Predicting NCAA Men's Basketball Win Percentages Post-COVID: A Comparison of Regularization in Linear Regression and Neural Networks

Elena Muyo de Bonrostro, Colby Eagan, Julia Katsoulis, Charles Moseley, Joseph Sachtleben
STOR 565 - Group 4
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill

1 Introduction

Predicting the win percentages of NCAA men's basketball teams can be challenging due to the many factors that influence team performance. Logistic regression and other traditional statistical models have been commonly used in sports analytics. However, as machine learning techniques evolve, there now exists the opportunity to explore complex interactions and nonlinear relationships within sports data. This project hopes to aid coaches in focusing on practicing more valuable skills, as well as recruiters in scouting players exhibiting more statistically important skill sets.

This study examines the effectiveness of various regularization methods in predicting NCAA basketball win percentages in the seasons following the COVID-19 pandemic; a new college basketball landscape with the transfer portal, NIL, and a fifth year of player eligibility. Specifically, we compare L1 (Lasso) and L2 (Ridge) regularization within linear regression models and dropout regularization in neural networks. Additionally, the linear regression models will be evaluated with hyperparameter tuning using cross-validation. These models can be generalized to future years, which is significant in post-COVID seasons where historical trends may no longer hold value and overfitting is a concern.

To evaluate model performance, we will use Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared alongside diagnostic plots. By comparing these regularization methods, we aim to identify which approach best balances predictive performance with the ability to generalize to unseen data. Ultimately, we hope this study contributes to the growing field of sports analytics and machine learning, providing valuable insights for predicting team win percentages.

2 Related Works

There have been many studies analyzing college basketball performance with most focusing on the NCAA Men's Tournament rather than the regular season. One study by Rhonda Magel and Samuel Unruh used basic linear and logistic regression to predict NCAA Tournament outcomes (Magel & Unruh, 2012). They aimed to identify the most significant in-game statistics influencing Division I men's basketball results, analyzing 280 games from the 2009–2011 seasons and focusing on differences in team stats like assists, turnovers, and rebounds. Two models were built: a least squares regression model to predict point spread and a logistic regression model to estimate win probability. Four variables were significant in both: assists, free throw attempts, defensive rebounds, and turnovers. Turnovers were the most influential. The models showed 94% accuracy on a 132-game test set and even worked on 2013 Tournament games, but their performance dropped to 62–68% when using pre-game stats. Our study attempts to improve forecasting by focusing on regular season performance, which is more stable than tournament results.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Brady T. West proposed the Ordinal Logistic Regression and Expectation (OLRE) method to predict expected tournament wins using predictors like winning percentage, point differential, strength of schedule, and wins against top 30 teams (West, 2006). He used an ordinal logistic regression and computed expected wins from the predicted probabilities across seven outcomes (0–6 wins). Applied to data from 2003–2007, the OLRE method was compared to a Bradley-Terry simulation approach. OLRE outperformed in 2006, while Bradley-Terry was better in 2007. The OLRE's limitations include sparse final-stage data and reliance on tournament structure. We apply similar methods to full-season performance and compare modern regularization to traditional approaches.

Michael J. Lopez and Gregory J. Matthews combined logistic regression with Las Vegas point spreads and KenPom efficiency metrics (Lopez & Matthews, 2015). They built two models with one using point spread (M1), and one using 15 possession-based stats (M2). Final predictions averaged both models. This combination improved prediction accuracy, and simulations showed that randomness plays a major role in tournament outcomes. Our study will model full-season success which involves different patterns. Additionally, their model inspired us to explore regularization for season-long performance.

Francisco J. R. Ruiz and Fernando Perez-Cruz developed a generative Poisson model tailored to NCAA Tournament predictions, incorporating latent team and conference-level offense and defense (Ruiz & Perez-Cruz, 2015). Match outcomes were modeled with Poisson-distributed scores using latent vectors. Home court advantage and conference trends were also included. Variational inference was used for parameter estimation. Their model outperformed betting markets and the Kaggle competition winner, and generalized well across seasons. Our study builds on ideas like conference-level statistics for season predictions.

Garvyn Jay Chua explored machine learning models for predicting NBA championship success which are methods applicable to modern college basketball (Chua, 2023). He tested six models: Logistic Regression, Random Forests, XGBoost, PCA, SVM, and Neural Networks. Cross-validation was used for tuning. Random Forests and Neural Networks achieved the highest accuracy (86–90%), and variables like blocks and playoff participation were more predictive than expected. Our study draws from these machine learning methods, particularly Neural Networks, and applies them to NCAA basketball. While these studies provide useful baselines, few directly compare modern regularization techniques over full regular-season data. Our approach expands this literature by examining both traditional and neural network models across multiple seasons.

