Predicting Playoff Appearance of MLB Teams

By Linus Jen

Introduction

At the start of the summer, a friend asked me if it were possible to create a model to predict how likely a team would win the World Series (for MLB) given the team's spending. While I never did find the answer to this question, I decided instead to broaden the question a bit. In this project, I will be addressing the following questions:

- What models perform the best at predicting which teams make the playoffs (including wild card games)?
- Do models perform better when trained on the raw, per game statistics, or do they perform better with ranked statistics, I.E. where each statistic is turned into a ranking compared with other teams of the same season?
- Is there a subset of data or a certain time frame before a given season, that creates the best models?
- Are there certain variables that are the best indicators to determine if a team will make the playoffs or not? And if so, what statistics?
    - Following this, are there variables that are important for both models using the per game statistics and ranked statistics?

The models tested are: multinomial logistic functions, K Nearest Neighbors, a simple neural network, and Naïve Bayes. These are all supervised classification models, as we are predefining the groups (if a team makes the playoffs or not).

Models will be scored using the following metrics: accuracy, precision, recall, and F1 score.

Unfortunately, variable selection will be based off the variables selected with our multinomial logistic function. This topic was not covered in class, and from online research, there does not seem to be a straightforward way to determine the best variables. In addition to this, the neural network function in R (from nnet) cannot handle the sheer volume of variables present in the dataset. Thus, variable selection will be done in the beginning, and all models will use the same variables.

Data

All data was found and pulled from baseball-reference.com. Variables were taken of both the offensive (running) and defensive (pitching) stats. Because the statistics on baseball-reference.com were the totals for each season, these values were divided by the total number of games played by each team, unless values were already given as per game stats, and saved as our per game stats. Then, rows were grouped by their respective seasons, and each statistic was ranked among all the other teams that season and saved as the ranked dataset. The full CSV file can be found on the GitHub.

Methods

1. Variable Selection

Using both AIC and BIC, the step() function was applied to our full dataset to reduce the AIC and BIC values for the logistic function. Then, the coefficients for the chosen variables were investigated to determine their importance to the model. An ANOVA table was constructed to provide insight on how much variation was captured by these variables, and then VIF was checked for multicollinearity. These steps were applied to both our per game and ranked data, and a subset of each dataset was created and saved as "model_pergame_stats.csv" and "model_ranked_stats.csv", respectively.

2. Model Creation/Training

Models were broken into two parts: using the historical data or using a subset of data.

Using the variables found above, each model was first created using the historical data (I.E. the entire dataset). Then, models were applied to each season's teams and their stats to create the predictions. Instead of the typical rounding to 1 if TRUE and FALSE if 0 to determine if a team made the playoffs or not, an additional step was included. Because the playoffs can only contain a certain number of teams each year, the team's with the highest predictions (closest to 1) were predicted to be the teams that made the playoffs, while all other teams were predicted to not make the playoffs. These predictions were then saved, to be compared later.

To train models using only a subset of data, a for loop was first implemented to represent the number of years we want to train our model with. The goal of using a subset of the data is to see if there are any trends over the past few seasons that may improve the model's accuracy. Thus, first we check if there are valid years prior to the year we want to predict for to create our model and if not, the next iteration is run. Otherwise, for each year, we create a model using the predetermined number of seasons prior to that given year and use this year as our test data. Once each year has been predicted for, we save these predictions to be scored later.

3. Model Scoring

All models, except for the historical K Nearest Neighbors model, are scored based on accuracy (how many predictions were correct), precision (how many of the TRUE predictions were actually TRUE), recall (how many of the TRUE values were actually predicted by the model), and F1 score (weighted average between precision and recall). A confusion matrix between the predicted and actual values was created, then the formulas were applied to find the metrics above.

4. Year Comparisons

Graphs were made for each type of model (historical vs. subset, and per game vs. ranked) that compared the metrics with the number of years used in the training data. Comments were made about the shape, and changes in different scoring metrics.

Results and Discussion

1. Model Selection – Per Game Data

Final Variable Selection

Per Game Data (11 variables): R/G (runs per game), SO/W, number of batters per game, own BB (bases on balls), own HBP (hits by pitch), SH (sacrifice hits/bunts), SF (sacrifice flies), own LOB (runners left on base), IP (innings pitched), runs allowed, and WHIP.

