Using Segmentation Analysis to Predict Winning MLB Teams

By

Steven Martinez, James Keegan, and Joseph Annand[1]

May 11, 2025

[1] *Each author contributed equally the design, coding & development, analysis, and writing of this project*

Elevator Pitch

In order to predict the outcome for an MLB team in a given season, two classification models were created. The first model is more specific in its predictions by distinguishing between teams that miss the playoffs, make the playoffs but not the World Series, lose in the World Series, and win the World Series. The second model is excellent at predicting if a team will make the playoffs or not while also providing specific team statistics that are most important in deciding if a team will succeed.

Introduction

Millions of people participate in sports betting with the number of users of popular sports betting websites, like FanDuel, growing each year (Statista Research Department, 2024). Bettors are able to put money on the line when predicting which teams will make the playoffs and win championships. Baseball is one of the oldest sports that has been consistently played at the professional level. The World Series, the championship series for Major League Baseball, has been played since 1903. Since before then, statistics for player and team performance have been recorded (National Baseball Hall of Fame). In recent years, advanced metrics for baseball have become popular and are readily available with MLB Statcast. Bettors and analysts can use this available data to inform various predictions ranging from the achievements of specific players to teams' regular season win-loss records.

Leveraging this data in a meaningful yet simple way could be invaluable to sports bettors, analysts, teams, and fans. Bettors can use this model to guide their betting decisions. Analysts can use it to inform their predictions on team success. Team

organizations can use the information to learn which team attributes are most influential in successful MLB teams and find talent that positively impacts those attributes. For these reasons, in this project, segmentation analysis and predictive modeling are explored to create a method for predicting MLB team success in a given season. The following research question and hypothesis are explored and tested using predictive modeling.

Research Question: *Can MLB team segments, derived from offensive and defensive statistics, help predict whether an MLB team will make the playoffs, win the league pennant, or win the World Series?*

Hypothesis: Specific offensive and defensive statistics as well as additional information regarding payroll management and home game ticket sales will contribute to accurate classification of teams by their success because statistics are indicators of overall team ability and team morale.

Prediction: Certain offensive and defensive statistics, like On-base percentage (OBP) and Walks and hits per innings pitched (WHIP), that represent a team's ability to create scoring opportunities or prevent the opposing team from creating scoring opportunities will be most influential in predicting a team's outcome for the season. Other statistics, like batting average and home runs, that reflect individual player ability will be less influential in the model. We believe that these segments that are based on team offensive and defensive statistics will be most important in predicting which teams will win the most.

Data

In order to answer this question, it was necessary to acquire reliable and relevant data. Therefore, the data that was used was taken from a very accurate baseball statistics website called Baseball Reference. There were nine different datasets that were taken from this website, and these datasets were: standard batting data, standard pitching data, standard fielding data, player value data for batting and pitching, advanced batting and pitching statistics, sabermetrics data, and miscellaneous data. The standard batting, pitching, and fielding data had basic team statistics for each team in each given season that can tell us how a team performed at a general level. The player value data for batting and pitching data along with the sabermetrics data gave us advanced statistics that could give us a deeper understanding of the players' performance in that season. Finally, the miscellaneous data gave us statistics like payroll and attendance that do not represent team and player performance, but can be impactful when it comes to team success. This data was a very good choice because it offered information about offensive, defensive, and miscellaneous statistics that told a story about what happened with each team in each year. Even though these statistics were very useful, there was still no variable that could be used as the target variable for team success. Because of this, a multi-class team success variable was created that represented how each team performed each year, and the different classes included missing the playoffs (1), making the playoffs and not going to the World Series (2), losing in the World Series (3), and winning the World Series (4). After creating this variable and joining all of the data sets together, there were 196 predictor variables and 750 observations.

Once the final dataset was created, the exploratory data analysis (EDA) was able to be conducted. One of the first pieces of information that was found through this process was that there was a major class imbalance within the team success variable due to there being a lot of teams missing the playoffs and only having one team win the World Series every year and one team lose the World Series every year. Due to this class imbalance, there was a new team success variable that was created where the World Series winners, World Series losers, and teams that made the playoffs were combined into one class. This meant that this alternative team success variable was binary where the two classes were missing the playoffs (0) and making the playoffs (1). Because there was a second team success variable, a second final dataset with the new team success variable was created. Both the multi-class team success variable and the binary team success variable were used in this project to compare which results were best.

Now that there were two datasets, the EDA process could continue with using tables and graphs to see the relationship between the variables. One of the most important visualizations that was found can be seen in Figure 1 where the teams were grouped by their team success variable and plotted by their median estimated payroll. This line plot shows a great relationship between how the estimated payrolls could be a great predictor of team success as most of the world series winners and runner ups had a higher estimated payroll than the other teams.

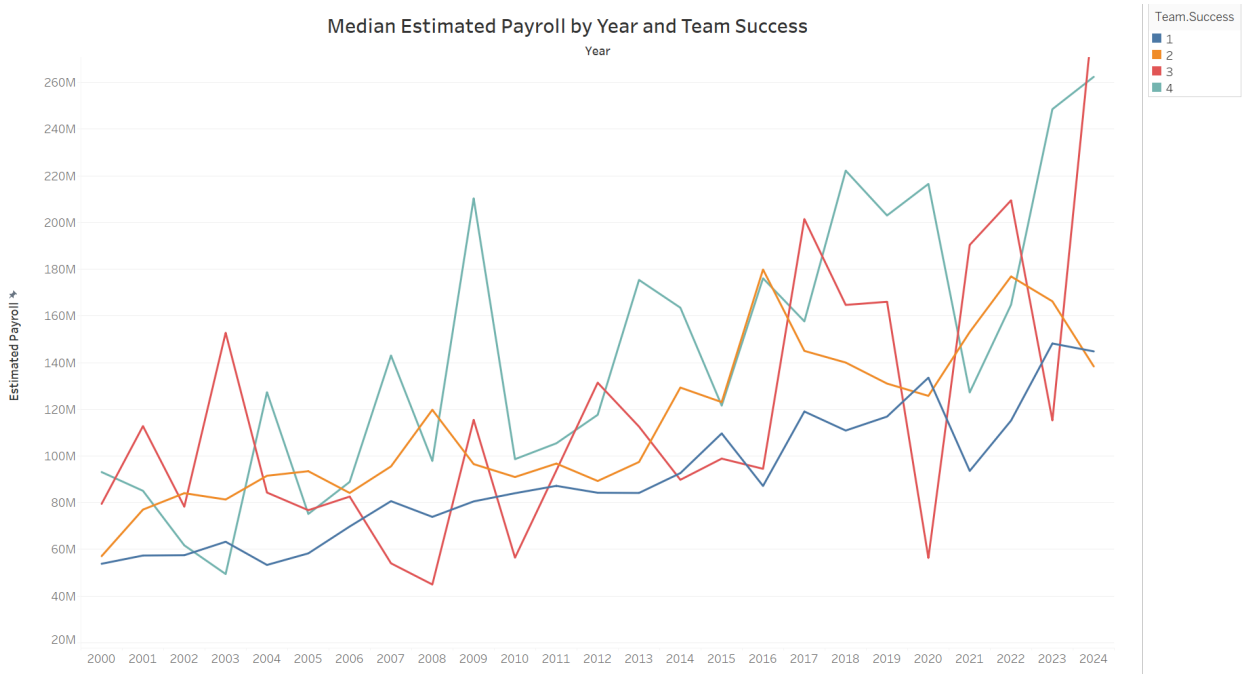

Figure 1: Median Payroll by Team Success (2000-2024)

After visually seeing the relationships of these variables, a principal component analysis (PCA) and a k-means clustering were conducted which presented a lot of great information. First, the PCA was conducted for both the multi-class variable and the binary variable where Figure 2 and Figure 3 show the distribution of the first two principal components with colors demonstrating their team success. In Figure 2, it can be seen that there is a split around 0 of PC1 where most teams that missed the playoffs fell to the right of that split and most of the other classes fell on the left of the split. Similarly, in Figure 3 with the binary there is a split near -5 of PC1 where teams that missed the playoffs are to the right of the split and teams that made the playoffs are to the left of that split.

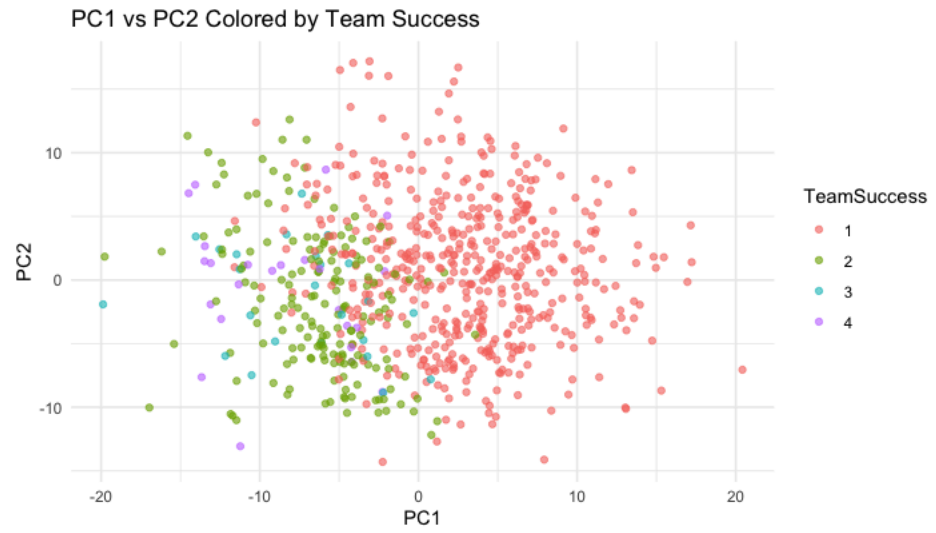

Figure 2: PCA plot of team success (four classes)

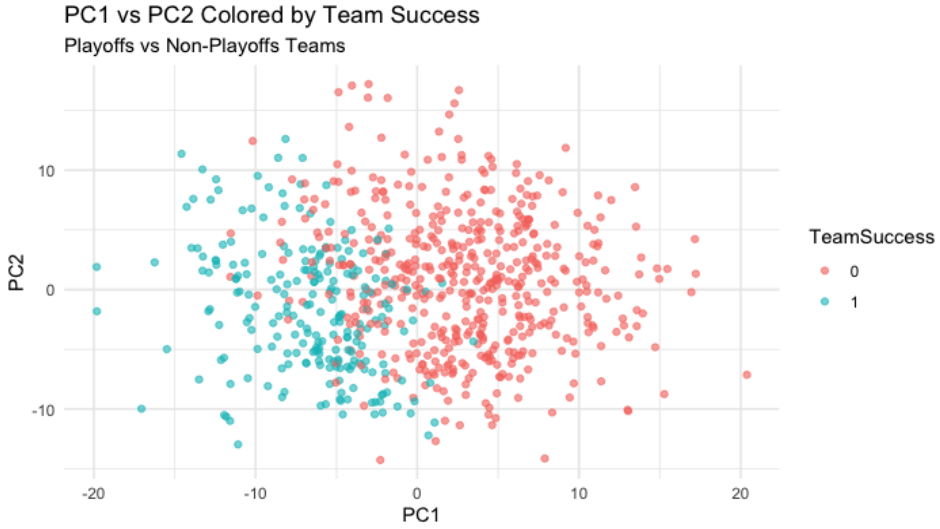

Figure 3: PCA plot of team success (two classes)

Using the rest of these principal components, the number of variables that were used was able to be reduced to 77 predictor variables and the team success variable. From this point, a k-means clustering was able to be conducted from that reduced data. Using the elbow method to find the optimal k-value, the reduced PCA data was grouped into four different clusters. Once these clusters were created, Figure 4 was made to show the distribution of the first two principal components with colors representing the

different clusters. This plot was very important to this process because of the extreme values of this data. The outliers in the figure are all from the 2020 season, which was an unusual season for the MLB as COVID-19 shortened the season and offered more teams a chance to make the playoffs. Due to all of these facts and seeing the extreme data, the year 2020 was removed from the data sets.

Figure 4: K-Means Clustering Identifying 2020 season as outlier

Finally, the part of exploring the data involved checking for multicollinearity between the variables. Looking at Figure 5, it can be seen that there seems to be some multicollinearity between the variables. This is not great because it means that there are variables that are highly correlated with each other. This could impact the results of the models that are created and needed to be dealt with. Therefore, the collinear variables were removed, and in Figure 6, the new correlation plot can be seen. This new correlation plot shows much better results and shows which variables needed to be removed in order to reduce the multicollinearity in the data.

