Shawn Fang

Final Report: NBA Game Prediction Using Machine Learning

Problem Statement

A long time fan of basketball and the National Basketball Association, the statistics behind the game has always been a fascination to me. With the opportunity to choose a guided capstone project, naturally an analysis of NBA statistics was a top choice. In this project, we put in practice machine learning theories towards the analysis of NBA regular season statistics and built a model to predict wins and losses, based on game statistics.

From research, there have been several attempts to model and predict game outcomes. Our goal was to match or better the upper bound of currently achievable prediction accuracies ranging between 66-72% for regular season games. Data collected over the period of 2008 to 2020 were used to train our model. Outside of cleaning and wrangling data, in addition, several features were created and appended to the datasets. Building and tuning several models, our tuned logistic regression was able to achieve a precision of 0.64. Furthermore, the model can be used to predict additional matches in the future, or conduct head-to-head evaluation of prior era teams.

Data Wrangling

The data was acquired from various data providers on kaggle and came in the form of several seasonal sets of player statistics and team statistics. After combining the relevant seasons, our initial data included a team stat (15348, 124), and player stats (37369, 26). Initial reviews showed several features that were of no use such as 'Gamelink', % columns, and NaN rows where players who were on the bench and played zero minutes in games were collected. These were dropped to form clean stats dataframes for both player and teams.

During EDA, it was identified that teams who have changed their names or cities within the 2008 - 2020 time period, were creating non-uniform data in several features, especially the GameID feature created to join the player and team datasets. Some modernizing/changing of prior team abbreviations helped solve the issue.

We ended data wrangling with cleaned player stats (291008, 24), and cleaned team stats (14348, 80) to go into feature engineering.

Feature Engineering

Having watched a fair share of NBA games in conjunction with managing years of fantasy basketball teams, it is understood that a player's or team's individual game statistics are constantly fluctuating. Using those individual statistics to train a model wouldn't be as effective. Therefore, features that will help explain the team and player progress throughout the season were created to help our analysis.

1) Elo Rating

As the NBA modernized, advanced statistics has become a large facet of the game. There are established frameworks that statisticians have built which can be applied in our machine learning models. One such concept was the Elo system created by Arpad Elo, which was first invented as an improved chess-rating system, and is now used in various competitions. The Elo system is a calculated stat that keeps track of relative skills and qualities of teams within a league. Every team supposedly starts with the same Elo rating at the beginning of a season, and will either add to or subtract value from their opponents as they win/lose throughout the season.

2) Team Recent Performance

Creating new values for each sample, I took the average of each teams performance stats from the previous ten game windows. This gave us an average performance value for each statistic leading up to game night.

3) Player Recent Performance

Similarly, I took the average of each player's performance stats leading up to game night from their previous 10 game window.