Ranked Data (19 variables): R/G (runs per game), SLG, HR9 (average home runs per game), SO9 (average shutouts per game), SO/W, AB (at bats), B2 (number of doubles), B3 (number of triples), own SO (strikeouts), SH (sacrifice hits), own IBB (intentional bases on balls), own LOB (left on base), number of pitchers, RA/G (runs allowed per game), cSho (shutouts), saves, HBP allowed (times hit by a pitch), BF (batters faced), and ERA+.

1.1 Per Game Data

Using AIC, 23 variables chosen for the multinomial logistic function, and they were:

R.G, OPS, H9, HR9, BB9, SO/W, number of batters, AB, own H, B2, B3, own home runs, RBI, own BB, own HBP, SH, SF, own LOB, RA/G, CG, IP, runs allowed, and WHIP.

The AIC value is 536.65 in this final model. However, upon deeper analysis, this model is not valid. From the ANOVA plot, we see that most variables do not capture a significant amount of variance. In addition to this, we have high collinearity between many of the variables, leading to VIF values above 1000. Thus, this model was ignored.

Under BIC, the final model results in 13 variables and were the following:

R.G, SO/W, number of batters, AB, own BB, own HBP, SH, SF, own LOB, RA.G, IP, runs allowed, and WHIP.

The BIC score here is 547.23. It is important to note that the variables chosen in the BIC model is a subset of the AIC model. Upon deeper analysis, while all the coefficients are significant, several variables do not capture a significant amount of variance. In addition to this, several variables have issues with collinearity.

For the final model, two variables were dropped, to have the following 11 variables:

R.G, SO/W, number of batters, own BB, own HBP, SH, SF, own LOB, IP, runs allowed, and WHIP.

The final AIC score was 692.55. When referring to the coefficients, many of the variables lose their significance. However, all but two of the variables (own HBP and own LOB) are significant. This model no longer has issues with multicollinearity and was chosen as the final dataset used for all future models.

1.2 Ranked Data

Using AIC, the following 19 variables were chosen:

R.G, SLG, HR9, SO9, SO/W, AB, B2, B3, own SO, SH, own IBB, own LOB, number of pitchers, RA.G, cSho, saves, HBP allowed, BF, and ERA+.

The AIC score was 628.82. When looking at the coefficients, roughly half of the variables were not significant. Following this, several variables were not significant in the ANOVA table. However, unlike the per game models, models made with ranked data had no issues with collinearity.

Using BIC, the following 7 variables were chosen:

R.G, HR9, RA.G, saves, own BB, CG, and SLG.

The BIC model has a score of 6.19.81. Looking deeper, this model seemed valid without any issues. All the coefficients were significant, and the ANOVA table showed that every variable captured a significant portion of the variance. Lastly, there was no issues with collinearity in this model. This subset of data will only be used for the logistic function.

Due to the lack of multicollinearity, the variables found under AIC will be used for all models except the logistic function, where the BIC variables will be used.

2. Model Scoring

Models will be compared via their accuracies, precisions, recall, and F1 scores. For each classification type, several comparisons will be made.

- Do models using the historical data perform better than models trained by certain years?
- Does the number of years used to train the models influence how well the model performs?
    - Is there any specific year(s) of data that produce the best predictions for all models?

