our first regressor, least squares linear regression, was chosen for its simplicity and speed across many regression problems [13] [6]. Despite its lack of regularization and susceptibility to outliers and non-normality, researchers have often found it to be a reasonable baseline and yield accurate predictions [1] [6]. Next, lasso regression was chosen due to its use of regularization to avoid overfitting and ability to shrink and remove coefficients that reduce variance [19] [6]. Thirdly, ridge regression was used due to its ability to shrink coefficients without setting the majority of them to zero [16]. To predict player performance, this is especially helpful given the importance of various statistics other than points scored in determining impact. Finally, elastic net was included due to its ability to capture the benefits of both lasso and ridge regression: sparsity and stability [7] [6].

We explored dimensionality reduction prior to clustering our player data. Principal component analysis (PCA) was chosen to create a reduced representation of the players due to its ability to deal with sparsity and yield an effective representation of complex data while retaining the most variant information [21]. PCA provides a good approximation for NBA player data as it provides a low rank representation of the original matrix. To perform clustering, we use k-means for its guarantees on convergence and generalization to varying numbers of clusters [2] [5]. Lastly, we investigated creating digraphs with teams as nodes and win-loss relationships as edges to see if an overall hierarchy exists between teams across years or within a given year. We also examine graph properties like pagerank, HITS, and indegree to see if they can be used to predict the NBA champions and also create additional features for classification.

2.4 Expectations

In predicting game outcomes, we expect that our classifiers will reach 100% accuracy when including features pertaining to the points scored by each team. In this case, it would be straightforward to recover the final game status as a model simply needs to learn to select the larger value. We expect that when removing the features pertaining to points scored by each team, our models will result in accuracies worse than random. This is likely because the only information available will be non-score based features such as types or percentages of shots made and defensive statistics. With this, we believe that a random forest will produce the highest accuracy due to its ability to withstand noise among the game data. For example, there are often cases when a team has low shooting percentages but wins due to a high number of shots taken and ample free throw opportunities. For classification feature importance, we predict that the game outcomes will be most dependent on the percentage of three pointers a team makes since these are the most difficult, but highest point shots in a game. Next, to predict player impact we expect that elastic net will yield the best performance due to its ability to capture stability, but still leverage sparsity in sending less significant features to zero.

For clustering, we expect that players will be separated into groups based on position. After examining the average statistics of players in each of the three positions (Center, Forward, and Guard), we found that certain positions like “point guard” tend to produce a lot more points and offensive-side impact in the game, whereas other positions such as “center” are more defensive in nature. With this, we believe that players can be put into groups based on similarity of certain positions. For our network analysis, we expect that a directed digraph network will reveal a “hierarchy” of teams. In other words, since all teams in a conference play each other, we believe that including edges between teams that beat others will reveal a “best” team across all seasons. Our research suggests that if such a team is found, it will be the Boston Celtics or Los Angeles Lakers as they have the most championships in NBA history.

2.5 Evaluation

For each classifier and regressor, 10-fold cross validation was performed to tune model hyper parameters on the training data. This enabled us to find the most optimized model architectures for the NBA data set. We compared classifiers by accuracy and F_1-score. Accuracy quantifies the percentage of correctly classified points. F_1-score is the harmonic mean between $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$ where we use False Positives (FP), False Negatives (FN), and True Positives (TP) to compare the trade-offs between the two values. We evaluated regressor performance using mean absolute error and R^2 value. R^2 shows the level of fit a regressor has to the data and mean absolute error is an arithmetic average of the differences between true and predicted values, which succinctly indicates model performance. Moreover, for both classification and regression, we build 95% confidence intervals via 150 boot strap samples to uncover the likely values of our metrics. To further evaluate performance, we examine feature importance scores for all classifiers and regressors to find the features which are most impactful in predicting game outcomes and player performances.

For dimensionality reduction, we evaluate PCA by the interpretability of the latent components. We determine the optimal number of components for PCA by plotting the variance explained per component and then finding the “elbow” in the plot. For clustering, we plot the top two principal components and visualize the separation of players by position. To determine and validate the optimal number of clusters for k-means, we create and interpret silhouette plots. For network analysis, to evaluate our digraphs, we find the node with the max pagerank, HITS, and indegree. These network-based metrics will give indication of the best NBA team for a given subgraph. Finally, we also find the strongest connected component for each of our subgraphs in order to determine if there are any interesting groups of teams.

3 Spotlight Technique: Principal Component Analysis

Principal component analysis (PCA) is a dimensionality reduction technique that can be used to reduce the dimensions of a feature space. We may want to apply PCA for many reasons, such as reducing redundant data, finding features that encapsulate all other features, or visualizing data in lower dimensions. PCA has two main assumptions: the data is a linear combination of the features and the features are multivariate Gaussian distributed [4]. These are two reasonable assumptions to make for our NBA data set given that we have data of different players and teams for each feature.

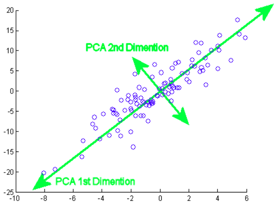


Figure 1: Visualization of principal component analysis along first two components [Python ML].