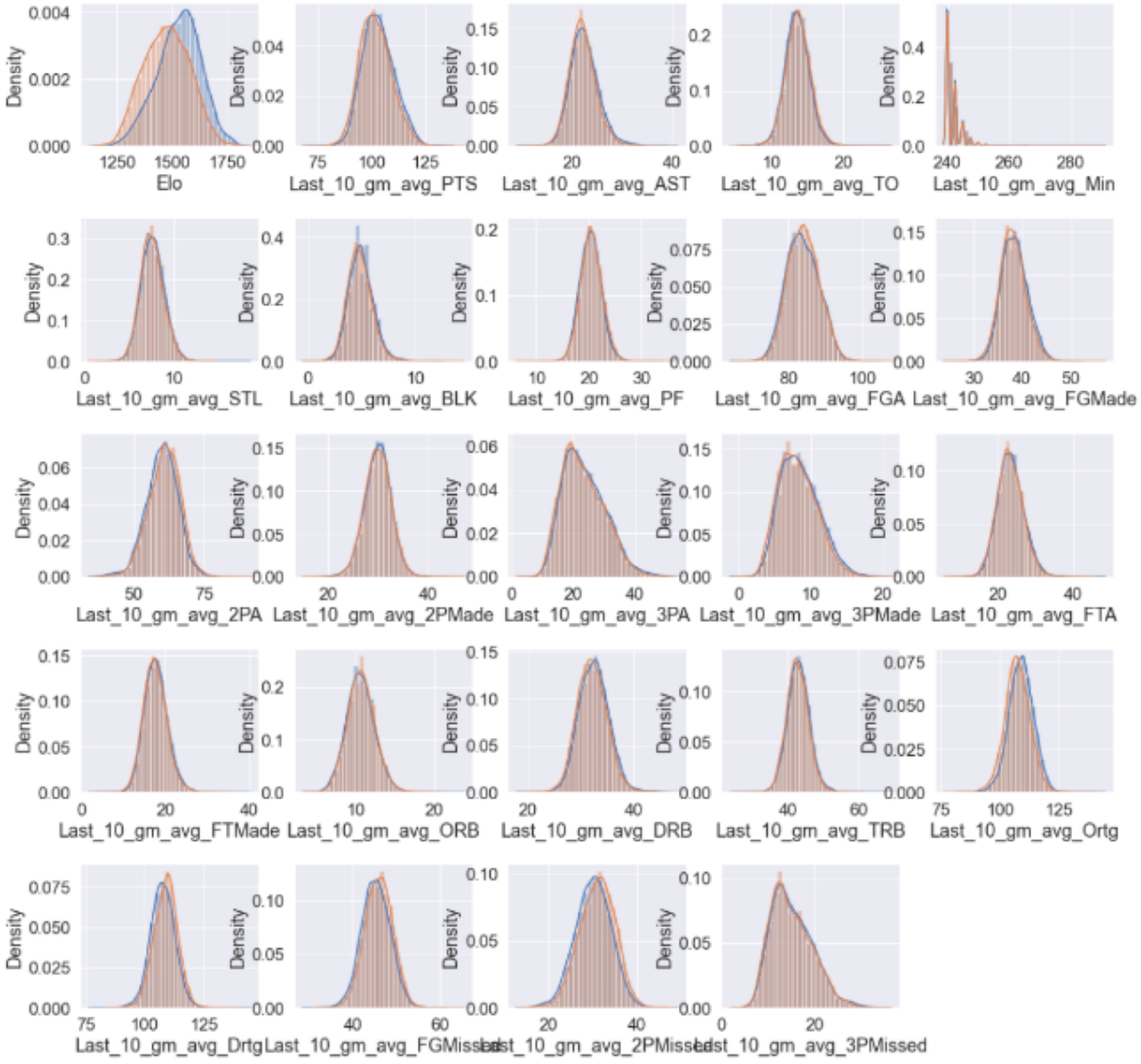
4) Player Season Average Performance

A player's seasonal average is a baseline for comparison. As players play, their performances do shift, but most of the time they will revert to their average, without taking into account career trajectory.