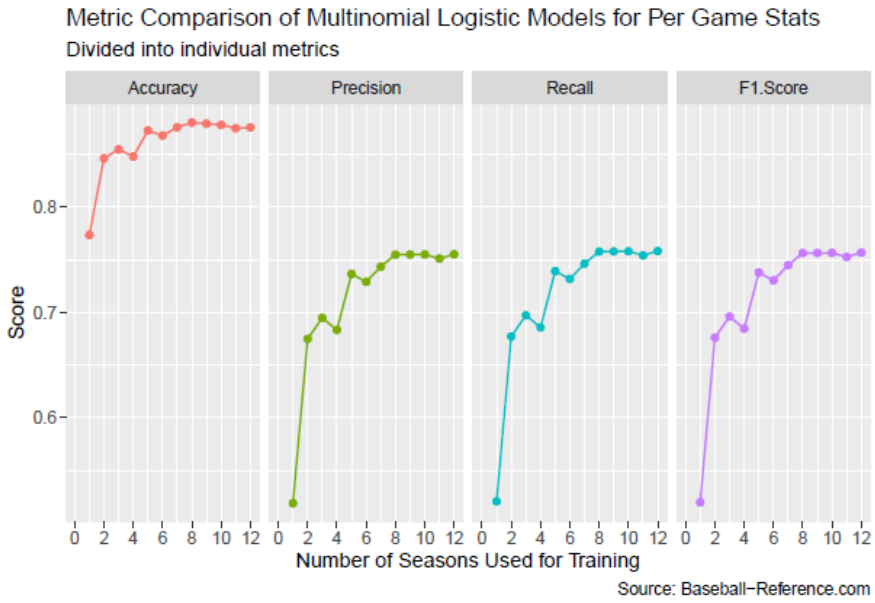
The following sections elaborate in detail at how each model performed, but here I will focus on the best models. Historical data trained the best models, leading to better playoff predictions over using a subset of years Neural networks performed the best overall, garnering the highest scores in accuracy, precision, recall, and F1 scores. The neural network using the ranked stats performed better than the model using the ranked stats. In general, ranked data seemed to better predict what teams make the playoffs, compared to models trained with the per game statistics. Ranked statistics showed less levels of collinearity, while less variables were needed to create the same, if not better results (as evidenced by the logistic function). However, for other models, the greater accuracy may be due to the larger number of variables available in the ranked data used to train the models.

2.1 Logistic Functions

2.1.1 Per Game Data

Historical – In the final logistic model using the historical data, the model had a .884 accuracy, .751 for precision, .753 for recall, and .752 for the F1 score. While this does seem high, the BIC model, which included only 2 more variables, had an accuracy of .912, .811 precision, .814 recall, and .813 F1 score – around 3-5% better for each metric.

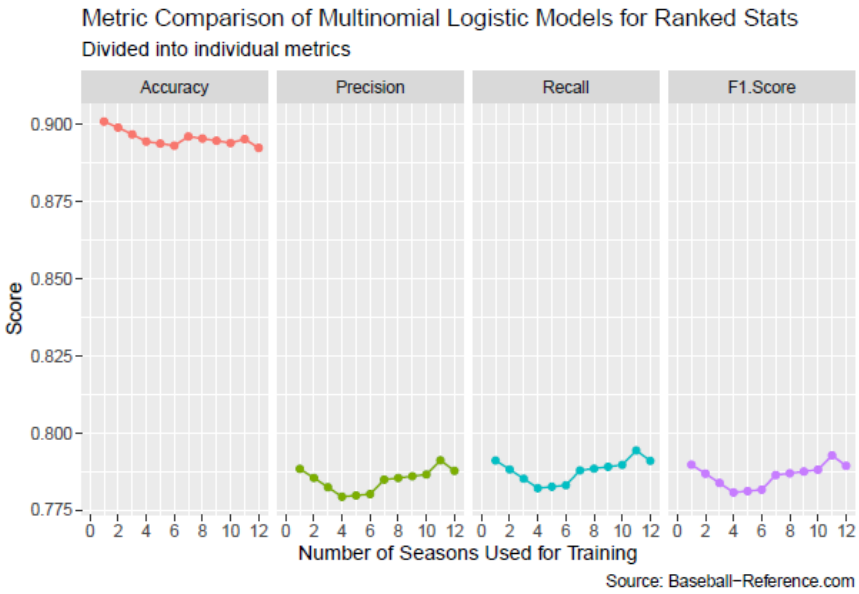
Subset of years – Using a certain number of years before each season to predict what teams make the playoffs each year, in the graph below, we can clearly see that the more seasons used to train our model, the better the model performs. The logistic function reaches its best prediction capabilities with an accuracy around .875, and precision, recall, and F1 scores of around .75. These values are very close to how our model using the historical data performs. While these similarities do not add much to conclude which model is better, we can note that around 8 years or more of data prior to each season to train each model results in roughly the same prediction scores.



2.1.2 Ranked Data

Historical – In the final logistic model using the historical ranked data, the model had a 0.901 accuracy, 0.79 precision, recall, and F1 score. This model only did slightly worse than the full model (off by ~0.01-0.03). However, the final model only had 7 variables without any collinearity.