To describe PCA, we first assume that we would like to reduce our n-dimensional feature space into k dimensions where $k << n$. This requires finding k orthogonal vectors $u^i \in R^n$ to project our data onto, which will minimize the reconstruction error. These k vectors are known as the *principal components (PC)* of PCA. Intuitively, minimizing the reconstruction error is the same as reducing the average distance of every feature to the lower k-dimensional subspace. This also maximizes the variance of the projection vectors u^i in this k-dimensional subspace. There are many ways to select k such as picking the first k until a certain percentage of variance is explained, creating a scree plot and choosing the value of k at the elbow, or treating k as a hyper parameter for cross-validation [4].

To mathematically define PCA, we define m data points: $(x^i, ..., x^m)$ with n features so $x^i \in R^n$. PCA generally assumes mean centered features so we scale features indexed by j as follows:

$$x^i_j = \frac{x^i_j - \mu_j}{s_j}$$

where μ_j and s_j are the mean and standard deviation of a feature j, respectively. Next, we compute the covariance matrix defined by the following:

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^i (x^i)^T$$

The eigenvectors of this covariance matrix capture the max variances of our data and so these eigenvectors are our PCs. To compute these eigenvectors in a straightforward manner, we first find the SVD for our covariance matrix as shown below:

$$\Sigma = UDV^T$$

With the SVD computed, our eigenvectors are given in the columns of the U matrix. Additionally, the eigenvectors are ordered in descending order when using SVD, with the first eigenvector containing the maximum variance. To reduce to k dimensions, we take the first k columns of our U matrix defined by U_k and can now compute our reduced data representation $z^i \in R^k$ in k dimensions:

$$z^i = (U_k)^T x^i$$

Overall, the full algorithm for the PCA technique is as follows:

1. Mean normalize and scale the features x^i_j
2. Compute the covariance matrix Σ
3. Compute the eigenvectors of the covariance matrix, potentially using SVD
4. Use the first k columns of the U matrix from SVD to finally compute the reduced representation $z^i = (U_k)^T x^i$

Ultimately, we can also approximately reconstruct the original data by $\hat{x_i} = U_k z^i$ and this is justified because the PCs minimize the reconstruction error and U is a unitary matrix so $U_k^{-1} = (U_k)^T$.

4 Results

4.1 Classification: Predicting Game Outcomes

To predict game outcomes, we compared three different sets of features as shown in Table 1. We hypertuned the number of features and found that a feature space of size 12 maximized classifiers' performance. First, we observe that models in column one in Table 1 achieved nearly perfect accuracies and F_1 scores. These results are expected as we provide the models with the number of points scored by each team, which reduces the models' task to learning which team has the higher score.

After we removed the team score features, we obtained the Table 1 outcomes in column two. We note that the accuracies and F_1 scores drop by about 15%. Interestingly enough, none of the features used in column two form a linear combination of the points scored. Nonetheless, we were able to obtain 84.08% accuracy and 0.87 F_1 score with logistic regression, which suggests that the outcome of a game is significantly influenced by the types and percentages of a team's shots. Finally, we were able to improve these results by adding a feature from our network analysis in Section 4.2. Column three in Table 1 removes the team score features, but adds an indegree feature representing the number of team wins in a season. Accuracy increases by about 1% and F_1 scores remain at roughly 0.87. Notably, logistic regression reaches its highest accuracy of 84.89%. These improvements suggest that the indegree information is valuable in providing context into a team's holistic performance in a season, which enables us to to better predict performance in an individual game.

Classifier	**Including Team Score Features**		**Removing Team Score Features**		**Adding Network Analysis Feature**	
	Accuracy	F_1	*Accuracy*	F_1	*Accuracy*	F_1
Naive Bayes	88.48: 95% [88.03, 88.65]	0.90	83.82: 95% [83.80, 83.99]	0.87	83.42: 95% [80.32, 85.12]	0.86
Linear SVM	100.0: 95% [100.0, 100.0]	1.0	81.70: 95% [81.02, 82.21]	0.86	84.85: 95% [83.44, 84.89]	0.87
Log. Regression	100.0: 95% [100.0, 100.0]	1.0	84.08: 95% [83.32, 84.33]	0.87	84.89: 95% [83.13, 85.11]	0.87
Random Forest	98.01: 95% [97.67, 98.05]	0.98	83.87: 95% [83.12, 84.05]	0.88	84.53: 95% [83.24, 85.45]	0.87

Table 1: Accuracy with 95% confidence intervals and F_1 scores for predicting game outcomes when including team score features, removing team score features, and adding a custom digraph feature.

With regards to feature importance, classifiers found different features to be significant for prediction depending on whether we included or removed team score features. Table 2 demonstrates that when including team score features, all models found these features which relate to the points scored by the home and away teams to be the most important. However, in addition to these features, we note that models also gleaned information from other features. For example, random forest prioritized field goal (FG_PCT) and three point shooting (FG3_PCT) percentages by both the home and away teams. These values are important as they show how accurately the team is shooting the ball on any given night and more accurate shots suggests more points scored. After removing the points scored by each team, Table 2's right-hand side shows a difference in which features are most important in predicting the game outcomes. Namely, we observe that the number of assists (AST) is important for linear SVM and logistic regression. The number of assists provides insight into a team's chemistry, which suggests that more assists represents better team dynamics. For random forest, field goal (FG_PCT) and three point shooting (FG3_PCT) percentages remained significant. Our custom "indegree" feature proved important to all three classifiers, notably being of top four importance to our best-performing model, logistic regression.