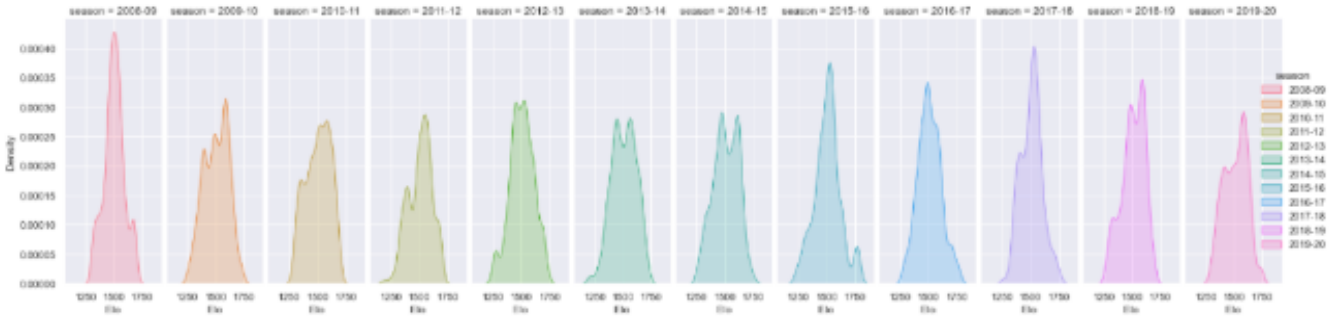
Exploratory Data Analysis

With the features built, the Elo was the feature I wanted to explore the most in relation to the rest of the statistics. When comparing the performance features between winning teams and losing teams, it's clear that the ELO has a strong correlation for differentiating the two. Other stats that showed good differentiation is Offensive rating

(scoring per 100 possession), and defensive rating (Stls*Blocks+Opponent Possession Differential). These features summarize the teams offensive and defensive efficiencies, and makes sense that they show some correlation to differentiating winners and losers. This led to the hypothesis that summarizing features better different the two.



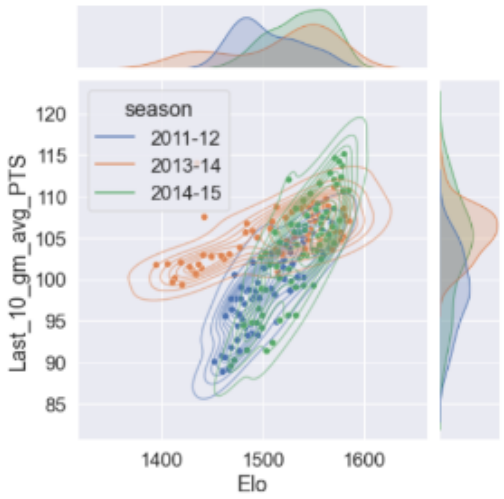
Looking deeper into the Elo rating, we checked to see it's distribution throughout the season.



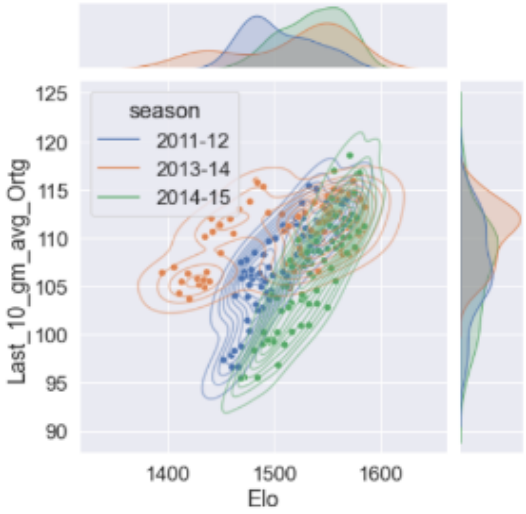
Interestingly enough the distribution tells the migration of strength in the league. The closer we get to the modern era, the more condense are teams with high Elo's.

When reviewing Elo in relation to other stats, we were able to see a clear correlation. Below, we plotted data from the Phoenix Suns 2011 - 2015 stats.