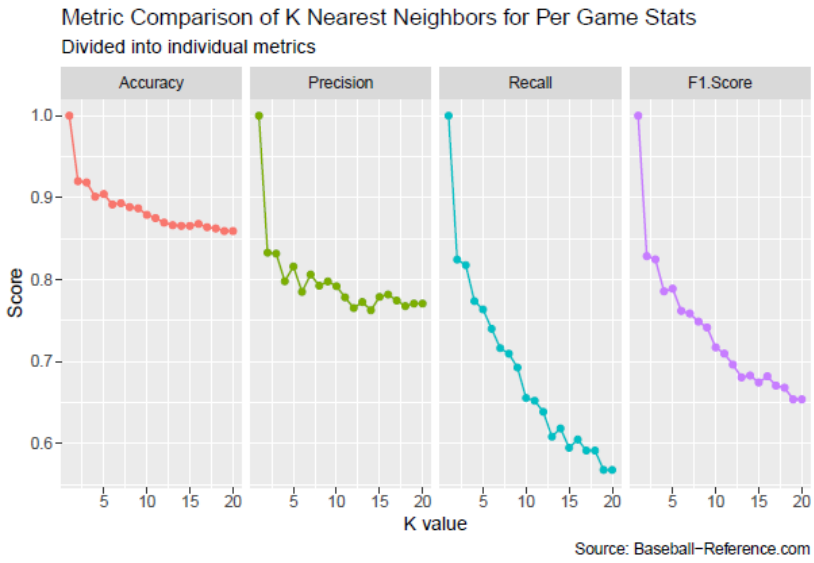
Subset of years – Using a certain number of years before each season to predict the following season's playoff teams, the metrics scored similarly with the historical data. Accuracy hovered around 0.89, and precision, recall, and F1 score were all around .875. While there were slight differences in metrics, there was not a specific number of seasons that led to marginally better metrics, as you can see below.



2.2 K-Nearest Neighbors

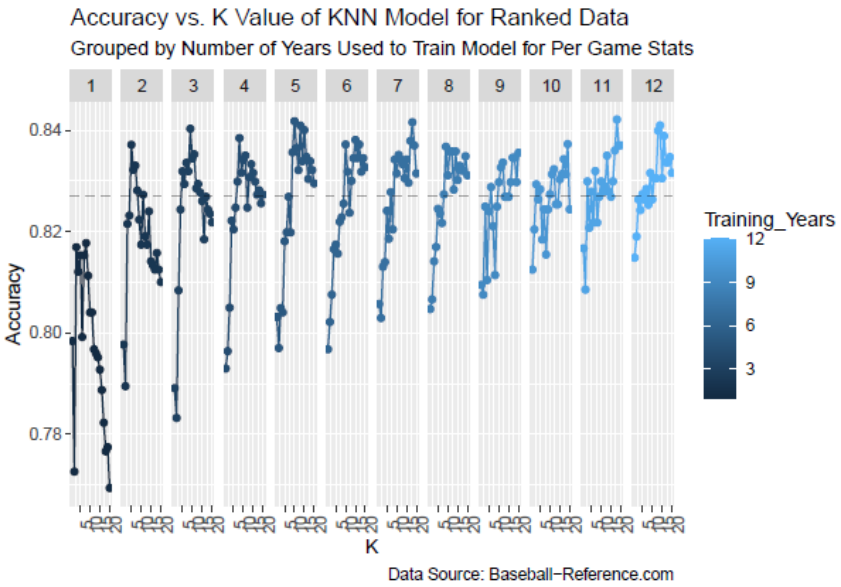
2.2.1 Per Game Data

Historical – In the final KNN model using the historical data, K values were compared to see which K value gave the best predictions. Because we did not have training data, the model was tested against the true predictions used to train the data, and as such, a K value of 1 must be ignored. In general, however, we see that as the K value increases, the overall metrics tend to decrease, especially with recall, dropping to scores below 0.6 when K is greater than 15. Using a K value under 5 seems to be the best bet to create the best KNN model. Overall, accuracy seems to hover around .88, precision around 0.8, and recall and F1 scores never stabilizing.