Feature Importance Values when using Team Scores			**Feature Importance Values when removing Team Scores**		
Linear SVM	*Log. Regression*	*Random Forest*	*Linear SVM*	*Log. Regression*	*Random Forest*
PTS_away: 0.500	PTS_home: 0.676	PTS_home: 0.251	AST_away: 0.080	AST_away: 0.167	FG_PCT_away: 0.187
PTS_home: 0.500	PTS_away: 0.674	PTS_away: 0.250	FG_PCT_home: 0.079	AST_home: 0.155	FG_PCT_home: 0.177
FT_PCT_away: 0.006	AST_away: 0.031	FG_PCT_away: 0.128	FG_PCT_away: 0.079	REB_away: 0.104	FG3_PCT_home: 0.087
FG3_PCT_home: 0.003	REB_away: 0.025	FG_PCT_home: 0.108	AST_home: 0.068	indegree: 0.084	FG3_PCT_away: 0.075
FG_PCT_home: 0.003	REB_home: 0.014	FG3_PCT_home: 0.040	REB_away: 0.050	REB_home: 0.071	AST_home: 0.069
FG3_PCT_away: 0.002	AST_home: 0.008	FG3_PCT_away: 0.037	FG3_PCT_home: 0.049	FG3_PCT_away: 0.010	REB_home: 0.068
FT_PCT_home: 0.002	FG3_PCT_home: 0.004	REB_away: 0.036	REB_home: 0.048	FG3_PCT_home: 0.008	REB_away: 0.068
FG_PCT_away: 0.001	FG3_PCT_away: 0.003	REB_home: 0.034	FT_PCT_home: 0.047	FG_PCT_away: 0.008	indegree: 0.058
AST_away: 0.001	FG_PCT_home: 0.003	AST_home: 0.033	FT_PCT_away: 0.046	FT_PCT_away: 0.006	FT_PCT_home: 0.005
REB_away: 0.001	FT_PCT_home: 0.002	AST_away: 0.033	indegree: 0.022	FG_PCT_home: 0.005	AST_away: 0.005

Table 2: Top 10 important features for predicting game outcomes when providing the models team scores versus removing them. "Indegree" is the feature extracted from Section 4.2 network analysis.

4.2 Network Analysis: Visualizing Team Digraphs

Across all years, we initially created a directed graph with 30 nodes and 870 edges. Each node represents a team and each directed edge represents a team that loses to a winning team. Figure 5a in the appendix shows the full directed graph for our data set. This graph, and its strongest connected component in appendix Figure 5b, were too dense to analyze. Thus, we constructed subgraphs for different years, such as the subgraph for 2014 and its strongest connected component in appendix Figures 6a and 6b, respectively. While the density decreased, we observe that the "yearly" networks are still too dense to analyze, likely because each team plays other teams multiple times.

To further reduce density, we placed a constraint on the edges. An edge is only added when a team beats another team by a certain threshold of points. Figure 7 in the appendix represents our networks for 2014 games with point thresholds of 10, 20, 30, and 40 points. We found that all years yield similar subgraphs. From these subgraphs, we observe that the density of both the full graph and its strongest connected component decreases as the point threshold increases. This makes sense because it is hard to gain large point differentials in the time-limit of games. A particularly interesting result is seen in appendix Figure 7f, which is the strongly connected component for 30 points with only 5 nodes. We note that, compared to the previous 20 point differential in appendix Figure 7d, the number of nodes and edges has greatly decreased. This suggests that a ~ 20 point differential is quite common in basketball games but a ~ 30 point differential is not. Additionally, since there are only 5 nodes in Figure 7f, the ~ 30 point differential amount is a boundary at which we can identify groups of teams that are completely winning or losing. To further support this hypothesis, the 40 point difference in appendix Figure 7h reveals a strongest connected component which is a single node, which shows that 40 point differences rarely happen and are outliers.

We also computed graph properties including the max pagerank, HITS, and indegree node and value per year. These results are presented in appendix Figure 8. We see that the maximum nodes and values change depending on the metric applied. The maximum pagerank, authority, and indegree nodes are the same in year 2005, but they generally differ across the different seasons as shown in Appendix Figure 8. This is surprising as we expected that these three metrics would output the same best team for a given year, given the similarity of the metrics themselves. Additionally, the maximum nodes' teams don't usually agree with the actual NBA champions (overall season winner) given in the NBA champion column of appendix Figure 8. The percentage correct for each of the metrics across the years is given in Table 3 below.

Metric	Pagerank	Hub	Authority	Indegree
Percent Correct	0.11%	0%	0.33%	0.28%

Table 3: Accuracy when using the max pagerank, hub, authority, and indegree to compute the NBA champions across all seasons from 2004 to 2020.