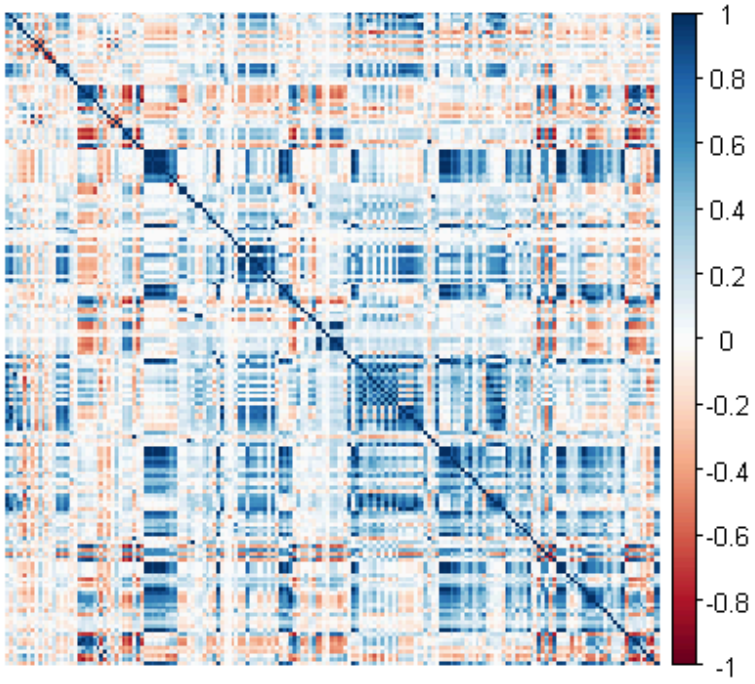

Figure 5: Initial Correlation plot using all predictors

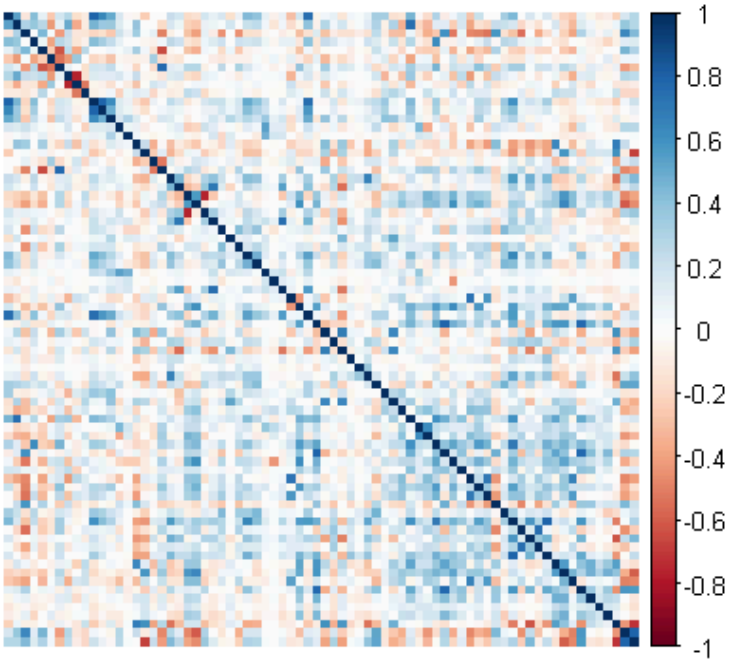

Figure 6: Correlation plot after removing collinear variables

Models

Pre-processing included PCA and LDA, identifying and removing collinear variables, removing incomplete and duplicate variables, applying mathematical transformations, and scaling variables.

Unsupervised method employed in pre-processing first was Principal Components Analysis (PCA). First 20 principal components were used as predictor variables in k-Nearest Neighbor (kNN) model. Additionally, Linear Discriminant Analysis (LDA) was performed to compare against the PCA results. Multicollinearity can affect LDA results; therefore, for the LDA, multiple sets of collinear variables were identified and no more than one predictor variable from each set was chosen to be included in further pre-processing and modeling steps. Next, a Shapiro-Wilks test was performed on the remaining variables. Non-normally distributed predictors had Box-Cox transformations applied to them. LDA did not satisfy multivariate normality assumption, but was still used for pre-processing because the data set includes numerical predictors and categorical response. An LDA model was fit using both the multiclass and binary Team Success response variable as the target, and the linear discriminants were used as predictors in kNN models.

Four kNN models were trained using dimensionality reduction data. Two kNN models used principal components and their scores from PCA. Two kNN models used the linear discriminants from LDA. For each dimensionality reduction technique, one model used the multiclass response variable and the other used the binary response variable. For each of the initial models, the data set was split into training and test sets. The models were trained using the training data. The models were trained using k-values from 1-10, and the k-value with the greatest training accuracy was chosen to

use on the test set. Using the models with the chosen k-values, predictions were made on the test set and compared to the actual values. Table 1 shows the training and test accuracies for each model using the multiclass response variable, while Table 2 includes the results from the models with the binary response variable.

Table 1: Comparison of results for kNN models using PCA and LDA data with multiclass response variable

Dimensionality Reduction Method	Training Accuracy	Test Accuracy
PCA	80.25%	79.62%
LDA	69%	84%

Table 2: Comparison of results for kNN models using PCA and LDA data with binary response variable.

Dimensionality Reduction Method	Training Accuracy	Test Accuracy
PCA	89.87%	87.34%
LDA	71%	90%

Tables 1 and 2 showed that the models using LDA performed better on the test set than the models using PCA data. For this reason, the kNN model using LDA was chosen for further tuning.

Stratified 10-fold cross-validation was performed for the Random Forest model and the kNN model using LDA. For each model, SMOTE was applied to the training data to address the class imbalance in the response variable. kNN model using LDA provided low interpretability because using the linear discriminants as the predictors obscures how individual team statistics influence the success of an MLB team. Best k-value, or number of neighbors, was determined for the kNN model.

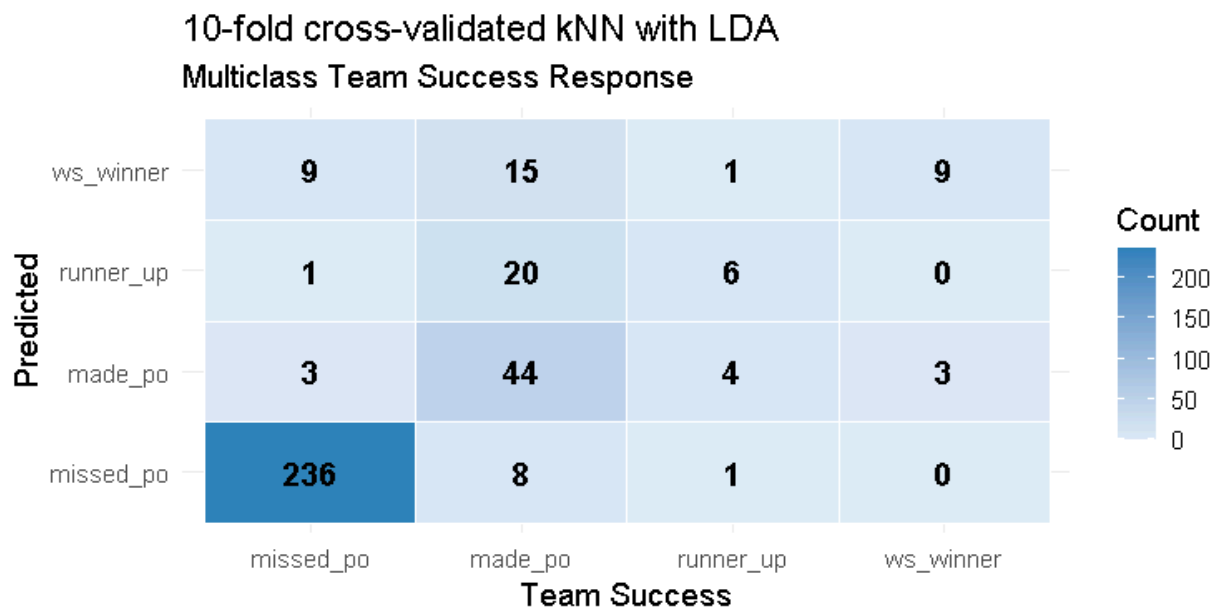

Figure 7: Confusion matrix for cross-validated kNN model using LDA predictors and multiclass response variable.

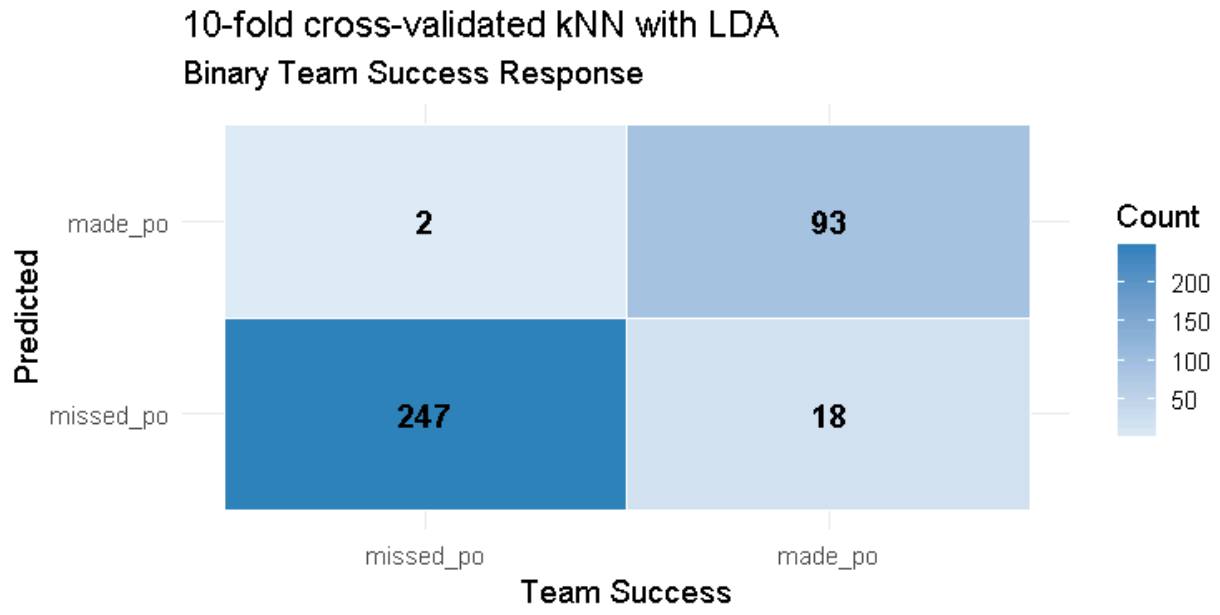

Figure 8: Confusion matrix for cross-validated kNN model using LDA predictor and binary class response.

Figure 7 shows the test results of the kNN model using LDA predictors and the multiclass response variable. Overall test accuracy is 81.4%. The cross-validated model correctly predicts half of the runner-ups and 75% of the World Series winners. Note that initial models without cross-validation failed to make any correct predictions for these classes. The model does, however, noticeably struggle with correctly identifying teams that made the playoffs but do not advance to the World Series. In its misclassifications of the teams in this category, though, the model rarely (8 out of 87 times) predicts that these teams missed the playoffs. Figure 8 shows the confusion matrix with the prediction results for the kNN model using the binary class response variable. The model correctly predicts that a team missed the playoffs 99.2% of the time and that a team makes the playoffs 83.8% of the time.

Figure 9 shows the ROC curves for each class in the multiclass response using the kNN model. The curves and area-under-curve (AUC) values for missed playoffs (red) and World Series winners (purple) indicate that the kNN model is an excellent classifier for these two categories. The model performed worst on the World Series runner-up class; however, the AUC value still supports that the model is an acceptable classifier for this category.

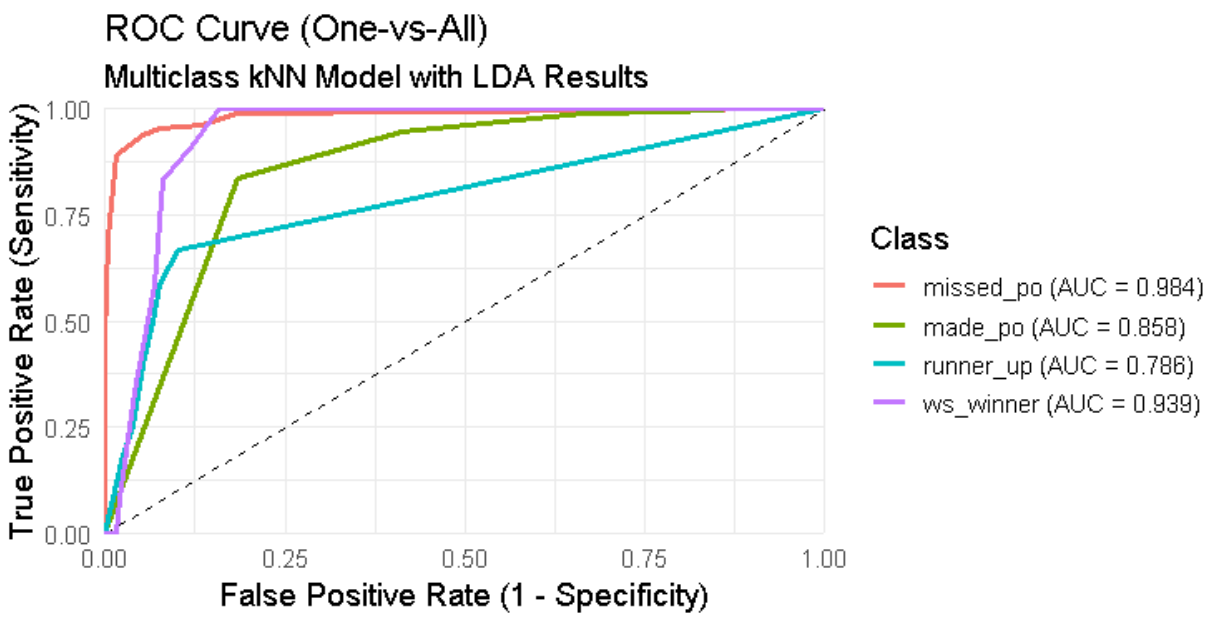

Figure 9: ROC Curve for Multiclass kNN model using LDA

Random Forest model was fit using a data set with collinear predictor variables removed to satisfy low/no multicollinearity assumption. Best number of predictors tried at each branch in the Random Forest model was determined in tuning. To begin our modeling process, we used a custom function, *calcSplitRatio* (Geist, unpublished) to determine the optimal ratio for partitioning our dataset into training and testing sets. This function considers the number of predictor variables and the size of the dataset to suggest a statistically sound split that balances model training requirements with evaluation reliability. Given that our dataset contained 78 predictor variables, the function recommended a training-to-testing split of 89:11. We adopted this ratio to ensure the model had sufficient data to learn patterns effectively while retaining a meaningful portion of data for test evaluation.

After determining the optimal split ratio, we partitioned the dataset into training and testing sets using stratified sampling to preserve the distribution of the target variable. We then focused on the training set, creating 10 stratified folds to prepare for

cross-validation. This setup ensured that model evaluation would be based on balanced, representative subsets of the data.

To address class imbalance in the dataset, we applied SMOTE exclusively to the training data within each fold of cross-validation. This approach avoided data leakage by ensuring that validation sets remained untouched and representative of real-world distributions.

We conducted 10-fold stratified cross validation, training a random forest model on the SMOTE augmented training folds and evaluating performance on the original validation folds. The random forest models were tuned by setting *mtry = 8*, aligning with the square root of the number of predictor variables, a commonly recommended starting point for classification problems.

Across the 10 folds, the model achieved an average validation accuracy of 84.43% and an average training OOB accuracy of 93.74%. The moderate gap between training and validation accuracy suggests mild overfitting, a typical outcome when applying SMOTE, but overall model generalization remained strong and consistent across folds.