Elo vs. Avg Score



Elo vs. Avg Off Rating

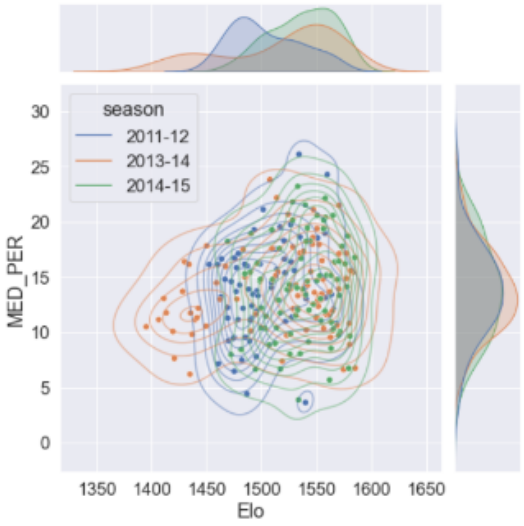


Building on our hypothesis that efficiency ratings can be helpful or provide more insightful relationships in our data, the PER is an advanced statistic that is utilized widely in the modern NBA analysis. PER essentially rewards positive stats such as scoring, rebounding and penalizes negative stats such TOs, fouls, and missed shots. This stat would allow us to sum up each player's performance and test whether the players performance efficiency helps with winning leading to high team ELO.

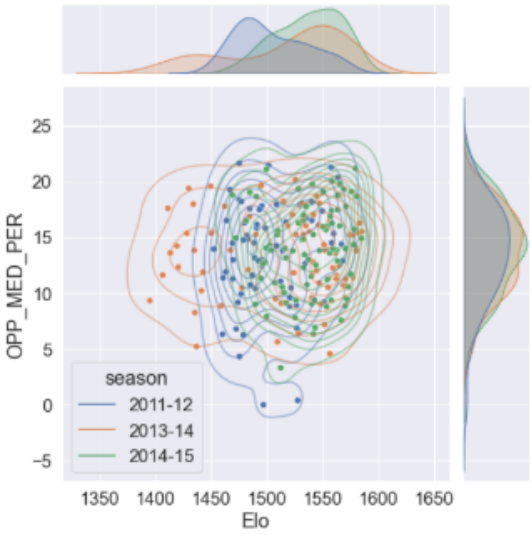
Sticking with Phoenix, we compare their Elo against the team's average PER (averaging the PER of all players on a team). While building this hypothesis feature, we noted that

due to the way the PER is calculated, players that sometimes do not play many minutes and only racked up negative values, their PER's are incredibly skewered in the negative direction. Because the impact from a minutes standpoint on court is quite negligible, it would more accurately reflect the teams average performance efficiency using the median team PER.