We see that the authority metric was the best at computing the NBA champs for a given year. The hub metric was included as a control to confirm that it should never predict the NBA champion, which it never does. Overall, each of the metrics are below $< 50\%$ accurate, indicating poor prediction of future NBA champions. Lastly, we include our values for each of the metrics given in appendix Figure 8 for a given year as additional features for classification. The classification results are presented in Figure 1. We note that our classification accuracy and F_1 scores generally improve for all models when using these additional features. This indicates that these features aid the prediction of whether a team will win a game or not. Finally, from the features table in Figure 2, we see that "indegree" is added as one of the most important features which further supports our conclusion.

4.3 Regression: Predicting Player Performance

To predict player performance (plus-minus score), we compared two different sets of features as shown in Table 4. We hypertuned the number of features and found that a feature space of size 15 maximized regressors' performance. First, we observe that models in Table 4's first column demonstrate consistent mean absolute error values of roughly 8.14 with R^2 values near 0. This illustrates the difficulty of predicting a player's individual impact on a game based solely on his statistics; positive or negative impact can be more accurately judged based on relative performance to other teammates.

We were able to improve our player performance predictions by adding features we extracted from our clustering analysis in Section 4.4. Through adding one-hot encoded features pertaining to the cluster label which a player belongs to, we added information that indicates the "latent" position of a player. Table 4 demonstrates consistent model performance again, however we observe a decrease in errors by about 5% and R^2 values of about 0.16 on average. This improved performance can be attributed to our clustering features' ability to capture contextual information about the player's role relative to other teammates. Overall, we note that while all models perform quite similarly, ridge regression offers slightly lower mean absolute errors with a tighter 95% confidence interval.

Regressor	**Using Player Statistics Only**		**Adding Cluster Label Features**	
	Mean Absolute Error	R^2	*Mean Absolute Error*	R^2
Least Squares	8.14: 95% [4.06, 9.34]	0.01	7.57: 95% [7.23, 7.89]	0.15
Lasso	8.14: 95% [3.98, 9.01]	0.03	7.58: 95% [6.45, 8.05]	0.16
Ridge	8.14: 95% [4.02, 9.56]	0.02	7.56: 95% [7.20, 8.02]	0.17
Elastic Net	8.14: 95% [3.67, 9.65]	0.02	7.56: 95% [6.87, 7.59]	0.17

Table 4: Mean absolute error with 95% confidence intervals and R^2 values for predicting player performance when using only player statistics versus adding clustering label features.

In terms of feature importance, Table 5 demonstrates how all four regressors found the three point shots made (FG3M) to be the most important statistic. This feature is important likely due to the fact that FG3M represents the highest value shot in the game and if a player makes or misses many such shots, their individual contribution will directly be impacted. We also learned that all models found defensive statistics valuable, prioritizing turn overs (TO) and steals (STL), while elastic net had blocks (BLK) and steals (STL) among the top importance scores. These features have no direct relationship to the number of points scored, however they are likely found important due to the fact that they constitute information relevant to the impact a player has on stopping the other team from scoring. Finally, Table 2 shows the importance of our custom features which encode "latent position" based on seven cluster labels from Section 4.4. Specifically, we observe that "label_1" or "bench player" aids prediction for all models, notably being of top four importance for elastic net.

Feature Importance Values when Adding Cluster Label Features			
Least Squares	*Lasso*	*Ridge*	*Elastic Net*
FG3M: 2.994	FG3M: 2.873	FG3M: 2.930	FG3M: 2.891
FG3L: 1.467	TO: 1.392	FG3L: 1.382	STL: 1.179
TO: 1.281	FG3L: 1.341	TO: 1.311	BLK: 0.775
STL: 1.098	STL: 1.262	STL: 1.113	label_1: 0.613
FGL: 0.784	FGL: 0.830	FGL: 0.725	AST: 0.546
BLK: 0.756	label_1: 0.638	label_1: 0.651	label_5: 0.358
label_1: 0.553	BLK: 0.592	BLK: 0.598	PTS: 0.310
ST: 0.537	AST: 0.480	AST: 0.475	label_0: 0.273
label_2: 0.472	PTS: 0.325	label_2: 0.458	DREB: 0.242
PTS: 0.304	label_5: 0.314	FTM: 0.320	FG3_PCT: 0.154

Table 5: Top 10 important features for predicting player performance when including custom clustering features. "cluster_x" values represent the cluster labels extracted from Section 4.4 clustering.

4.4 Clustering: Finding Latent Positions

In order to cluster players, statistics (such as assists made) were averaged for each of the 2,407 players across all games a player participated in. Through inspection of silhouette and inertia plots for $k \in [2, 15]$, we produce $k = 7$ clusters. Figure 3 shows the clustering results. We note that clusters are well divided by both the first and second principle component. In contrast, as shown in appendix Figure 9, the official positions of "Front", "Guard", and "Center" are primarily differentiated along only the second principle component, with little differentiation along the first principle component.