Below, Figure 10 illustrates training and validation accuracy across all 10 cross-validation folds:

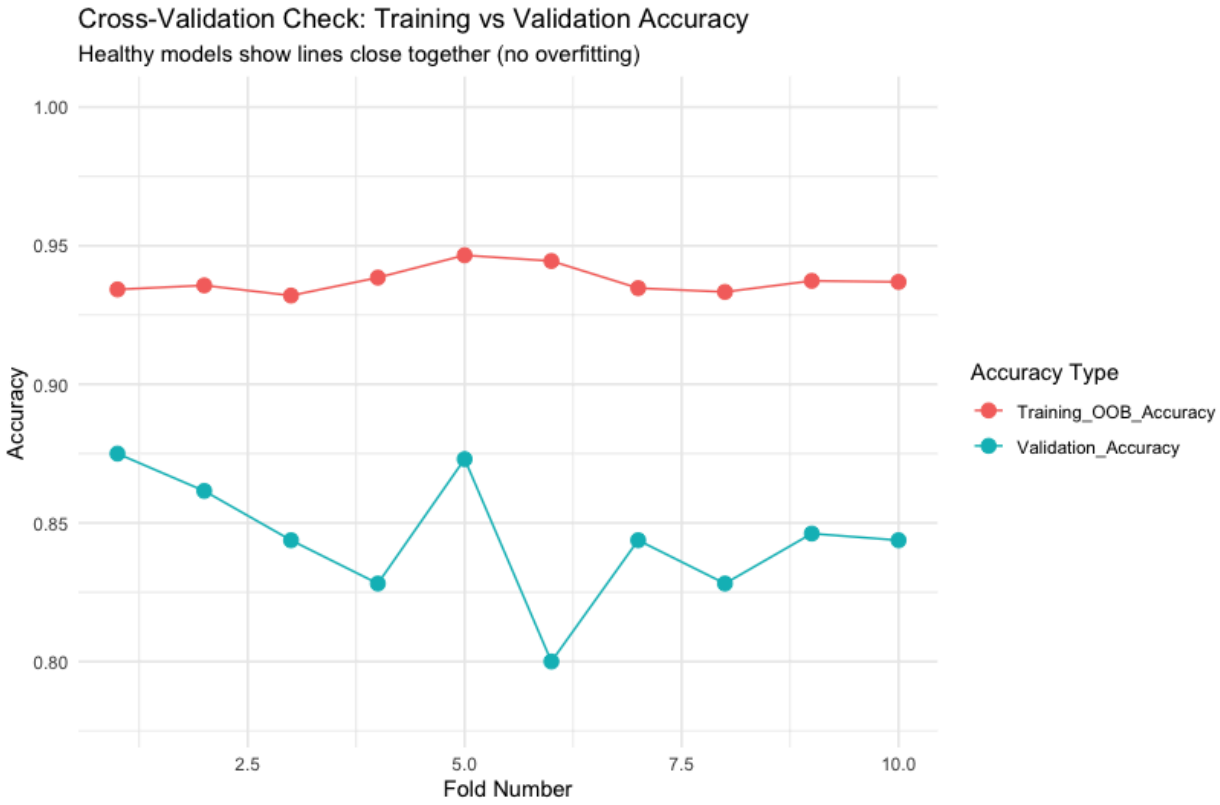

Figure 10: Training and validation accuracy across 10 cross-validation folds.

The red line represents training performance, measured using the random forest model's out-of-bag (OOB) accuracy, while the blue line shows validation accuracy on unseen data. Although the training accuracy remains consistently high (around 93–95%), validation accuracy exhibits slightly greater variability, ranging from approximately 80% to 87.5%. The gap between the two lines suggests mild overfitting, but the validation scores remain stable enough to indicate that the model generalizes reasonably well. These results support the reliability of the model, while also highlighting an opportunity for further tuning or regularization if needed.

To evaluate classification performance across all folds, we aggregated predictions from the validation sets and constructed a confusion matrix comparing predicted versus actual team success categories. As shown in Figure 11, the random forest model performed best at identifying teams in the lowest success category (Class 1), which accounted for the majority of correct predictions. Performance was also relatively strong for Class 2, though some misclassifications occurred between neighboring categories.

The distribution of predictions suggests that while the model is highly effective at distinguishing low-performing teams, it is less accurate in differentiating between middle and high-performing teams. This is expected given class imbalance and overlapping feature distributions. Overall, the confusion matrix supports the conclusion that the model captures the key patterns of team success.

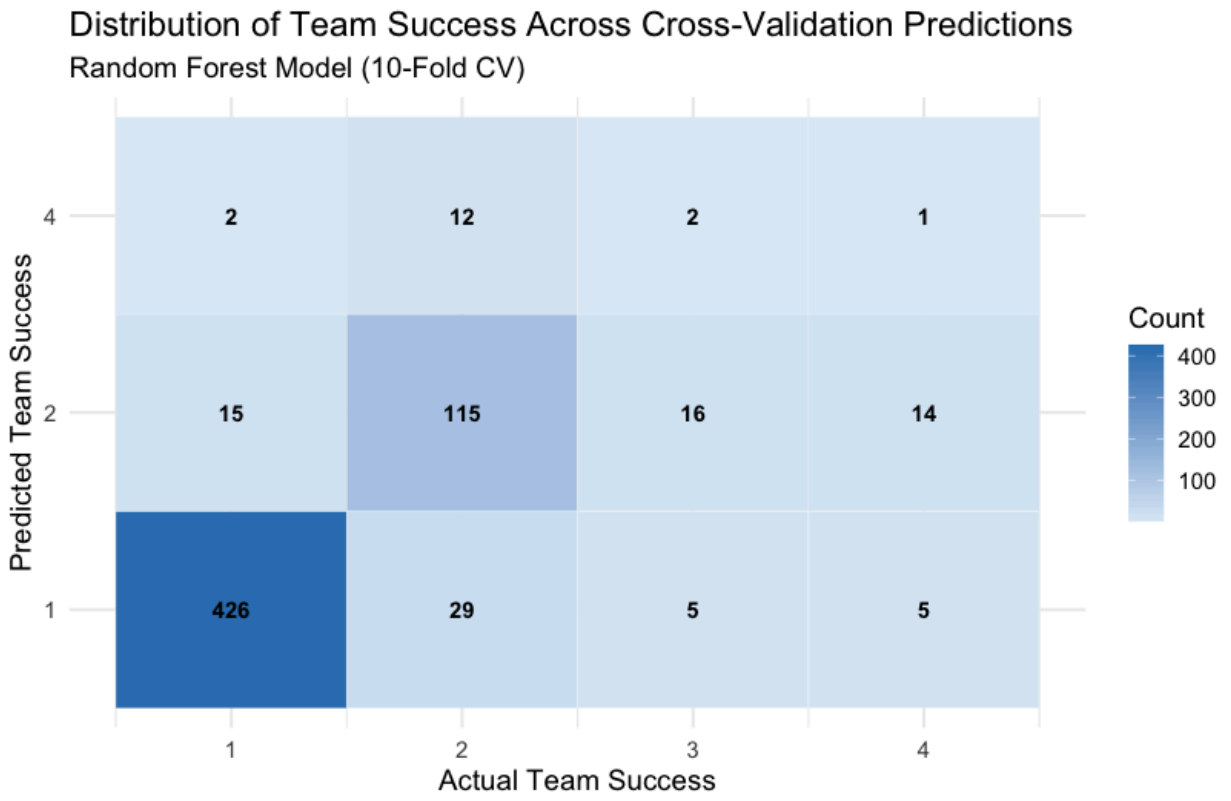

Figure 11: Confusion matrix of predicted vs. actual team success from 10-fold CV.

After completing cross-validation and model tuning, we evaluated the final random forest model on the 11% holdout test set that was not used during training. The model achieved an overall accuracy of 84.62%, with a Kappa statistic of 0.639, indicating substantial agreement between predicted and actual classes.

As shown in Figure 12, the model performed best on Class 1, correctly classifying 53 of 55 observations. Performance was reasonably strong for Class 2, while prediction accuracy declined for Classes 3 and 4 due to their low prevalence in the test set. Class 3 had no correct predictions, highlighting a limitation in the model's ability to generalize to underrepresented categories.

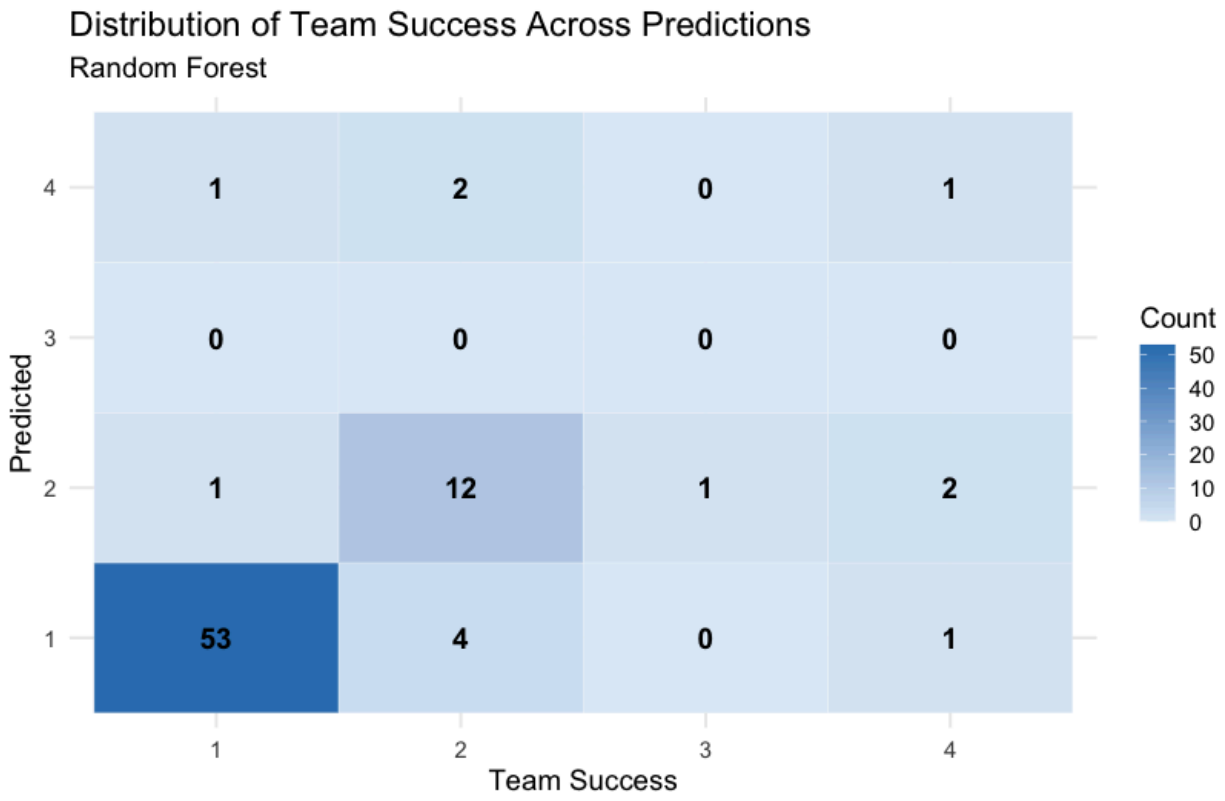

Figure 12: Confusion matrix of final model predictions on the test set.

Despite this, the model maintained good balanced accuracy across most classes, demonstrating reliable performance and generalizability when evaluated on real, unseen data.

To address poor predictive performance on Classes 3 and 4 in the four-class model, we reframed the target variable as a binary outcome: whether a team made the playoffs (1) or missed the playoffs (0). This simplified classification task allowed the model to focus on more clearly separable groups while avoiding the sparsity issues associated with World Series-level outcomes.

As shown in Figure 13, the model maintained consistently high accuracy across all 10 folds. The average training OOB accuracy was 95.45%, while the average validation accuracy was 93.29%, with only a slight gap between the two. This small

difference suggests that the model generalized well and exhibited minimal overfitting. The validation accuracy remained stable across folds, further supporting the model's robustness in binary classification.

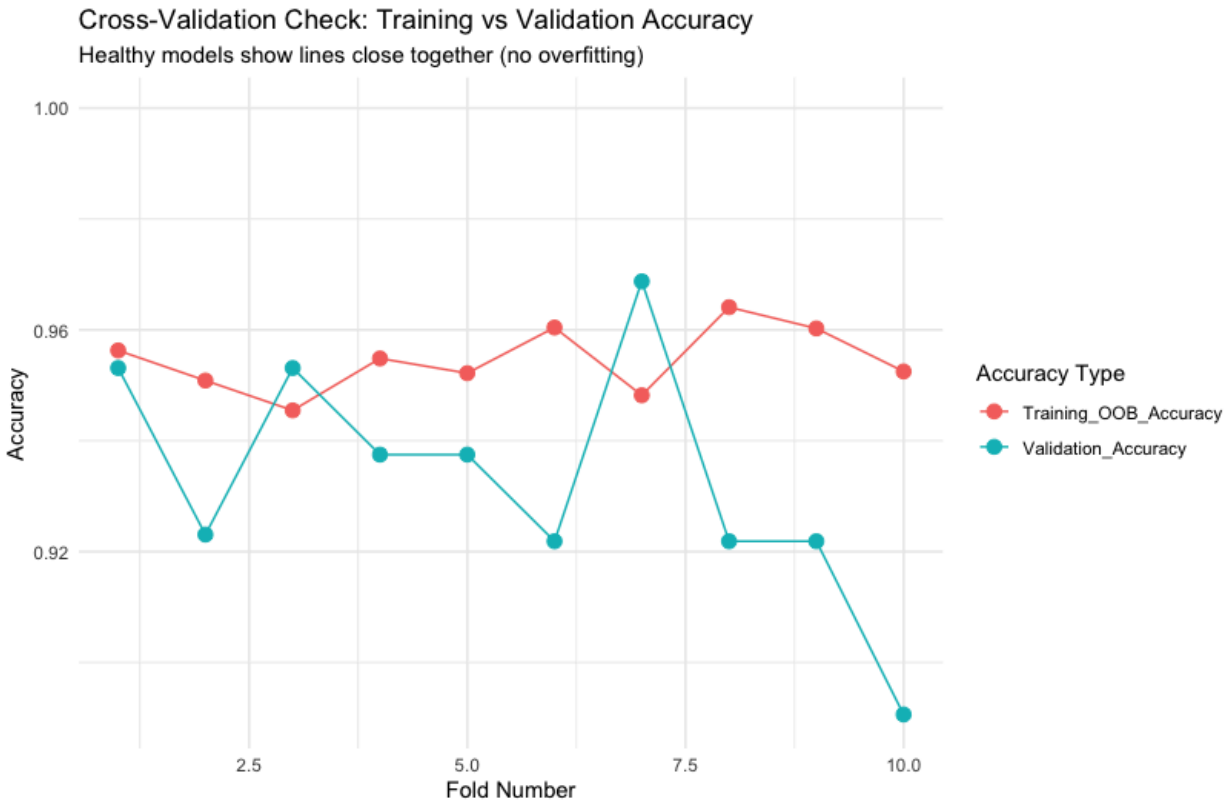

Figure 13: Training and validation accuracy across 10-fold CV (binary classification)

Figure 14 displays the confusion matrix summarizing predictions across all 10 cross-validation folds. The model performed well on both classes, correctly identifying 416 teams that missed the playoffs and 182 teams that made the playoffs.