3 Proposed Work

We obtained datasets from Kaggle for NCAA basketball seasons from 2022 to 2025. For year-to-year comparison, we structured the data so that each team appeared as a separate observation with features by year. One challenge was inconsistency in team naming across datasets, which we addressed by manually cleaning and standardizing team names before merging the data into a single dataframe using a Python script. After merging the datasets, we removed unrelated or redundant variables. For example, we excluded postseason data due to missing values and removed variables such as seed and effective field goal percentage (EFG), which were already represented in earlier numerical columns. To prevent data leakage, we removed columns closely related to wins, such as W, WAB, and others. We created a new column called win percentage, calculated as the number of wins divided by the total number of games played. We found the dataset to be balanced, as the win percentage variable exhibited an approximately normal distribution. A histogram of the data showed a symmetric spread centered around the mean, with a slight right skew and a mean of 0.58 (see Figure 1). This supports the suitability of the date for comparing the models chosen. The cleaned data was then imported into Google Colab for analysis. Three machine learning models will be applied to predict win percentages.

I. Lasso Regression (L1 Regularization): Lasso was chosen for its ability to perform both regularization and feature selection. Lasso reduces overfitting by shrinking feature coefficients to zero, eliminating irrelevant variables. This is useful for high-dimensional datasets, where many features may not significantly contribute to the target variable.

II. Ridge Regression (L2 Regularization): Ridge does not shrink coefficients to zero, but reduces their magnitude. This is beneficial if all features are expected to contribute to the model. This regularization technique helps prevent overfitting without discarding any features, making it a useful alternative to Lasso.

III. Neural Network with Dropout Regularization: Dropout regularization prevents overfitting by randomly dropping a fraction of the neurons during training, forcing the network to learn greater features. This is important due to the complex and potentially non-linear interactions within basketball data, where many features might interact. Dropout helps the model generalize better to unseen data which can be useful when predicting seasons that may differ from past patterns. By testing and comparing these models, we aim to identify the most effective approach for accurately predicting win percentages based on available data.

4 Experiments

4.1 Data Processing For L1 and L2

Since conference (CONF) is a categorical variable, we applied one-hot encoding so that it can be utilized in the L1 and L2 models. This transformed each conference into a separate binary variable, allowing the model to account for performance variations across conferences. After encoding the data, we selected only numeric features and applied standardization using z-score normalization. This ensured all features contributed equally to the model training. The standardized dataset was split into training and testing sets using an 80/20 split, where 20% of the data was used to evaluate model performance on unseen data. Both models will be fit to the training data and predictions will be made on the test set.

4.2 L1

Once the data processing was complete, a linear regression model with L1 regularization (Lasso) was created using scikit-learn (sklearn) and an implementation called Lasso. This model was optimized through coordinate descent which is the default optimization algorithm used in the Lasso function in Python. The α was set to 0.01 for a modest approach. This implementation yielded an MSE of 0.0055 and an R-squared value of 0.8043. This indicates that the model predicts close to the actual win percentages and that it predicts approximately 80% of the variation in team performance. Additionally, the model achieved an RMSE of 0.074 and MAE of 0.059. This suggests the model predictions are off by about 7.5 percentage points and an average absolute deviation of about 5.9% After training, the top ten non-zero coefficients were identified and ranked by magnitude. The five most influential features were Effective Field Goal Percentage Offense (EFG_O), Effective Field Goal Percentage Defense (EFG_D), Turnover Rate Defense (TORD), Turnover Rate (TOR), and Offensive Rebounds (ORB). From this analysis, EFG_D and TOR had strong negative coefficients. This means that the model indicated that teams with strong defensive shooting and offensive force turnover ability tend to result in teams losing games. In contrast, EFG_O, TORD, and ORB had positive associations with win percentage. This means that the model found offense, offensive rebounds, and defensive turnover rate to indicate a team's ability to win.

To assess the model fit visually, a residual plot with residuals versus predicted win percentage was obtained. The residual plot confirms that the lasso model's errors are randomly distributed around zero with no obvious pattern, suggesting that the linear assumptions of the model are fair and there is no existing major bias or misspecification(see Figure 2).

Furthermore, a Q-Q plot for the lasso model was also generated where the x-axis is the theoretical qualities from a perfect normal distribution and the y-axis is the lasso model's actual residuals for performance teams. From the plot, the points fall along the red line and even more so in the middle. This suggests that our residuals are approximately normally distributed. The tails slightly deviate, but this is not a major concern. Overall, the Q-Q plot confirms that the lasso model residuals are nearly normal, supporting the reliability of its predictions (see Figure 3).