We include the top component weights in Table 6. The composition of the second principal component confirms our understanding of the primary differences between official player positions. For instance, rebounding (OREB, DREB) and blocking (BLK) are trademark skills for "centers", as the tallest players excel in this role. In contrast, turnovers (TO) and points (PTS) are in any player's game, explaining why the first principal component poorly differentiates official player positions.

Furthermore, our cluster labels' improvement of our regression task in Section 4.3 indicates the possibility of "latent" positions. In other words, the clusters we find may indicate additional dimensions

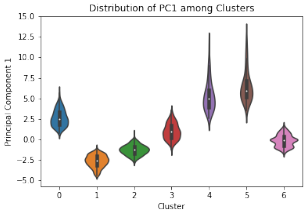


Figure 2: Distribution of PC1 across clusters.

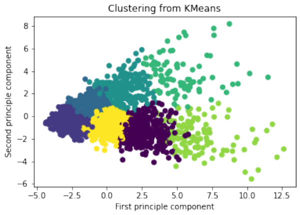


Figure 3: Clusters along first two PCs.

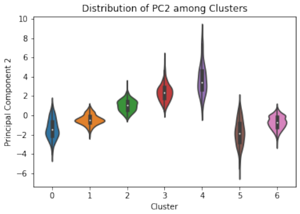


Figure 4: Distribution of PC2 across clusters.

Top 5 Highest-Weighted Features in first two PCs	
PC1	*PC2*
FGL	OREB
PTS	DREB
TO	FTL
AST	BLK
FG3L	FTM

Table 6: Top 5 highest-weighted features in principal components from PCA.

along which players' roles can be differentiated and understood. Average statistics for each cluster can be found in appendix Figure 9. We label three particularly interesting latent positions below:

- *Cluster 1 - "Bench Player"*: On average, cluster 1 performs poorly compared to all other clusters. Players in this cluster make very few shots and thus have low FG3M and FTM scores. However, they also do not miss many shots (FG3L and FTL), which suggests that their overall activity on the court is low. Nonetheless, their presence on the court is correlated with negative impact on score, with an average PLUS_MINUS score of -0.957. Finally, they also appear to interact less with their team, with the average assists (AST) being at least 50% less compared to all other clusters.
- *Cluster 4 - "Dual-Threat Center"*: Typically, "Center" players do not score many three point shots and primarily stick to rebounding and two point shots. Cluster 4 captures a group of players whose main role may be "Center" as players here have high defensive stats (OREB, AST, DREB). However, they also have a high number of points scored (PTS) and three point shots made (FG3M). Our findings reflect an NBA trend that has recently attracted more coverage, where "Center" players have become skilled perimeter shooters.
- *Cluster 5 - "High-Profile Shooter"*: Cluster 5 describes a class of player who excels in one particular area, but is average in other aspects of performance. This cluster is exceptional in making 'flashy' shots and individual plays, with many three point shots (FG3M), free throws (FTM), and steals (STL). These players struggle with defensive plays, indicated by low block (BLK) and rebound (OREB) scores, compared to other clusters.

5 Discussion and Conclusion

Overall, we find that logistic regression yields the best prediction of game outcomes. Our network analysis shows that most games do not exceed a 30 point differential and that generating features from team interactions improves classifier performance for all models. Our regression task does not perform as well, with generally higher mean absolute error, although ridge regression consistently performs the best by a modest margin. We are able to improve regression performance by adding cluster labels from k-means clustering. Our clusters enable us to uncover latent positions, which provide an additional avenue along which to understand player roles and intra-team interactions.

Looking forward, this work can be extended by using more advanced models for our regression and classification tasks, such as mixed membership models. Second, additional features could be extracted from our graph, such as ones relating to centrality of nodes, or incorporation of weighted edges to capture more nuanced relationships between teams. Finally, players' impact on their teams across seasons could be examined to understand how certain positions and roles develop over time.

References

[1] Özlem Gürünlü Alma. Comparison of robust regression methods in linear regression. *Int. J. Contemp. Math. Sciences*, 6(9):409–421, 2011.

[2] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas. Randomized dimensionality reduction for k -means clustering. *IEEE Transactions on Information Theory*, 61:1045–1062, 2015.

[3] G. Cheng, Z. Zhang, M. Kyebambe, and N. Kimbugwe. Predicting the outcome of nba playoffs based on the maximum entropy principle. *Entropy*, 18, 2016.

[4] Barbara Engelhardt. Lecture 16: Explore pca.

[5] Raval H. and Sivaraj A. Decoding nypd misconduct complaints via officer clustering, network analysis, and salary prediction. *COS 424 Assignment 3*, 2021.

[6] Raval H. and Sivaraj A. Importance of imputation and feature selection in predicting continuous outcomes for fragile families. *COS 424 Assignment 2*, 2021.

[7] Chris Hans. Elastic net regression modeling with the orthant normal prior. *Journal of the American Statistical Association*, 106(496):1383–1393, 2011.

[8] H. Heffernan. Using machine learning to optimize nba lineups, 2018.

[9] Georgiana Ifrim, Gökhan Bakir, and Gerhard Weikum. Fast logistic regression for text categorization with variable-length n-grams. *Bing Liu, Bing; Sarawagi, Sunita; Li, Ying: KDD 2008 : proceedings of the 14th ACM KDD International Conference on Knowledge Discovery & Data Mining, ACM, 354-362 (2008)*, 08 2008.