Only 15 playoff teams were misclassified as non-playoff teams, while 28 non-playoff teams were incorrectly predicted to have made the playoffs. These results indicate that the model was effective at distinguishing between teams that made and missed the playoffs, achieving strong predictive performance across both classes. The

relatively low number of misclassifications suggests that applying SMOTE during cross-validation helped the model generalize well to the minority (playoff) class.

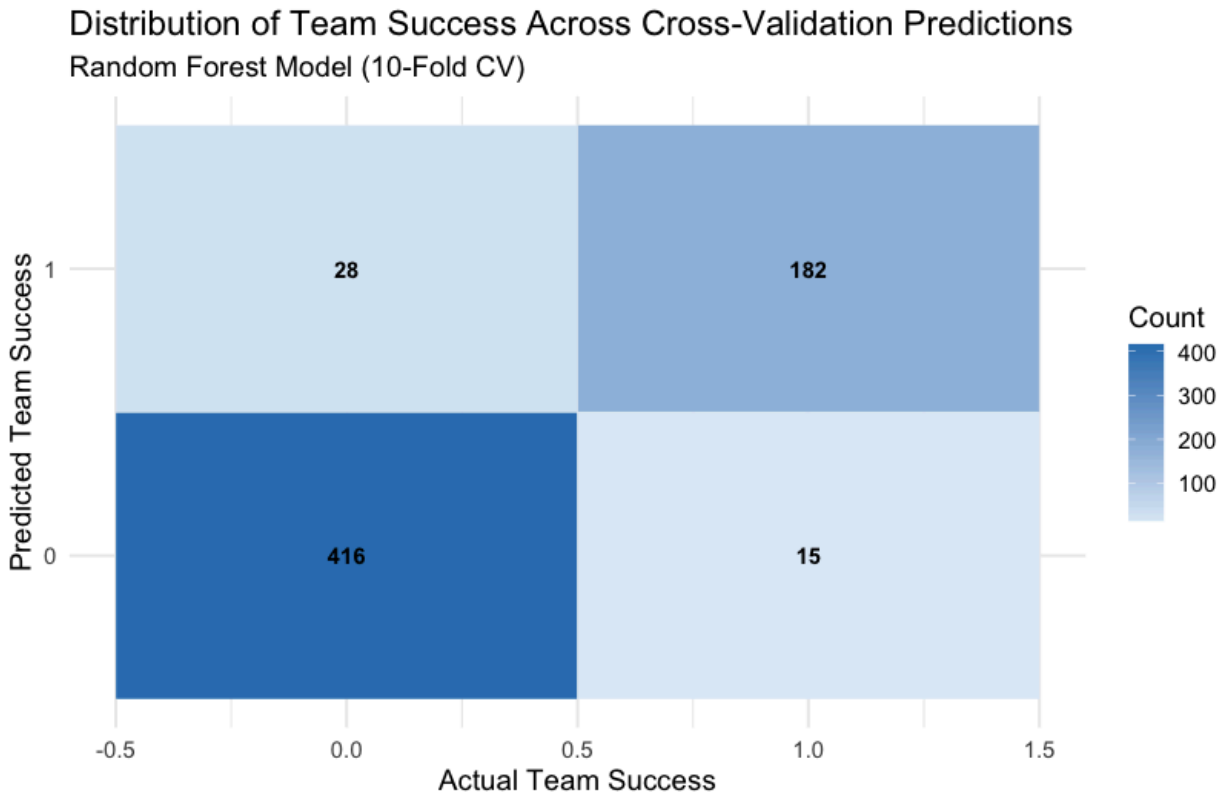

Figure 14: Confusion matrix of predicted vs. actual playoff status from 10-fold CV

After completing cross-validation, the final random forest model was evaluated on the 11% holdout test set. As shown in *Figure 15*, the ROC curve illustrates the model's diagnostic ability by plotting the True Positive Rate (Sensitivity) against the False Positive Rate (1 - Specificity) across various classification thresholds. The curve closely hugs the top-left corner, indicating strong predictive performance. The Area Under the Curve (AUC) is 0.957, which suggests that the model is highly effective at distinguishing between the two outcome classes. This high AUC value provides strong

evidence that the binary Random Forest model offers a robust and reliable classification approach for identifying successful teams.

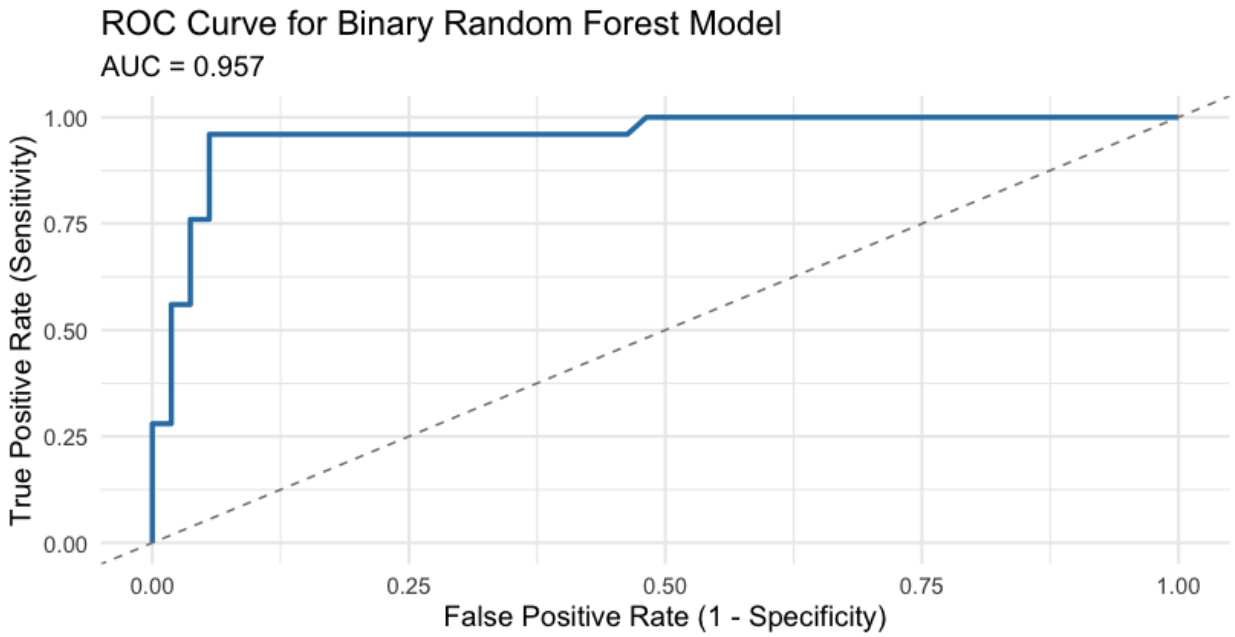

Figure 15: Binary Random Forest ROC Curve

In Figure 16, which shows the model demonstrated strong predictive performance across both classes. The model achieved an overall accuracy of 94.94% and a Kappa statistic of 0.885, indicating near-perfect agreement between predicted and actual playoff outcomes. Specifically, it correctly classified 51 non-playoff teams and 24 playoff teams, while only misclassifying 4 teams in total.

The model's sensitivity (correctly identifying non-playoff teams) was 94.44%, and its specificity (correctly identifying playoff teams) was 96%, resulting in a balanced accuracy of 95.22%. These results suggest that the model generalized extremely well to unseen data and successfully distinguished between playoff and non-playoff teams with minimal bias toward either class.

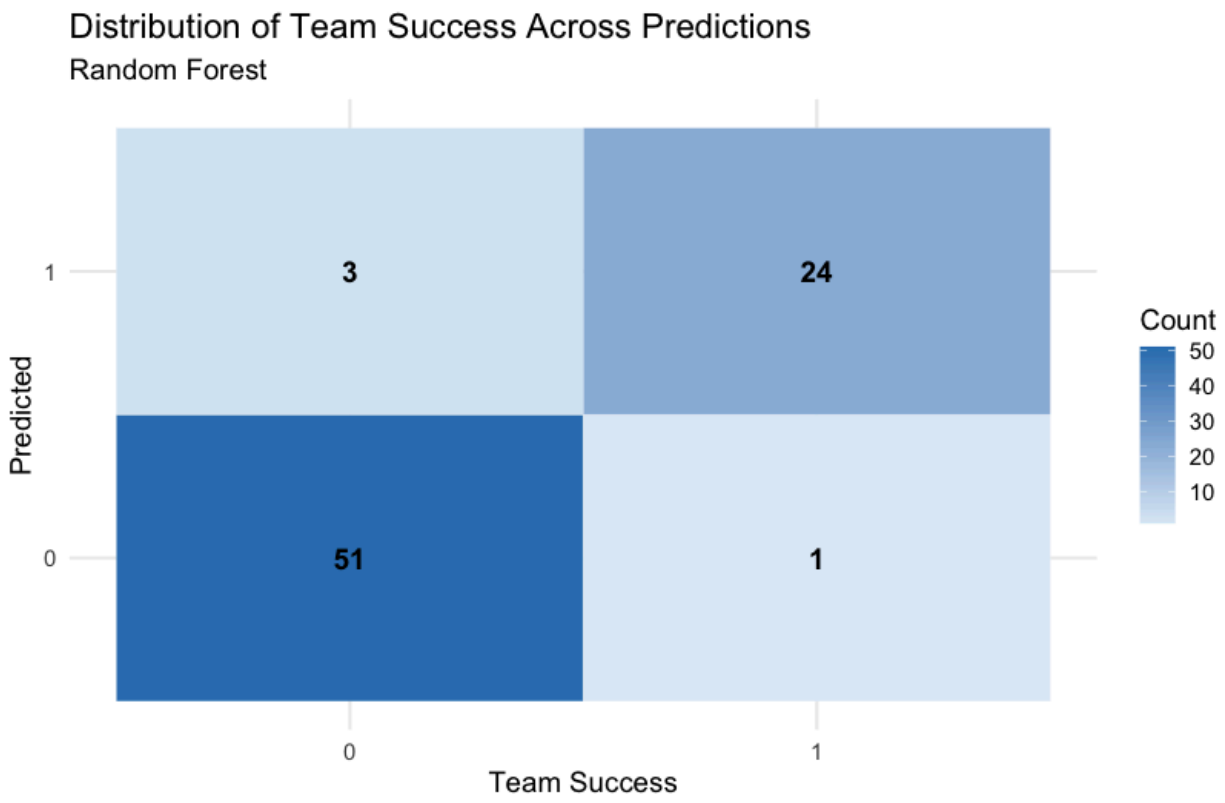

Figure 16: Confusion matrix of predicted vs. actual playoff status on the test set

To better understand the factors driving the model's high performance, we examined the top 10 most important variables based on their mean decrease in accuracy across cross-validated folds, as shown in Figure 17. The most influential predictor was **ap.cWPA** (Championship Win Probability Added from the advanced pitching dataset), followed by **ab.cWPA** (Championship Win Probability Added from advanced batting). These metrics reflect a team's contribution to championship odds through both pitching and batting, emphasizing the importance of postseason-impact metrics. Additional key features included **ap.RE24** and **ab.WPA**, which measure run expectancy and win probability contributions, respectively. Notably, **pvp.WAR** (player value pitching Wins Above Replacement) and **sbm.OPS** (on-base plus slugging from sabermetrics) also ranked highly, reinforcing the role of individual value and overall

offensive efficiency. Rounding out the list were **md.A.S** (number of All-Star players), and several standard pitching metrics—**sp.WHIP**, **sp.FIP**, and **sp.SV**—highlighting the critical influence of pitching consistency and bullpen performance in playoff success. These results suggest that both high-leverage metrics and traditional performance indicators collectively inform the model's ability to differentiate playoff and non-playoff teams.