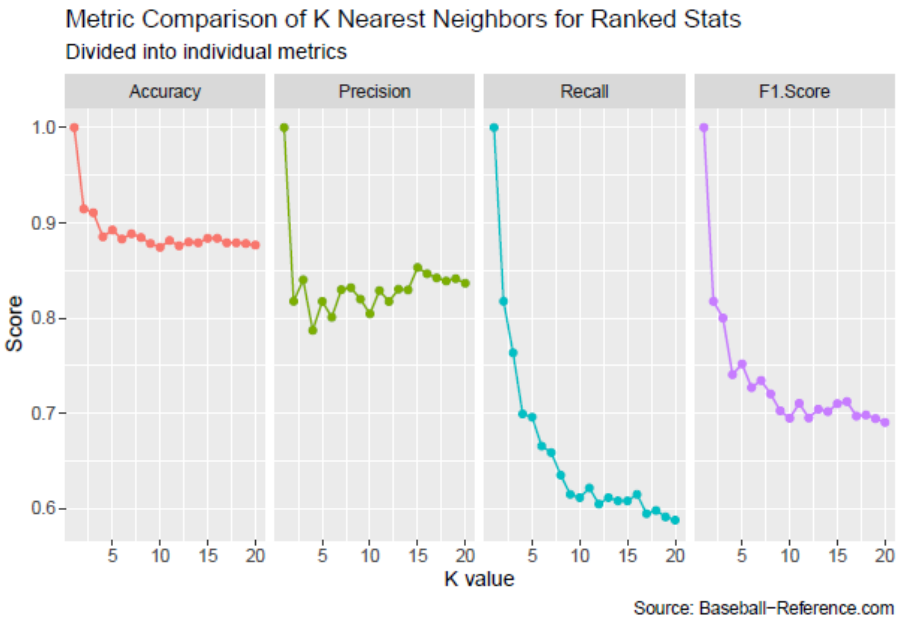
Subset of years – Because of the need to tune the K parameter, only accuracy was checked in the models. Overall, it seemed that larger K values and a middling number of years (between 8 and

15 years) used for training led to the highest accuracies. In general, the best number of seasons to use as training data for KNN models is 8 seasons, and the K value with the best accuracy overall was a K value of 11. However, the best combination of the two occurred when K = 11 and training years = 11.