[10] Md Zahidul Islam, Jixue Liu, Jiuyong Li, Lin Liu, and Wei Kang. *A Semantics Aware Random Forest for Text Classification*, page 1061–1070. Association for Computing Machinery, New York, NY, USA, 2019.

[11] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[12] Nathan Lauga. Nba games data. `https://www.kaggle.com/nathanlauga/nba-games`, 2021.

[13] Wei Liu, Mortaza Jamshidian, and Ying Zhang. Multiple comparison of several linear regression models. *Journal of the American Statistical Association*, 99(466):395–403, 2004.

[14] B. Loeffelholz, E. Bednar, and K. Bauer. Predicting nba games using neural networks. *Journal of Quantitative Analysis in Sports*, 5, 2009.

[15] H. Manner. Modeling and forecasting the outcomes of nba basketball games. *Journal of Quantitative Analysis in Sports*, 12, 2016.

[16] Donald W. Marquardt and Ronald D. Snee. Ridge regression in practice. *The American Statistician*, 29(1):3–20, 1975.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] F. Thabtah, L. Zhang, and N. Abdelhamid. Nba game result prediction using feature analysis and machine learning. *Annals of Data Science*, 6:103–116, 2019.

[19] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B, Methodological*, 58(1):267–288, 1996.

[20] Wikipedia. List of nba champions— Wikipedia, the free encyclopedia, 2021.

[21] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37–52, 1987. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[22] Shuo Xu. Bayesian naïve bayes classifiers to text classification. *Journal of Information Science*, 44(1):48–59, 2018.

[23] Harry Zhang. The optimality of naive bayes. volume 2, 01 2004.

[24] T. Zovak, A. Šarčević, M. Vranić, and D. Pintar. Game-to-game prediction of nba players' points in relation to their season average. *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019.

Appendix

5.1 Group Member Contributions

With regards to contributions, Hari wrote code for classification (4.1) and regression (4.3) and contributed 4 pages to the report. Tommy wrote code for network analysis (4.2) and wrote 2 pages of the report. Hannah wrote code for clustering (4.4), ran regression for results, and wrote 2 pages of the report. Isabella ran code on her computer for regression results and wrote no pages of the report.

5.2 Figures and Tables

(a) Full graph

(b) Strongest connected component

Figure 5: **Directed graph of the full NBA data set.** The blue dots represent nodes and the black text are the team id labels on the nodes. Edges are drawn with black lines representing losing teams pointing to winning teams. There are in total 30 nodes and 870 edges. The full graph is given in the left subplot and the strongest connected component is given in the right subplot.