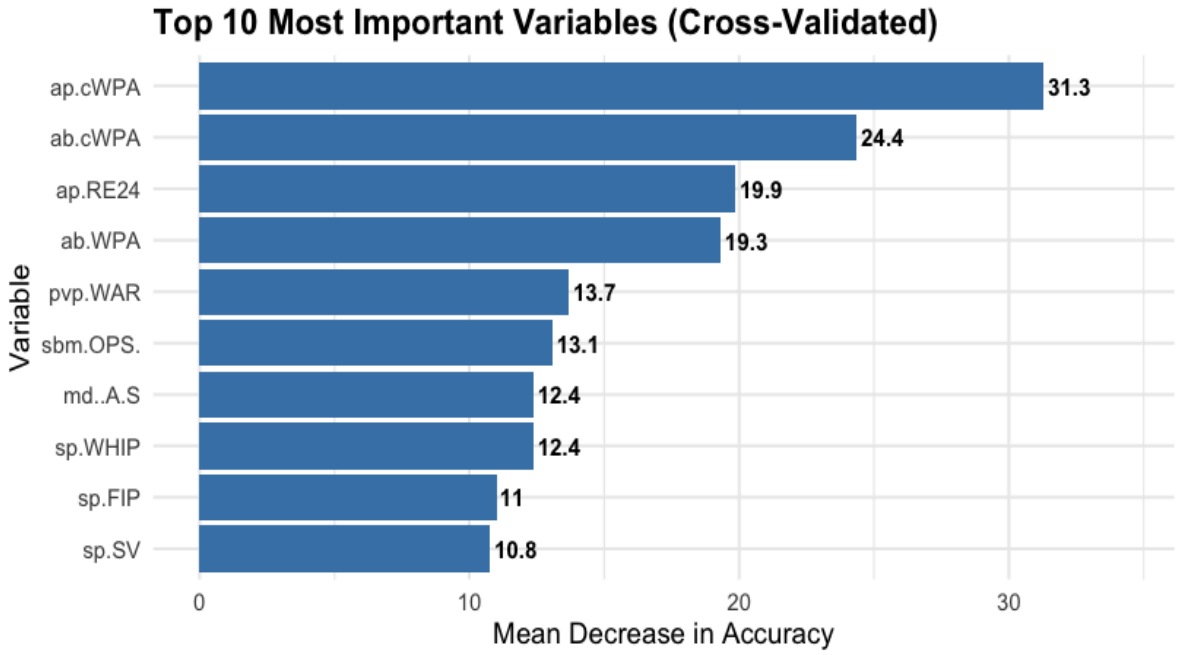

Figure 17: Top 10 most important variables from the binary Random Forest Model. Prefixes indicate metric type: ap. (advanced pitching), ab. (advanced batting), pvp. (player value), sbm. (sabermetrics), md.(miscellaneous), sp. (standard pitching).

Overall, the Random Forest model was the best model choice based on test accuracy results and interpretability. Random Forest had slightly better test accuracy than the kNN model, and the importance of each predictor variable can be calculated and visualized when ranking the variables by Mean Decrease in Accuracy.

Tables 3 and 4 compare the results of the Random Forest and kNN models using the multiclass response and the binary response, respectively. The small differences between validation and test accuracies for both models across the two response variables show that none of the models are significantly overfit. The test accuracies show that the Random Forest models have slightly better prediction accuracies overall; however, as shown in Figure 12, the Random Forest model struggles to predict two of the four classes.

Table 3: Accuracy Scores by Model from Cross-Validation and Test Accuracy with Multiclass Response Variable

Model	Validation Accuracy	Test Accuracy
Random Forest	84.4%	84.6%
k-Nearest Neighbor (LDA Dimensionality Reduction)	83.6%	81.9%

Table 4: Accuracy Scores by Model from Cross-Validation and Test Accuracy with Binary Response Variable

Model	Validation Accuracy	Test Accuracy
Random Forest	93.29%	94.94%
k-Nearest Neighbor (LDA Dimensionality Reduction)	96.9%	94.4%

Discussion & Next Steps

The Random Forest model has the advantage of ranking predictor variables by importance in a quantitative manner, thus allowing users of the model to understand

which baseball statistics are most influential in team success. Additionally, the Random Forest model has slightly better test accuracy using the binary response variable compared to the kNN model using LDA. The disadvantage of the Random Forest model is that it shows poor performance in predicting World Series winners and runner-ups. Due to its advantages and disadvantages, the Random Forest model is ideal for team executives who want to understand the most important factors to team success and use that information to inform decisions about their roster.

The kNN model using LDA dimensionality reduction correctly predicts World Series winners and runner-ups more than 50% of the time; however, it lacks the interpretability of the Random Forest. Therefore, the kNN model is ideal for bettors and analysts who want to make more precise predictions on an MLB team's success in a given season without needing the knowledge of the most influential factors to success.

Segmentation helped in data cleaning and feature engineering, but classification algorithms ultimately provided the ability to predict MLB team success. The Random Forest fails to accurately predict WS winners and runner-ups, while the kNN model is better able to make predictions with the targeted granularity. Accurate models were created using a variety of baseball statistics. According to the binary Random Forest model; player value statistics, OPS and WHIP were most important in predicting if a team made the playoffs or not.

Some caveats regarding this project arise from the amount of data used, the methods used in training, and feature selection. The data for this project included only 24 MLB seasons since the year 2000, which yielded only 720 observations. Baseball has a long history and more data is available that could be used. Specifically, seasons

in the late 1990s where analytics had just begun to be utilized could be insightful and add value to the models. Additionally, the stratified k-fold cross-validation was not time-aware, meaning that the training did not account for discrepancies in the data across time. Finally, in feature selection, a subjective method for excluding collinear variables was employed. It could be possible that the models would yield different results with a different subset of predictors.

It is recommended that the next steps for this project include the following actions. Initially, an exploratory analysis of the current models' incorrect predictions across time should be performed to identify bias related to time. If that is observed, a time-aware cross-validation should be employed during training to correct the bias. Additionally, a more objective approach to feature selection of collinear variables should be performed to identify if using different subsets of predictors yields significantly different model results. The kNN model using LDA had the disadvantage of having less interpretability than the Random Forest. The team should explore ways to extract feature importance from LDA results. Ultimately, the models should be tested using 2025 season data, then the models can be evaluated by comparing its predictions for this season to the outcome in October of this year.

Code Availability

GitHub Repository: Team beta GitHub Repo.

Appendix

Files

Data Dictionary: Team Beta Data Dictionary

Visualizations

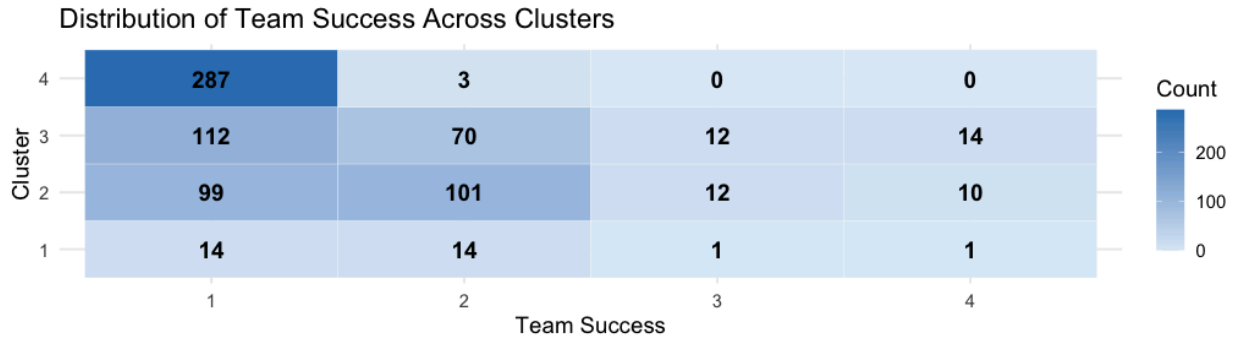

Figure 1A: Frequency Distribution of Team Success Across K-means clusters

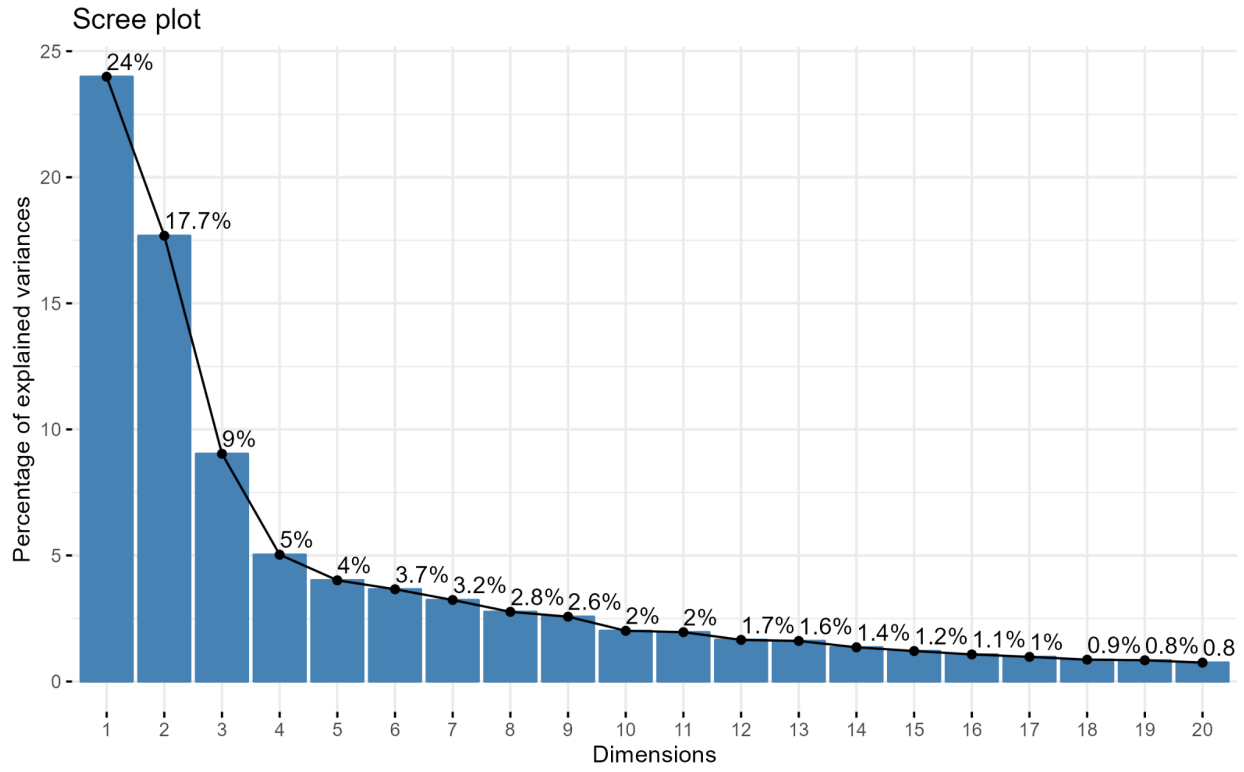

Figure 2A: Scree plot for PCA supporting the use of 20 principal components.

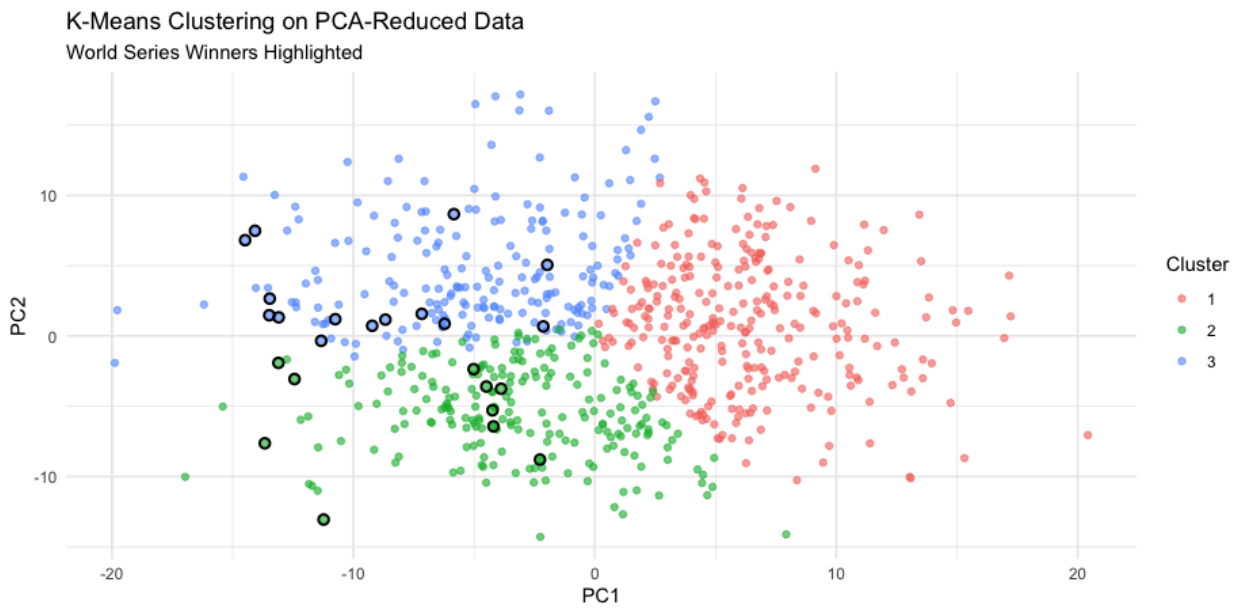

Figure 3A: K-means Clustering without 2020 season data

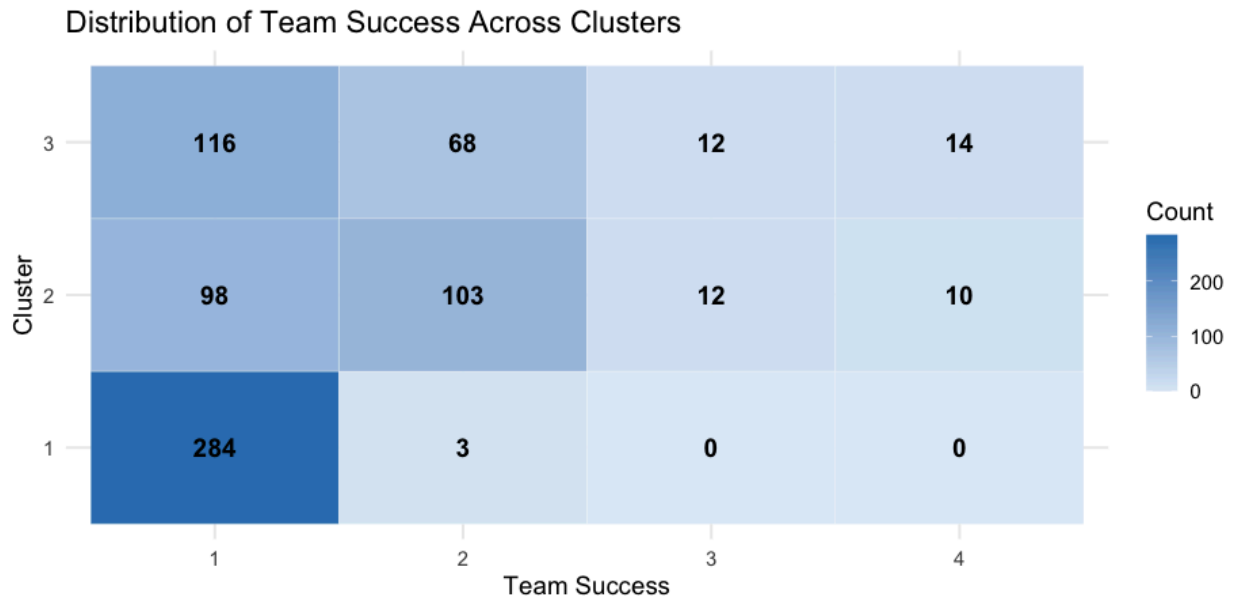

Figure 4A: Frequency Distribution of Team Success in k-means clusters without 2020 season data.