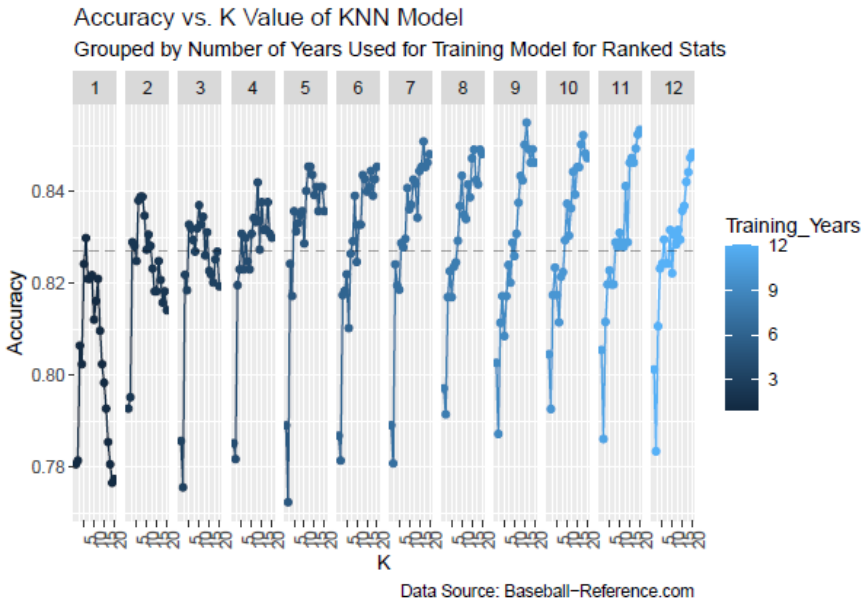


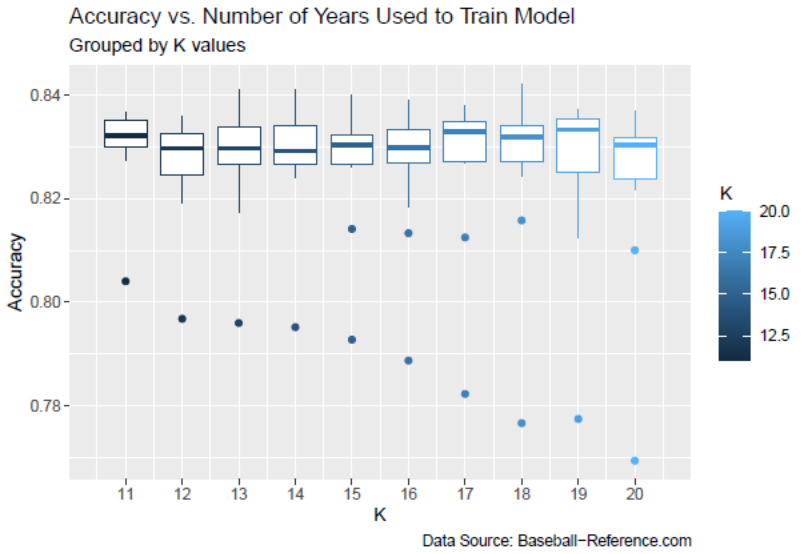
2.2.2 Ranked Data

Historical – Similar to the per game models, as the K value increases, the models tend to do worse, but with ranked data, the metrics tend to stabilize a bit. Accuracy eventually reaches around 0.88, precision around 0.85, and recall and F1 score both continually decreasing, as shown below. Again, we ignore when K = 1, as we used our training data as the testing data.



Subset of years – Again, only accuracy was scored due to tuning of the K value. From the graph below, it seems that as the number of years used to train the model increases and the larger the K value used, the better the model performs. The best performing model occurred when K = 16 and 9 years of training data was used. However, once K is greater than 10, the accuracy in general tends to stabilize around .83.





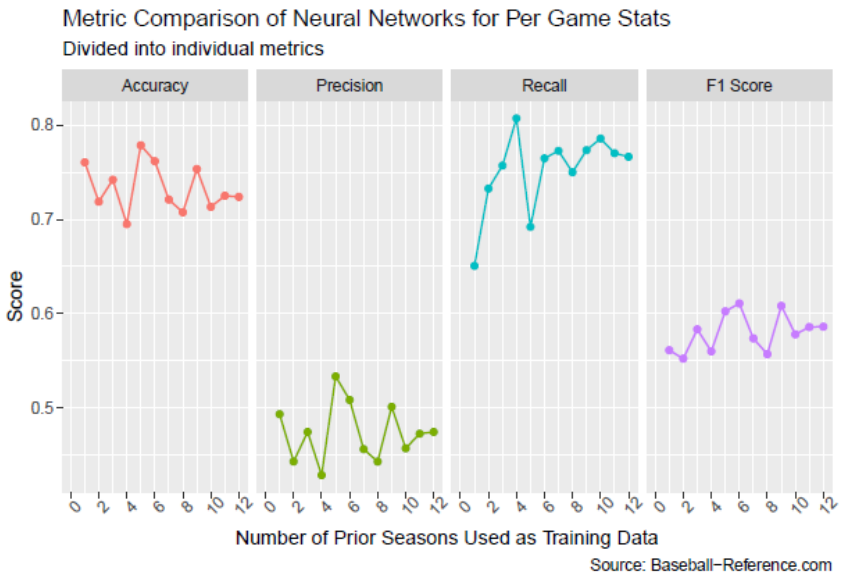
2.3 Artificial Neural Networks

2.3.1 Per Game Data

Historical – The final neural network model does the best job at predicting what teams make the playoffs. Using the historical, per game data, the model has an accuracy of 0.934, a precision of 0.859, recall of 0.861, and F1 score of 0.86. All these metrics score the best out of all the models

chosen. It should be noted that neural networks are far more complex than all the other models used.