(a) Full graph

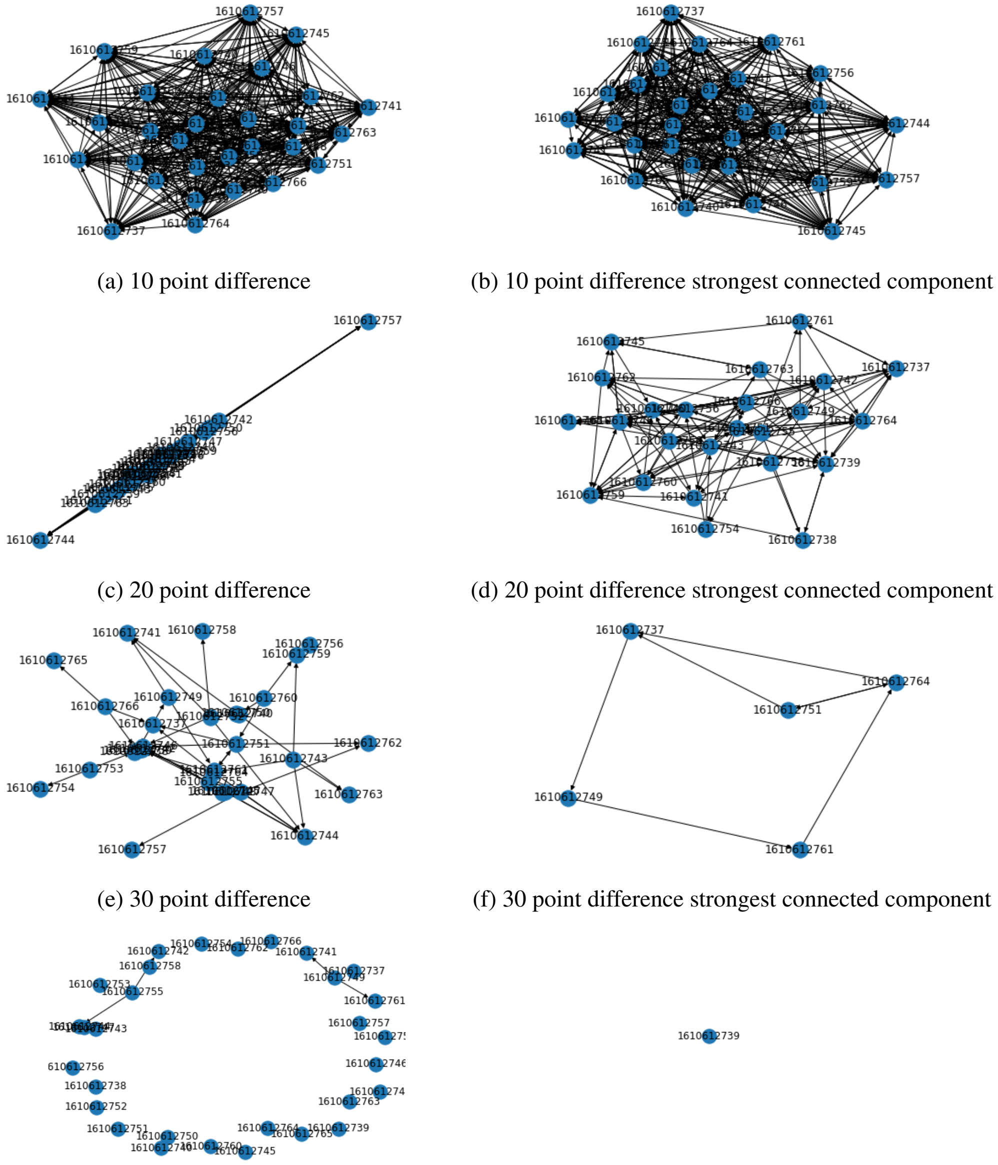
(b) Strongest connected component

Figure 6: **Directed graph of the NBA data set in year 2014.** The blue dots represent nodes and the black text are the team id labels on the nodes. Edges are drawn with black lines representing losing teams pointing to winning teams. There are in total 30 nodes and 705 edges. The full graph is given in the left subplot and the strongest connected component is given in the right subplot.

Statistics for Clusters and Player Positions

	Cluster 0	*Cluster 1*	*Cluster 2*	*Cluster 3*	*Cluster 4*	*Cluster 5*	*Cluster 6*	*Center*	*Forward*	*Guard*
FG3M	1.198617	0.166838	0.063942	0.093947	0.295311	1.407793	0.559394	0.185074	0.823327	1.206041
FTM	1.503782	0.326546	0.326546	1.209851	2.861568	3.409851	0.795771	1.591154	1.615349	1.839873
OREB	0.639293	0.223485	0.861966	1.602094	2.372371	0.814595	0.406295	2.110227	1.353523	0.648178
DREB	2.666573	0.633458	1.5595	3.08693	5.778903	3.496577	1.563566	4.377332	3.747358	2.659105
AST	2.238322	0.397547	0.442715	0.819186	1.996608	4.520577	1.405253	1.206099	1.66132	3.574574
STL	0.789191	0.172045	0.30474	0.528826	0.787366	1.173355	0.531602	0.600314	0.821391	1.026593
BLK	0.280673	0.075545	0.325121	0.687057	1.219247	0.348789	0.168021	1.025017	0.532555	0.229603
TO	1.32698	0.351445	0.351445	1.02165	1.848325	2.348907	0.869963	1.291222	1.30391	1.811484
PF	1.990164	0.644213	1.493716	2.356234	2.796177	2.170907	1.428463	2.879845	2.445433	2.183082
PTS	9.917213	1.765236	3.107107	6.334242	13.774074	16.589198	5.06495	9.012065	10.122355	11.518957
PLUS_MINUS	- 0.497880	-0.95722	-0.977667	-1.022577	0.968457	0.791213	-1.234447	-1.512242	-1.926969	-2.37839
FGL	4.863924	1.162016	1.495847	2.58559	5.165939	7.535365	2.815832	3.481153	4.774745	5.819551
FG3L	2.146322	0.438006	0.18025	0.24039	0.591125	2.564437	1.207497	0.382477	1.575081	2.265153
FTL	0.404964	0.137736	0.412472	0.649158	1.219608	0.857508	0.276681	0.783926	0.555974	0.490417

Table 7: Average statistics for each cluster versus actual NBA positions across all years.