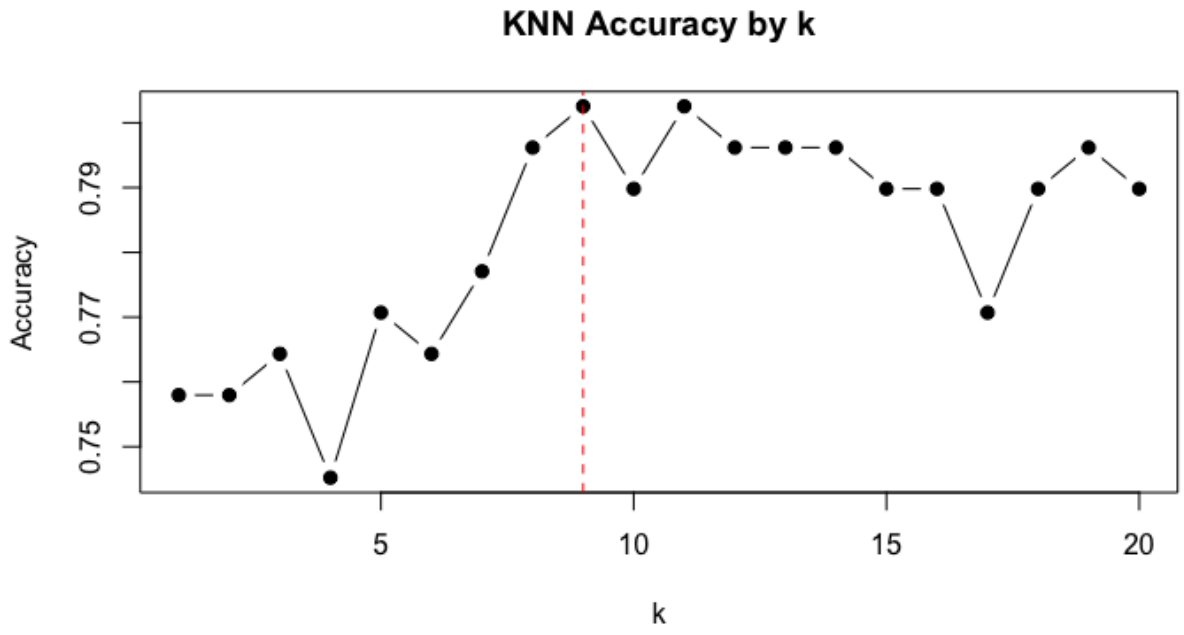

Figure 5A: Accuracy of kNN model using PCA results across different k-values.

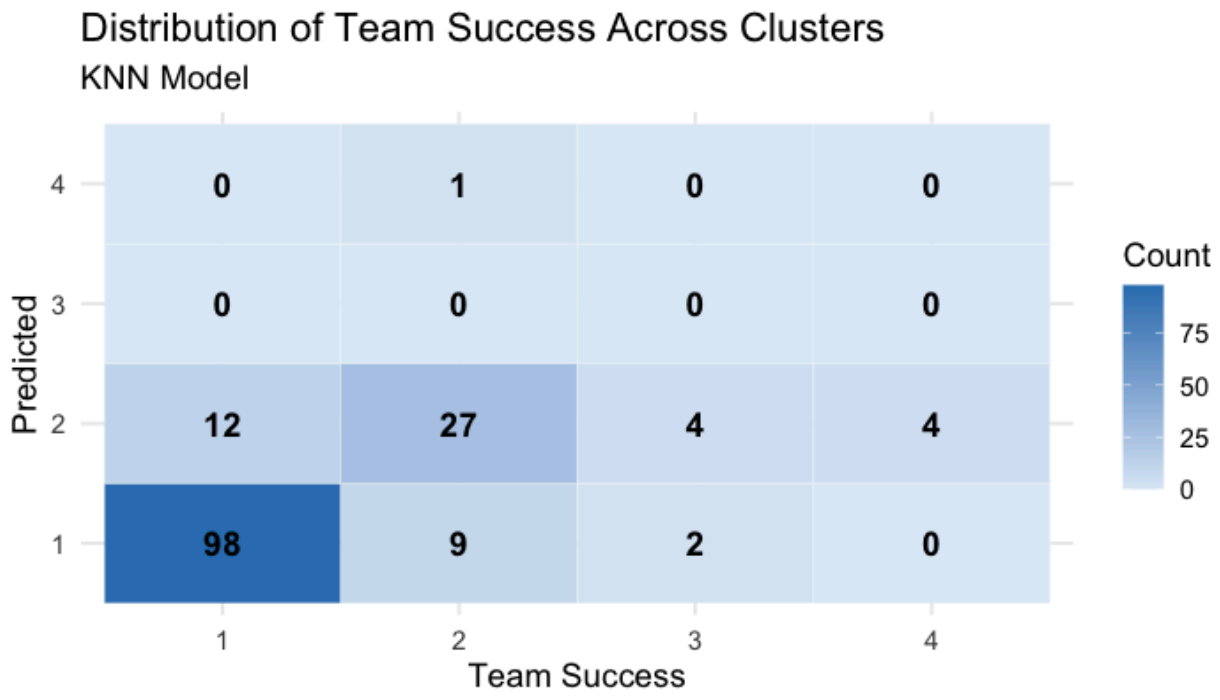

Figure 6A: Confusion matrix of kNN model using PCA results predictions using four-class team success.

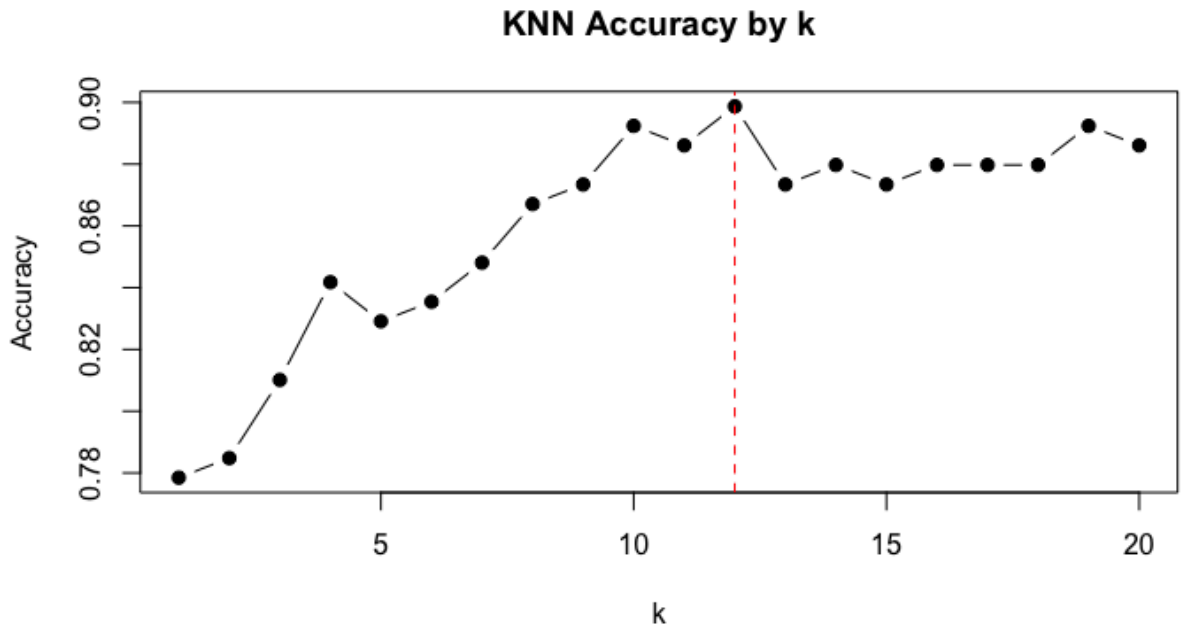

Figure 7A: Finding optimal k-value for kNN model using PCA results and Binary Response.

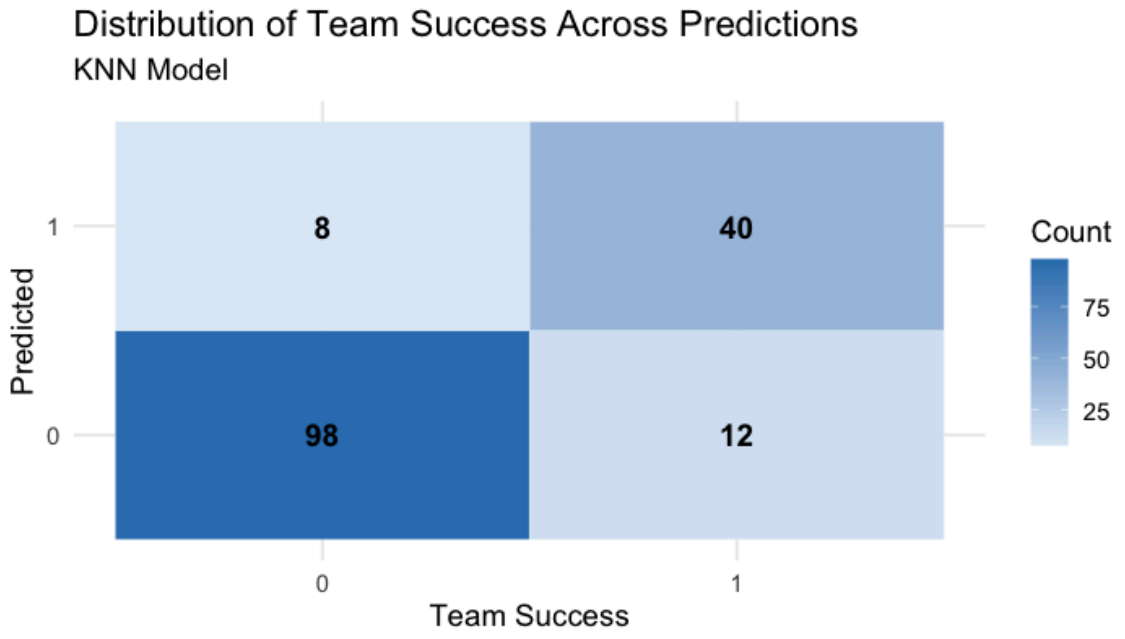

Figure 8A: Confusion matrix for kNN prediction results on test set using PCA results and binary response

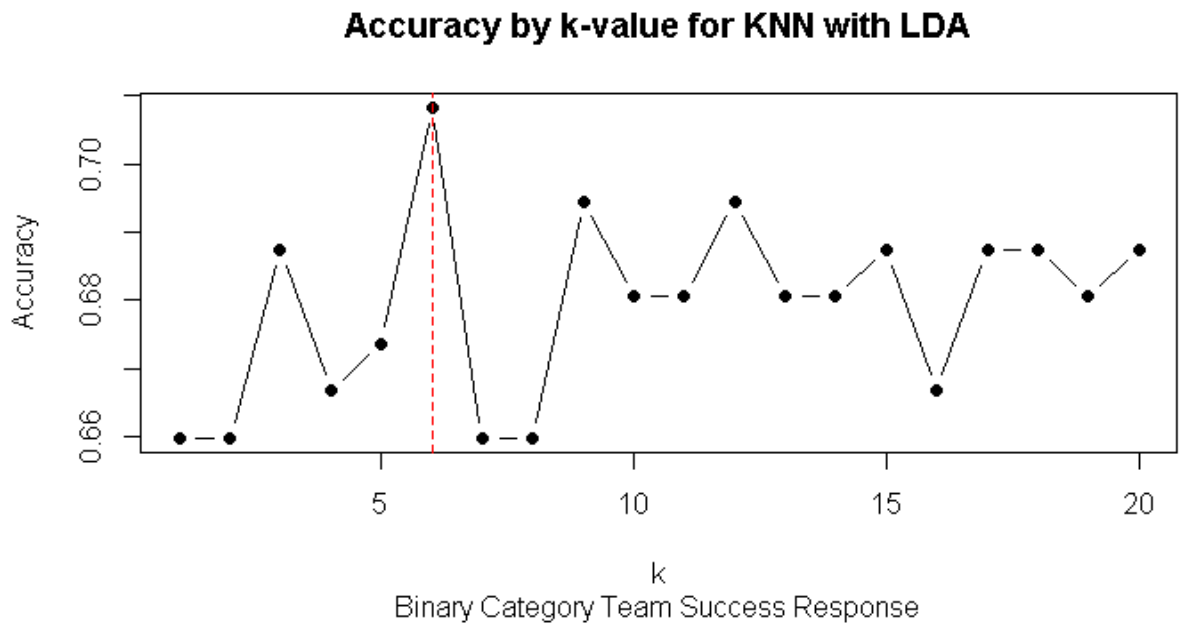

Figure 9A: Finding optimal k-value for kNN model using LDA results and Binary Response.