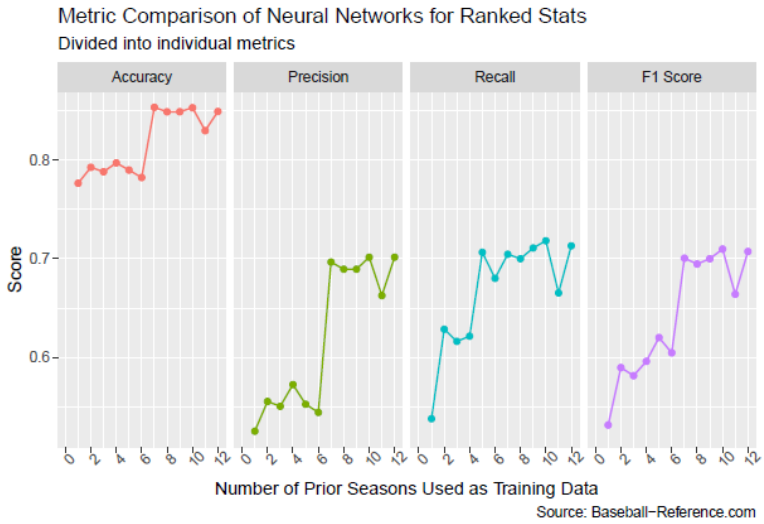
Subset of Years – Referring to the graph below, there does not seem to be a subset of years to be used to train the model that creates the best model. In general, the accuracy hovers around .74, precision around .48, recall around .77, and F1 score around .58. These values are far lower than when using the historical data, supporting the notion that the historical data does the best job at predicting playoff appearances.



2.3.2 Ranked Data

Historical – Using the historical, ranked data, the neural network performs the best across the board. The accuracy comes out to be 0.958, and precision, recall, and F1 score all around 0.91. This finding supports that neural networks perform the best. However, ranked data performing better than the actual data itself is noteworthy, but this may be the result of having more variables present to train on.

Subset of Years – Limiting the number of years to train the neural network again seemed to hinder its performance and using more than 6 years of prior data yields the best results. The accuracy stabilizes around .85, and precision, recall, and F1score hover around .7.



2.4 Naïve Bayes

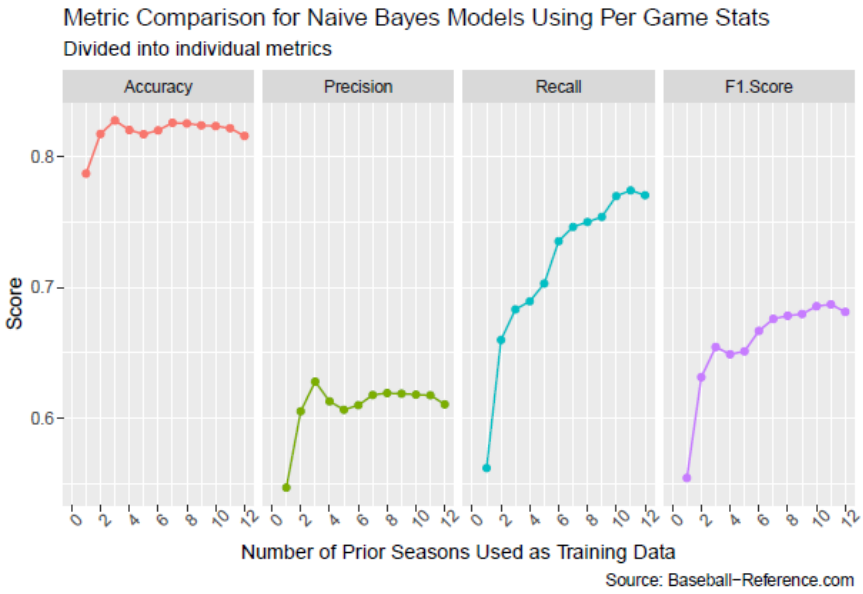
2.4.1 Per Game Data

Historical – Naïve Bayes tended to perform the worst among all the models. Using the historical, per game data, the accuracy came out to be 0.836, precision of 0.643, recall of 0.676, and an F1 score of 0.659. Unfortunately, given how the function works, I was unable to run the function that limits the number of teams that make the playoffs each season. However, we can clearly see that the Naïve Bayes model misclassified many teams incorrectly, with the confusion matrix as follows:

nb_predictions_pg	FALSE	TRUE
FALSE	857	96
TRUE	111	200

From the table to the left, there are 111 false positives from this model, as well has 96 false negatives. This was by far the worst performing model of all the models done thus far, and this is exemplified in the low metric scores.