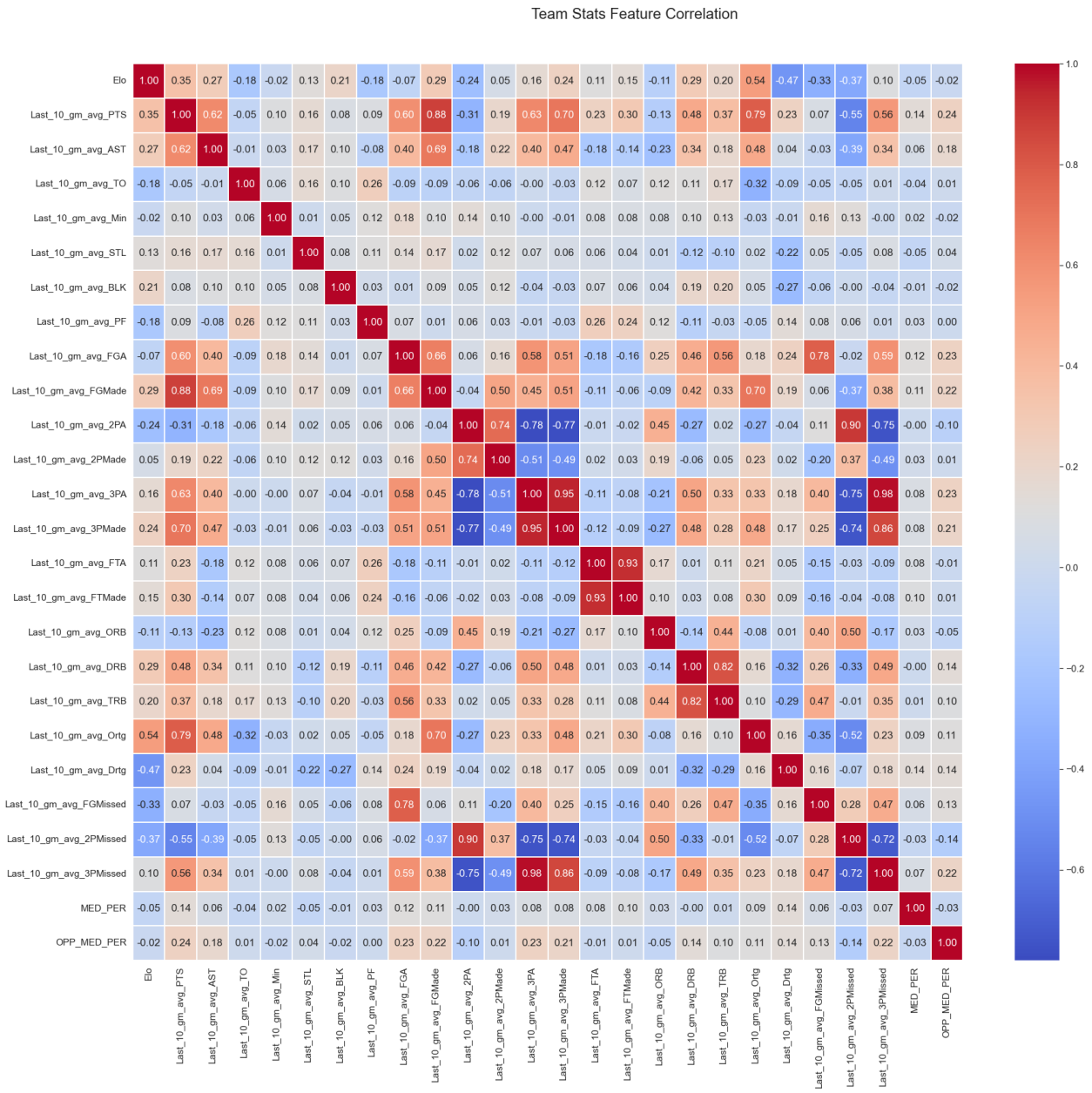
Elo vs. Med Team PER



Elo vs. Med Opponent PER

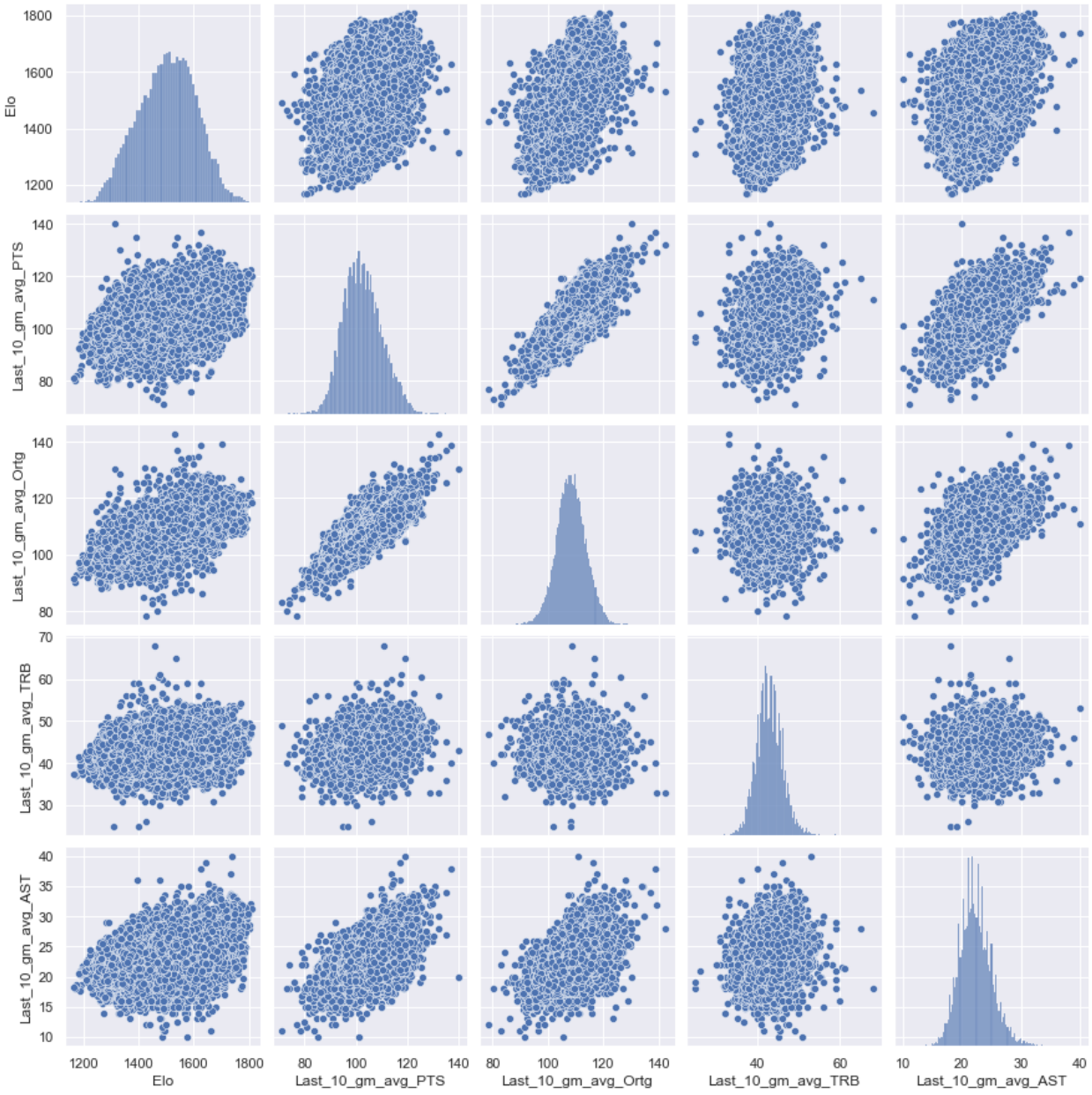


This data indicated that the PER didn’t show a strong correlation with Elo, and when we looked at the features heatmap, we saw that the Elo did not correlate well with PER.



From the heat map, we could also see several categories that did correlate well with Elo. Offensive and Defensive ratings as we saw in the winners, losers feature comparison kde graphs were highly correlated (positively and negatively) to ELO. In addition, positive correlations were found in features such as FGMade (field goals made), TRB (Total Rebounds), AST (assists), PTS (points).

Focusing on these features, we made a pairplot to see their relationships specifically.