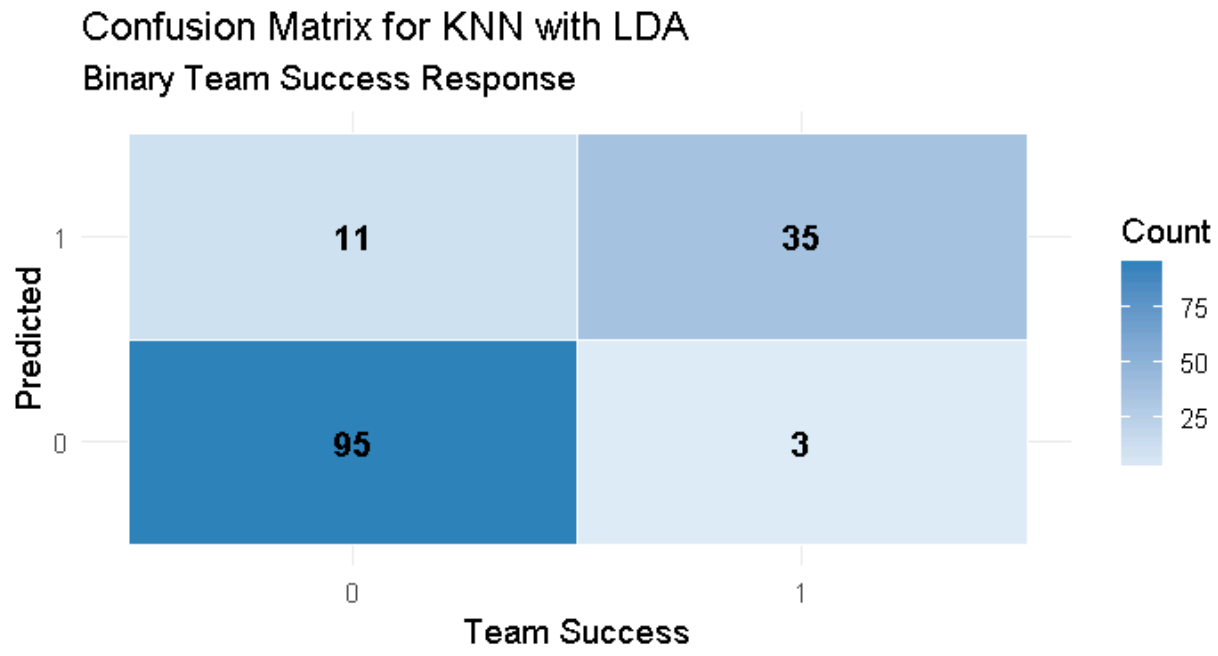

Figure 10A: Confusion matrix for kNN prediction results on test set using LDA results and binary response before using cross-validation.

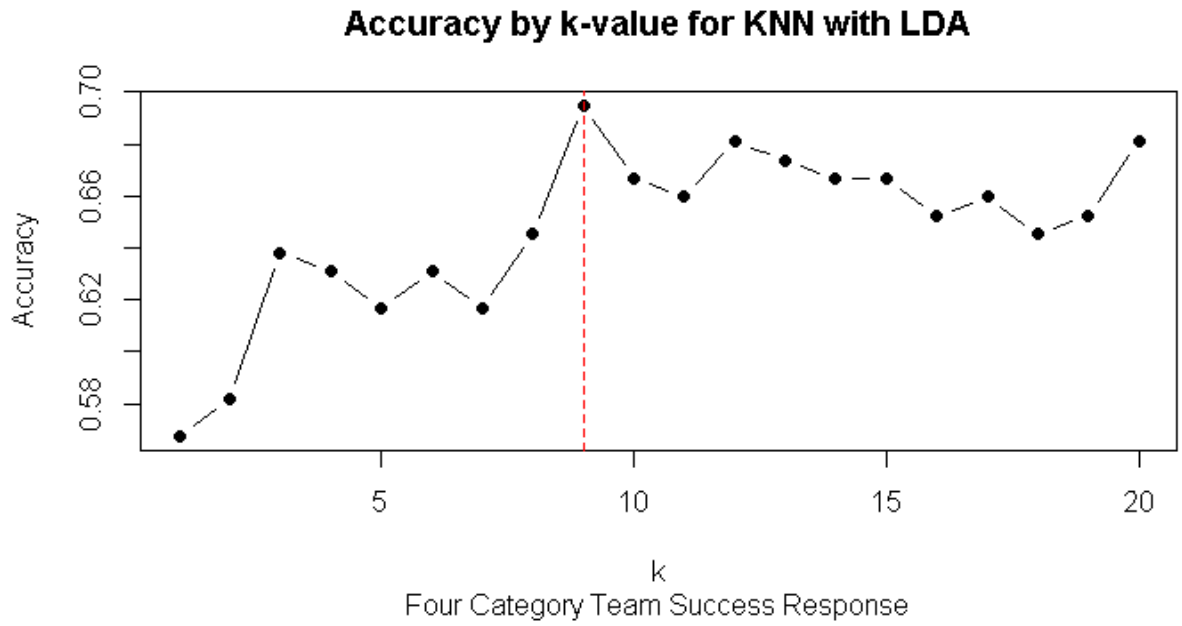

Figure 11A: Finding optimal k-value for KNN model using LDA results and Four-Category Response without cross-validation.

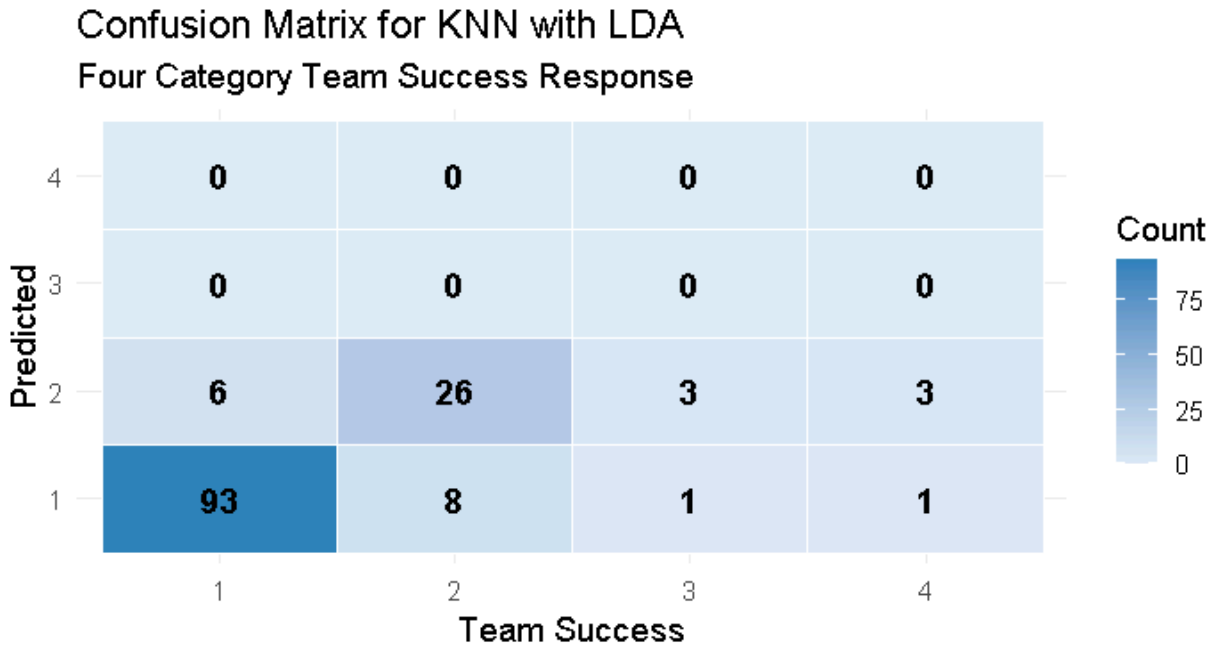

Figure 12A: Confusion matrix for kNN prediction results on test set using LDA results and binary response before using cross-validation.

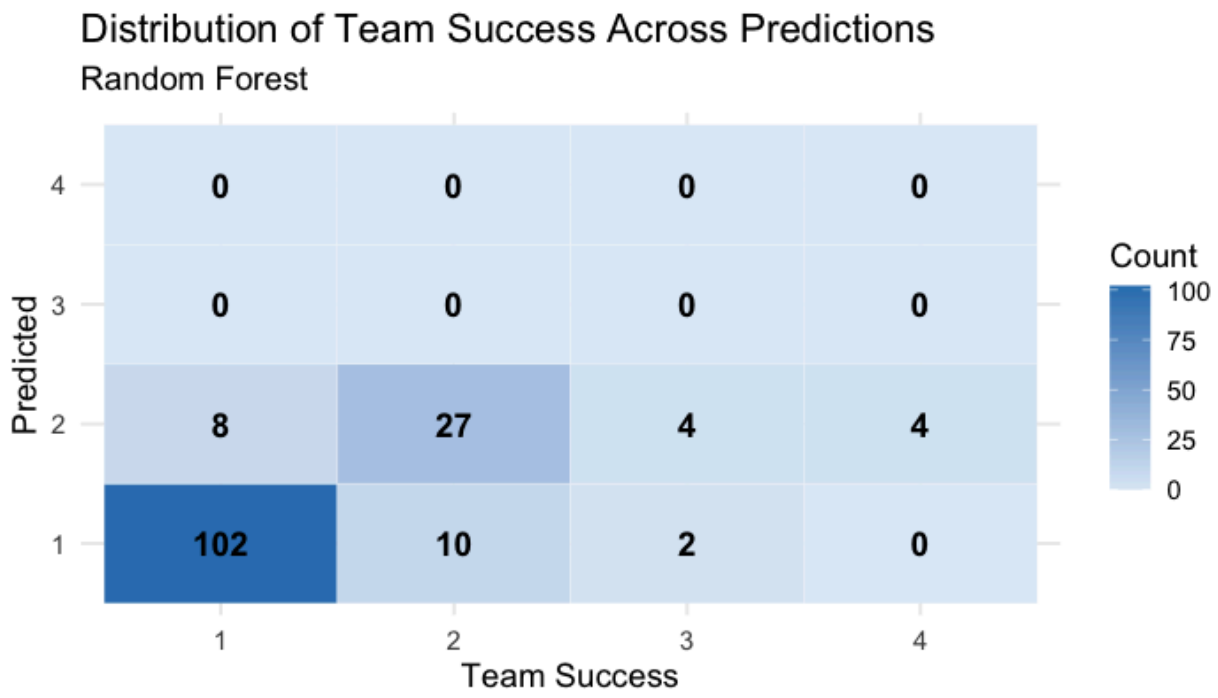

Figure 13A: Confusion matrix for Random Forest prediction results on test set using PCA results and four-category response

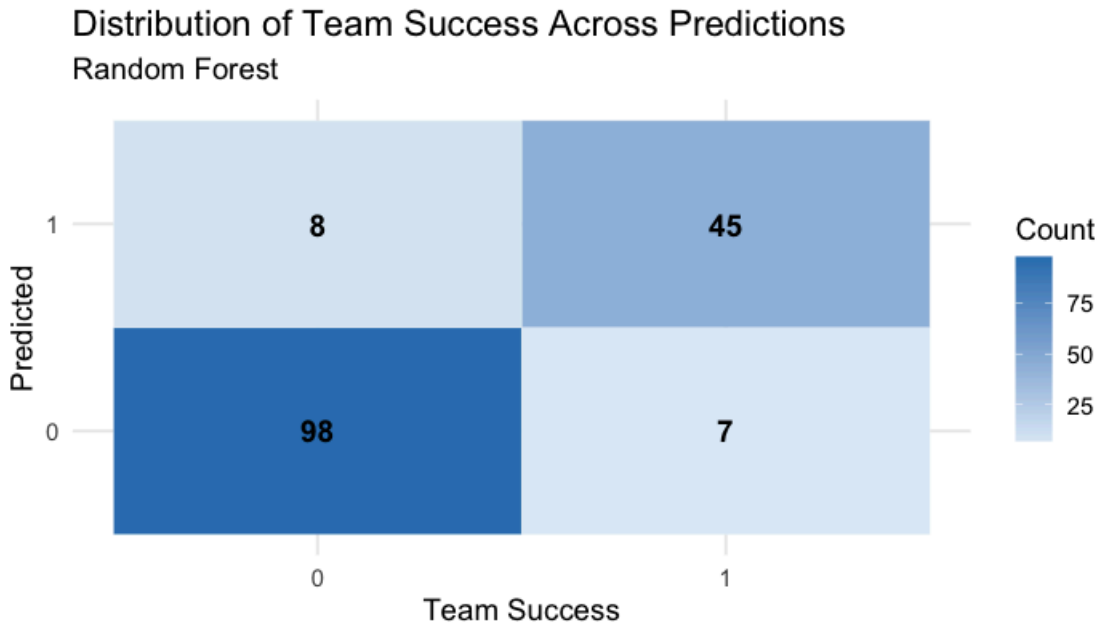

Figure 14A: Confusion matrix for Random Forest prediction results on test set using PCA results and binary response

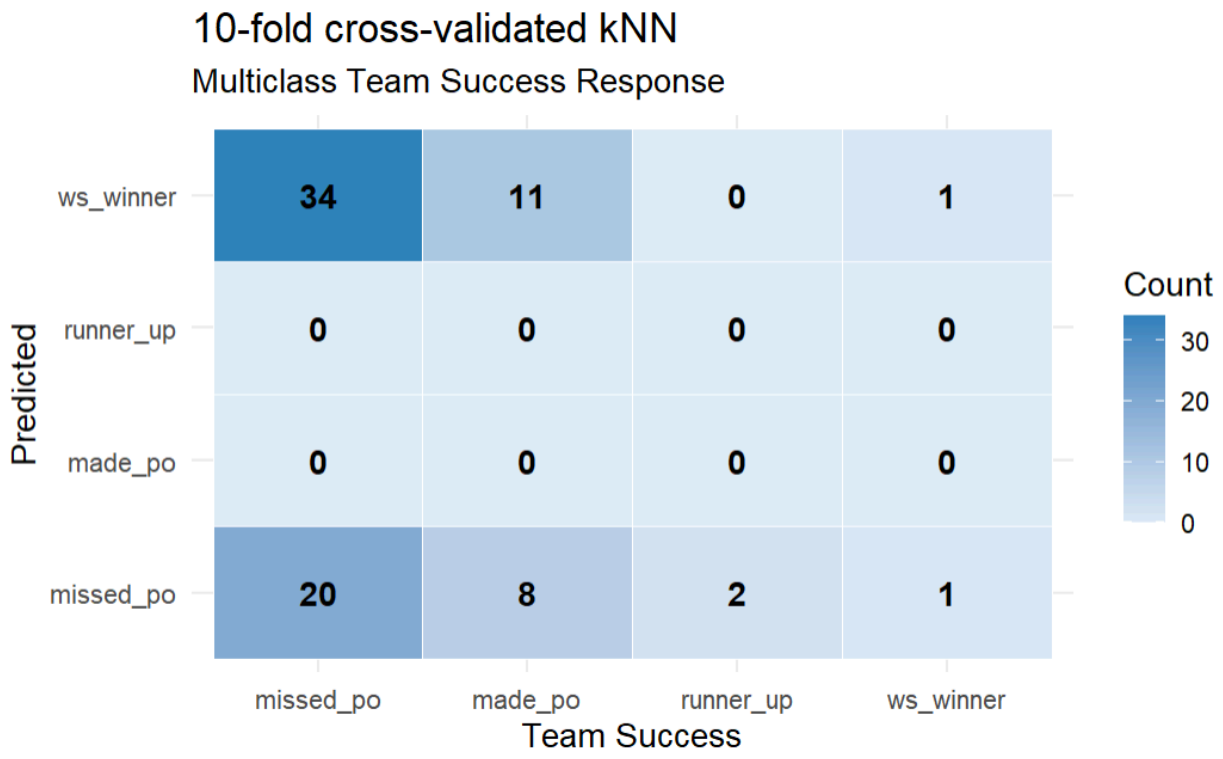

Figure 15A: Confusion matrix for cross-validated kNN model using no dimensionality reduction and multiclass response variable.