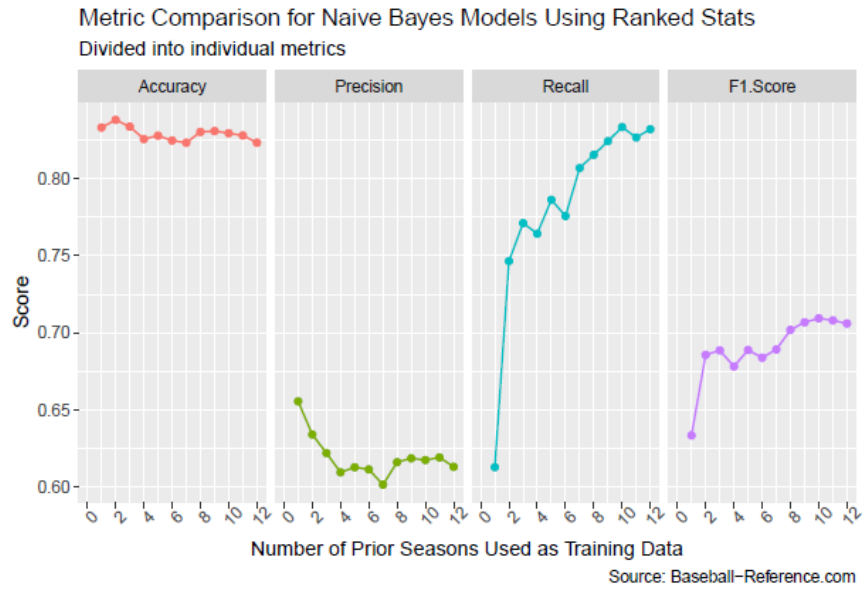
Subset of Years – This model tends to perform better than the model using the historical, per game dataset. The accuracy stabilizes to be around 0.815, precision around .62, recall around .75, and an F1 score around .68. Naïve Bayes seems to improve with the more years used in its training data, as the recall, as shown below, greatly increases as the training set increases. Surprisingly, using a subset of years to train the model is better than using the historical, per game data.



2.4.2 Ranked Data

Historical – Using the historical, ranked data, the model performs similarly to the historical, per game model found earlier. The accuracy comes out to be 0.841, precision of 0.621, a much higher recall of .834, and F1 score of .712. Following an earlier comment, Naïve Bayes seems to perform the worst out of the 4 models to compare with.

Subset of Years – Using only a certain number of years to predict playoff appearances results in similar results than the historical data. As shown in the table below, accuracy hovers around .825, but precision decreases initially to stabilize around .61. Recall consistently increases to be around .825, pulling the F1 score to hover around .70. These values are slightly higher than the model using a subset of years and per game data, but the trends are similar between the two.



Limitations

Variable selection – due to the lack of coverage, from classes and online, regarding how to properly select variables for all models except logistic functions, I was forced to use a subset of the data for all models, using the "best" subset of variables found in the logistic function. Because of this, not only were the number of variables used between per game and ranked data different, but models between these two types are no longer comparable. In addition to this, other methods of blending data together or keeping data integrity, such as principal component analysis, would be of great use here, especially with the logistic function. The step from the BIC subset to the final subset changed the metrics by over 3% from dropping two variables. If these variables could be combined with others, the model could have done a better job (though at the loss of interpretability).

Using premade functions for the models – Following variable selection, because each model was not built from the ground up (EX: did not back propagate the neural network to optimize each weight), I was limited to how much I could do with each model. For example, with variable selection of neural networks, variables with low weights could be removed from the dataset. However, there was no way to easily check each weight to compare which variables should or should not be included, nor could the function handle using the entire dataset, and thus I was forced to use a subset of the data available.

Lack of per game statistics at each game – because the data used was originally from each season's totals, it cannot be definitively stated that a team would average a certain number throughout the entire season. While I converted each statistic to per game stats, it is likely that a team's average for each statistic changes over time. Theoretically, these models should still provide quality insight, as the law of large numbers dictate that teams will reach their averages. However, teams going through slumps, or suddenly catching fire, may see their playoff likelihoods change drastically.