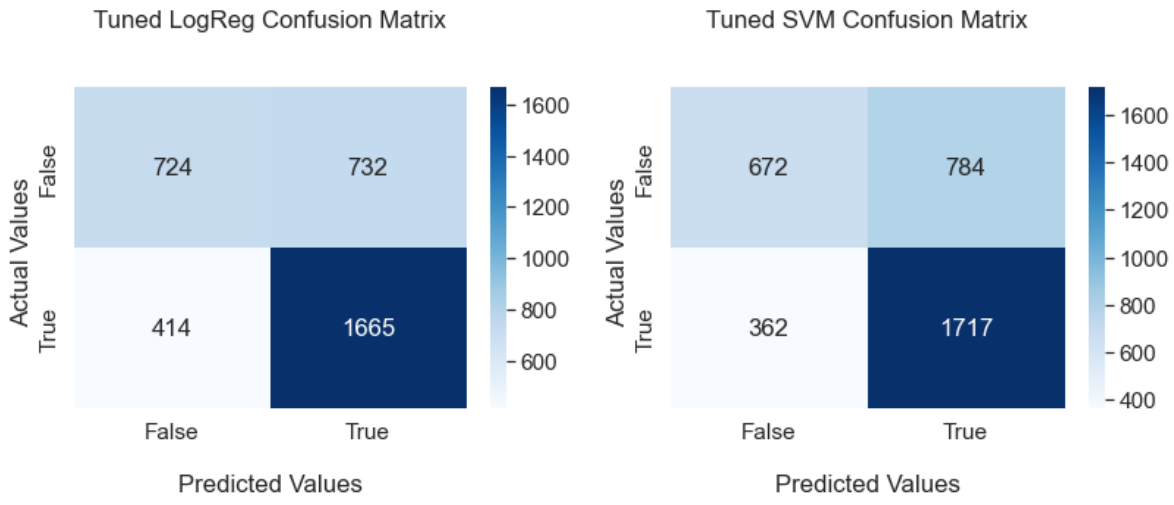


Model Selection of the Final Team Stats

In this section, the Final Team Stats data frame was split for training and testing. A variety of models were selected; logistic regression, K-Nearest Neighbor, Support Vector Machines, RandomForrest, and Gradient Boost. Each of these were tuned via RandomizedsearchCV or GridsearchCV and their classification reports were compared.

Through our testing, we found that the Support Vector Machine and Logistic Regression both have shown to produce very competitive results. Ultimately this could be narrowed down via multiple sampling and linear regression. Nevertheless, due to time constraint,

the tuned Logistic Regression model was chosen for moving forward. In our Logistic Regression we consistently got a Precision score of 0.64, and F1 score of 0.56. In comparison, SVM provided Precision scores up to 0.65 but an F1 score of 0.54, although the SVM model Precision score is not consistently higher than the Logistic Regression.



Using Model for Inferencing

To study how our model works, I took a few samples from late 2008, created an average of those game statistics and compared them to the averages of samples from late 2020, as a way to see if modern play styles would beat out on older NBA play styles. Recalling our Logistic Regression model with parameters below:

LogisticRegression(penalty='l1', random_state=42, solver='liblinear')

We tested, and were able to get a prediction indicating that the Home team in our sample (early 2008 game averages) were able to defeat the Away team (late 2020 games).

```
In [93]: early = early[h_col]
         late = late[a_col]

         x_new = early.append(late)
         x_new = x_new[col]
         loaded_model.predict(np.array(x_new).reshape(1,48))

         /Users/Shawn/Documents/UNH_ML_SLF/MLassignments/UNH-Repo/venv/lib/python3.9/site-packages/sklearn/base.py:445: UserWa
         rning: X does not have valid feature names, but LogisticRegression was fitted with feature names
           warnings.warn(

Out[93]: array([1])
```

We can test further pulling sets of games from both eras to see what win percentages each era gets. In addition, we can swap home teams and away teams to see how the match ups end up. Other potential uses for our model, such as predicting future games by placing their performance data leading up to game night as test sample for the model.

Conclusion and Future Considerations

In our investigation, we explored several different models to classify our NBA data. Acquiring data from kaggle, we wrangled, explored, engineered features, and finally selected a model that had the highest predictive precision.

The most consistently accurate model turns out to be our tuned logistic regression model but not by much from our other models such as Support Vectore Machines, GradientBoosting, RandomForest, and KNN. Our top accuracy ranged in the 64%, and falls short of the 66-72% range.

If we had additional time, we would further acquire additional defensive data for feature engineering. It is hypothesized that better features can be produced to improve the classification. Future features to explore would be to acquire player's defensive ratings, in which older NBA stats did not acquire until recent updates to the analysis principles for the game of basketball. In addition, utilizing the current data, a feature that could compare a players performance trend (compare to their seasonal avg) could help predict better player performance for predicting game results. In the same scope, we can review a players career trajectory up (age/performance base) as a player grows to their prime and the reversal of that trajectory while the player age passes their prime in order to assess a more accurate seasonal average.