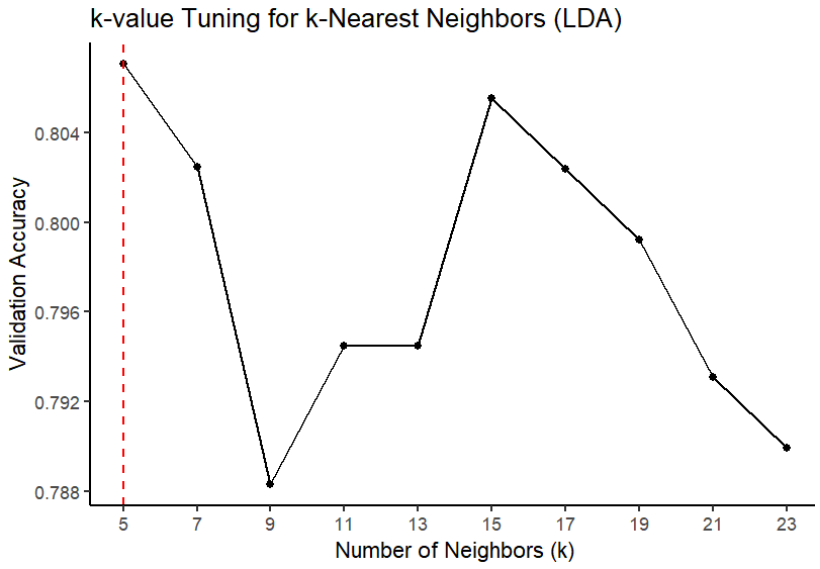

Figure 16A: Validation accuracy of kNN model with no dimensionality reduction and binary response variable over different k-values.

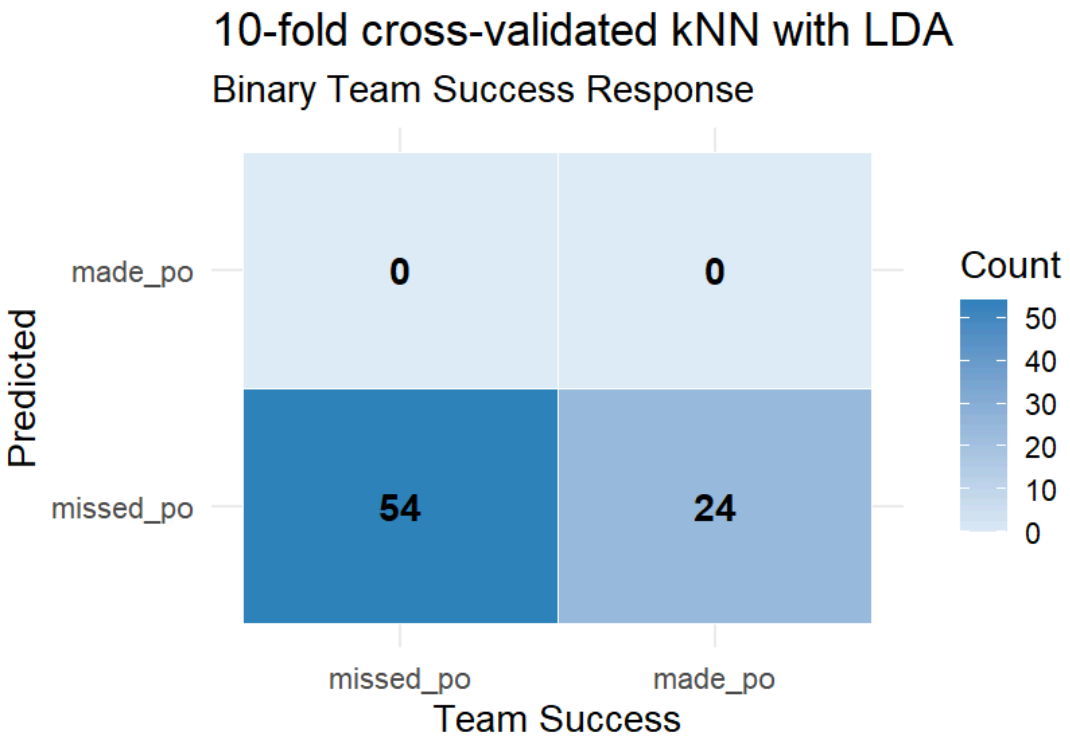

Figure 17A: Confusion matrix for cross-validated kNN model using no dimensionality reduction and binary response variable.

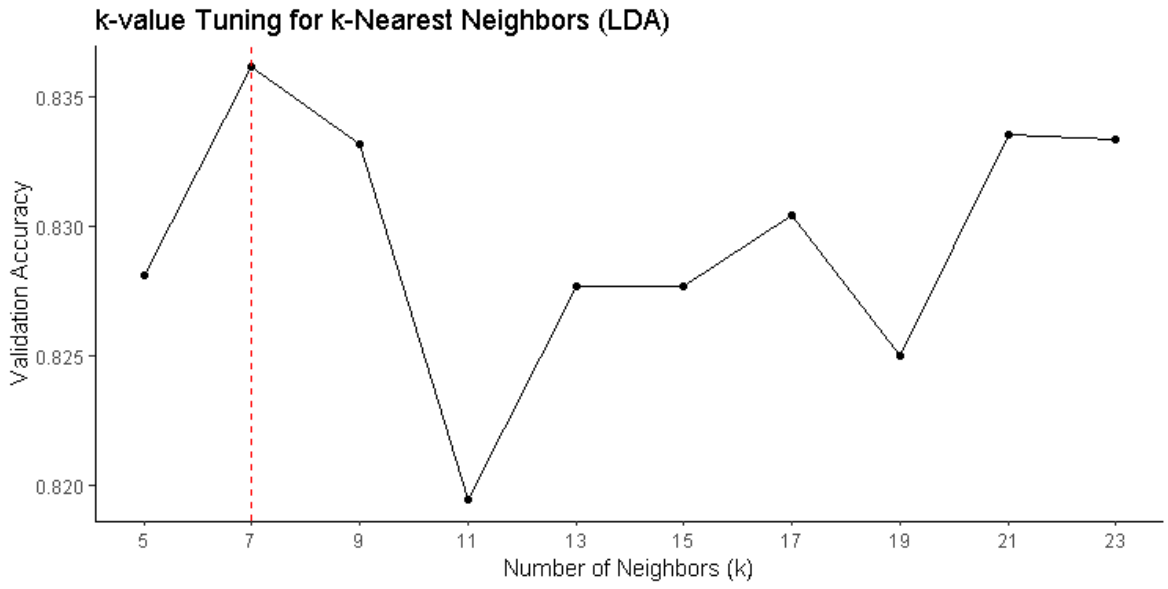

Figure 18A: Validation accuracy of kNN model with LDA predictors and multiclass response variable over different k-values during stratified 10-fold cross-validation.

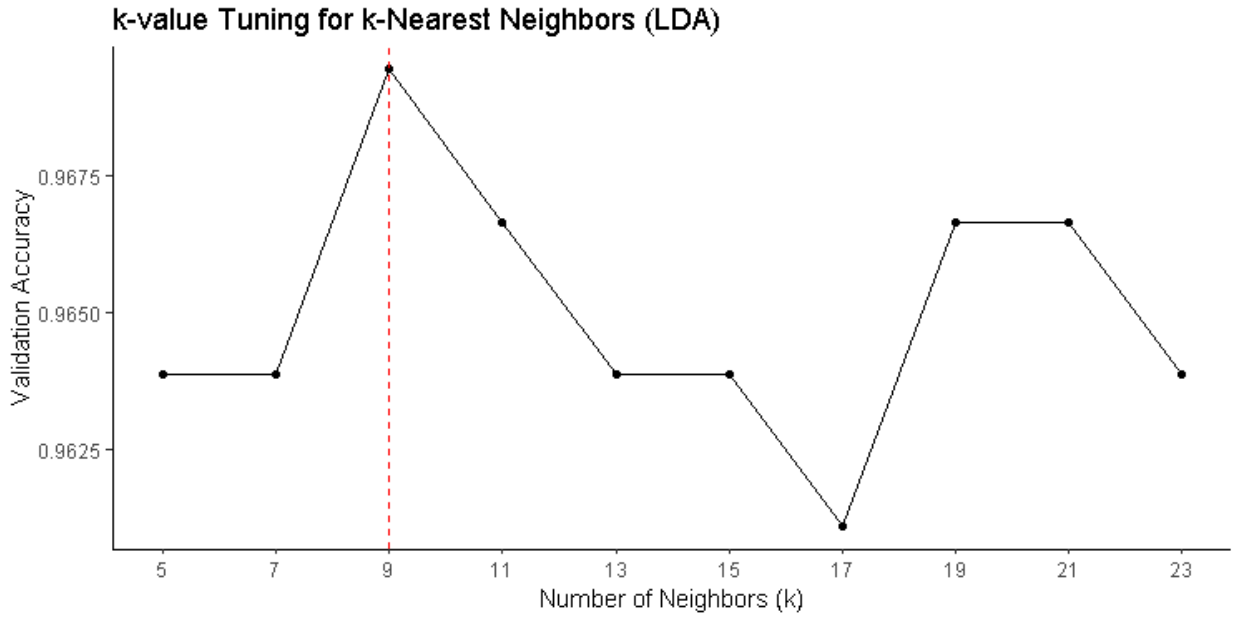

Figure 19A: Validation accuracy of kNN model with LDA predictors and binary class response variable over different k-values during stratified 10-fold cross-validation.

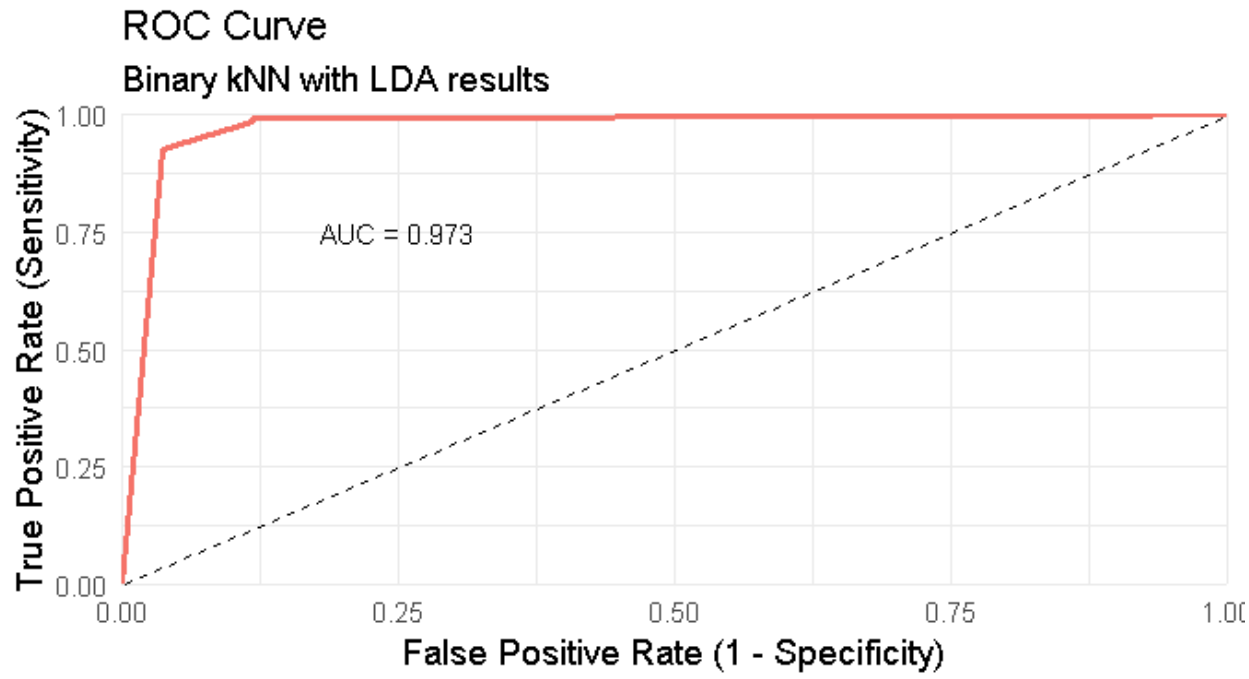

Figure 20A: ROC Curve for cross-validated kNN model using binary response variable.

References

National Baseball Hall of Fame. (n.d.). *Henry Chadwick*. National Baseball Hall of Fame. Retrieved March 2025, 24, from https://baseballhall.org/hall-of-famers/chadwick-henry.

Statista Research Department. (2024, July 3). *Sports betting worldwide - statistics & facts*. Statista. Retrieved March 24, 2025, from https://www.statista.com/topics/1740/sports-betting/#topicOverview

https://onlinegrad.syracuse.edu/blog/sabermetrics-baseball-analytics-the-science-of-winning-accessible/#:~:text=Though%20the%20term%20%E2%80%9Csabermetrics%E2%80%9D%20has,dissect%20the%20science%20of%20winning.