(a) 10 point difference

(b) 10 point difference strongest connected component

(c) 20 point difference

(d) 20 point difference strongest connected component

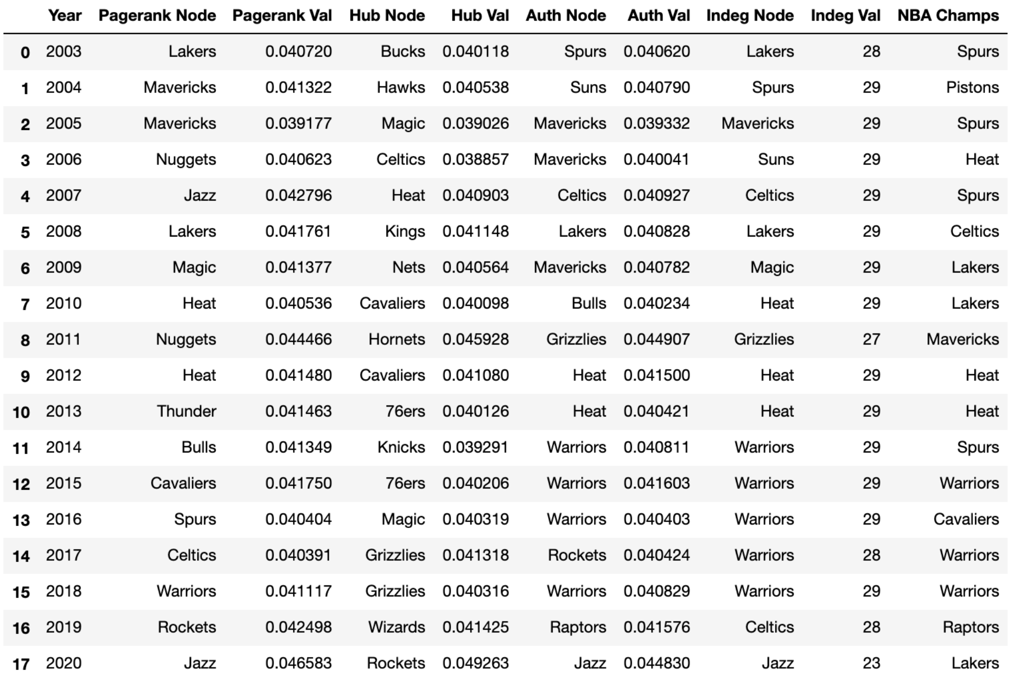
(e) 30 point difference

(f) 30 point difference strongest connected component

(g) 40 point difference

(h) 40 point difference strongest connected component

Figure 7: **Directed graphs of the NBA data set in year 2014 with edges only appearing when a certain point difference is achieved.** The blue dots represent nodes and the black text are the team id labels on the nodes. Edges are drawn with black lines representing losing teams pointing to winning teams. Edges are only drawn when the point difference between the two teams reaches a certain threshold given in the caption of each subplot. The full graph is given in the left subplot and the strongest connected component is given in the right subplot.



	Year	Pagerank Node	Pagerank Val	Hub Node	Hub Val	Auth Node	Auth Val	Indeg Node	Indeg Val	NBA Champs
0	2003	Lakers	0.040720	Bucks	0.040118	Spurs	0.040620	Lakers	28	Spurs
1	2004	Mavericks	0.041322	Hawks	0.040538	Suns	0.040790	Spurs	29	Pistons
2	2005	Mavericks	0.039177	Magic	0.039026	Mavericks	0.039332	Mavericks	29	Spurs
3	2006	Nuggets	0.040623	Celtics	0.038857	Mavericks	0.040041	Suns	29	Heat
4	2007	Jazz	0.042796	Heat	0.040903	Celtics	0.040927	Celtics	29	Spurs
5	2008	Lakers	0.041761	Kings	0.041148	Lakers	0.040828	Lakers	29	Celtics
6	2009	Magic	0.041377	Nets	0.040564	Mavericks	0.040782	Magic	29	Lakers
7	2010	Heat	0.040536	Cavaliers	0.040098	Bulls	0.040234	Heat	29	Lakers
8	2011	Nuggets	0.044466	Hornets	0.045928	Grizzlies	0.044907	Grizzlies	27	Mavericks
9	2012	Heat	0.041480	Cavaliers	0.041080	Heat	0.041500	Heat	29	Heat
10	2013	Thunder	0.041463	76ers	0.040126	Heat	0.040421	Heat	29	Heat
11	2014	Bulls	0.041349	Knicks	0.039291	Warriors	0.040811	Warriors	29	Spurs
12	2015	Cavaliers	0.041750	76ers	0.040206	Warriors	0.041603	Warriors	29	Warriors
13	2016	Spurs	0.040404	Magic	0.040319	Warriors	0.040403	Warriors	29	Cavaliers
14	2017	Celtics	0.040391	Grizzlies	0.041318	Rockets	0.040424	Warriors	28	Warriors
15	2018	Warriors	0.041117	Grizzlies	0.040316	Warriors	0.040829	Warriors	29	Warriors
16	2019	Rockets	0.042498	Wizards	0.041425	Raptors	0.041576	Celtics	28	Raptors
17	2020	Jazz	0.046583	Rockets	0.049263	Jazz	0.044830	Jazz	23	Lakers

Figure 8: **Table of maximum pagerank, HITS, and indegree nodes and values for each year from the created digraph.** The NBA Champs column corresponds to the team that won the NBA championship for the given year [20].

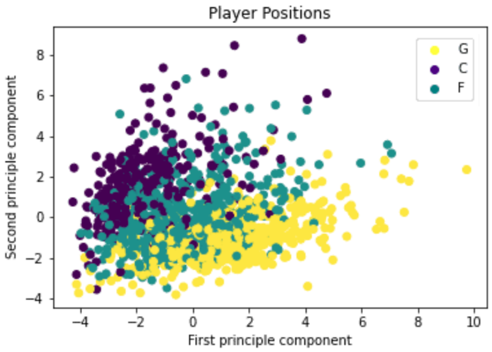


Figure 9: **Visualization of data along first two principal components, colored by player's official team position.** Note that only 1,400 players' positions were available, therefore this plot represents a subset of our data.