To further refine the lasso model, a cross-validation version was trained using a sklearn implementation called LassoCV which automatically selects the best alpha using 5-fold cross-validation. This model was trained using the same features and yield new top coefficients as follows: Strength of Schedule (SOS_Rating), Adjusted Offensive Efficiency (ADJOE), Adjusted Defensive Efficiency (ADJDE), EFG_D, and EFG_O. From this analysis, SOS_Rating and EFG_D had strong negative coefficients. This means that the lasso model with CV indicated that schedule strengths and defensive efficiency provide the greatest insight into team loss. Meanwhile, ADJOE and EFG_O had strong positive coefficients. This means the model believes offensive efficiency and offensive effective field goals are significant indicators of team wins. The Lasso CV model was fine tuned at α = 0.0003, which yielded an MSE of 0.0044 and an R-squared value of 0.8446. This indicates that the model predicts closer to the actual win percentages than basic lasso and that it predicts approximately 84.5% of the variation in team performance. Additionally, the model achieved an RMSE of 0.066 and MAE of 0.051. This suggests the model predictions are off by about 6.6 percentage points and an average absolute deviation of about 5.1% which are both lower than the basic lasso model.

Again, we derived a residual plot and Q-Q Plot and, as expected, the model performed very well. Compared to the standard Lasso model, LassoCV showed less outliers and a better spread of residuals. The Q-Q plot further confirms this as there are minor deviations at the tails. This suggests cross-validation showed moderate improvement to the model based on these plots (see Figure 4, 5).

4.3 L2

Similarly, a linear regression model with L2 regularization (Ridge) was created using the Ridge implementation in sklearn. This model was optimized through closed form solutions which gives the optimal weights directly without needing iterative optimization. The α was set to 0.01 for a modest approach and for a fair comparison to lasso. This implementation yielded a MSE of 0.00448 and an R-squared value of 0.8419, indicating that the model accounted for approximately 84.2% of the variation in win percentage among teams. Moreover, the model achieved a RMSE of 0.0669 and a MAE of 0.0511, meaning the model's predictions were off by about 6.7 percentage points on average and had an average absolute deviation of 5.1 percentage points.

After training, the top ten predictors were identified based on the absolute values of their coefficients. The key predictors were SOS_Rating, ADJOE, ADJDE, EFG_D, and TORD. SOS_Rating, ADJDE, and EFG_D had negative coefficients. This means Ridge determined that teams with tougher schedules and stronger defensive stats tend to have lower win percentages. This potentially reflects the difficulty of high-caliber opponents. In contrast, ADJOE and TORD, had positive coefficients. This indicates that Ridge found offensive efficiency and turnover pressure significant in winning games.

To analyze the Ridge model visually and compare it to lasso, a residual plot and Q-Q plot were made. The Ridge residuals in the residual plot are around zero, but are more tightly clustered than Lasso. This is especially true near the middle of the predicted range. There's less vertical spread overall compared to Lasso which might mean there is more consistent error behavior across predicted values. This suggests Ridge may be offering slightly better generalization, especially for teams with average to slightly above-average win percentages. In the Q-Q plot, Ridge has a tighter lower tail but slightly more extreme positive residuals than Lasso. Otherwise, its residuals are normal and reliable. Lasso might have a slight edge in normality (see Figure 6, 7).

To fine tune the Ridge model, a RidgeCV model was trained using 5-fold cross-validation to select the optimal α. The RidgeCV model converged on an α value of 1.0, showing nearly identical performance metrics to the Ridge model with an MSE of 0.0045, RMSE of 0.0670, MAE of 0.0511, and an R-squared of 0.8415. Although the RidgeCV model selected α = 1.0 and the basic Ridge model had a manual α = 0.01, their almost equal performances just indicate the model is robust to moderate changes in regularization strength. This resulted in the top predictors from the RidgeCV model being consistent with those of Ridge. Moreover, the residual plot Q-Q norm plot for RidgeCV looked identical to the Ridge plots. This means that both Ridge and RidgeCV capture accurate, unbiased relationships and their errors are behaving the way linear regression assumes.

4.4 Data Processing For Neural Network:

A neural network model with dropout regularization was also developed to predict win percentage among college basketball teams. To prepare the data for modeling, a few preprocessing steps were performed. The dataset was split into training and testing sets using an 80/20 split respectively, similar to L1 and L2. Feature scaling was then applied, standardizing all predictor variables to have a mean of zero and unit variance. After scaling, both the input features and target variable were converted into PyTorch tensors to allow them to be fed directly into the neural network model. The target tensor was reshaped into a column vector to align with the expected output shape for regression tasks.

4.5 Neural Network

The network was implemented using PyTorch, structured as a feedforward fully connected network with two hidden layers containing 64 and 32 neurons. Each hidden layer applies ReLU activations followed by dropout layers to mitigate overfitting. Dropout randomly disables a fraction of neurons during training, forcing the model to generalize better by preventing it from relying too heavily on any single neuron or subset of features. The hyperparameters were selected through a manual grid search over several learning rates and dropout rates, optimizing primarily for RMSE performance on a held-out test set.

Using RMSE as the primary evaluation metric, the best model was found with a learning rate of 0.01 and a dropout rate of 0.3. This model achieved a MSE of 0.00392, RMSE of 0.0626, and MAE of 0.0487. These results indicate that, on average, the model's predictions deviate by about 6.3 percentage points, with an average absolute error of 4.9 percentage points. The model achieved an R-squared value of 0.86, meaning it explained approximately 86.0% of the variation in win percentage. Based on R-squared, the neural network outperformed both the Lasso and Ridge regressions on the same dataset (see Figure 8).

To evaluate the neural network's fit visually, we noted traditional residual plots and Q-Q plots were not as appropriate as neural network residuals are not necessarily expected to follow a theoretical distribution. Instead, we analyzed the evaluation metrics across the epochs. In doing so, the decreasing model loss function, MSE, MAE, and increasing R-squared confirms the model steadily improved during training. Additionally, we observed an initial negative R-squared value which is a unique phenomenon that occurs when the starting model fits far worse than predicting the mean value. This is an artifact of using R-squared on a non-linear model (see Figure 9).

Unlike Lasso or Ridge, the neural network does not offer direct interpretability through coefficients. However, its superior performance metrics suggest it captures complex interactions among features that the linear models cannot. A downside of using all features directly without feature selection is that it becomes difficult to determine which variables hold the most predictive influence. Nevertheless, this tradeoff is worthwhile for better performance in non-linear, high-dimensional settings.

To gain insight into feature importance, we used a permutation importance technique. In this method, each feature is shuffled independently while observing the resulting degradation in model performance (MSE). The features SOS_Rating, ADJOE, and TORD remained important as seen in the linear models. However, ADJDE and EFG_D became far less critical in the neural network, suggesting that non-linearity in other predictors can capture the significance of defensive efficiency metrics previously provided (see Figure 10).

5 Conclusion and Further Direction

After evaluating linear regression models using L1 regularization and L2 regularization, as well as a Dropout-Regularized Neural Network, several observations emerged about both predictive performance and model behavior. Lasso regression showed noticeable improvement after applying cross-validation, while Ridge regression performed consistently well even before tuning. This reflects its robustness for different regularization strengths. The neural network ultimately outperformed both linear models, capturing a greater proportion of variance and achieving lower prediction errors overall (see Figure 11).

After consideration of the metric summaries across all three models, it was concluded that although the neural network was the most accurate model, its 2% higher R-squared and 0.004–0.005 lower MSE suggests its gain was only modest. This indicates that linear models remain competitive in environments such as NCAA men's basketball stats. Lasso and Ridge are highly effective, especially when feature selection and regularization are crucial. The advantages of neural networks become more prominent when there are highly non-linear relationships.

Interpretability was a major tradeoff among models. Lasso and Ridge provided direct coefficient-based interpretations, allowing their key predictors to be easily identified. On the other hand, the neural network was unable to provide this same understanding. Permutation importance assisted, capturing offensive efficiency better and revealing that defensive metrics such as ADJDE and EFG_D are not as important in the neural network as they were in the linear models.

In environments where prediction accuracy is significant, such as team scouting, betting, and strategic planning, neural networks may be worth the interpretability tradeoff. Conversely, when actionable insights into player or team characteristics are prioritized over modest performance improvements, linear models provide a compelling alternative. Overall, we have highlighted not only performance differences but the tradeoff between interpretability and predictive strength. As NCAA Men's basketball continues to change with factors like the transfer portal, NIL, and roster turnover, the ability to change predictive models becomes more and more significant.

Future work could incorporate richer contextual data such as player injuries, rest days, management turnover, etc. to more accurately reflect the real-world context surrounding team performance. Furthermore, we would later aim to incorporate individual player data including not only physical skills but also, emotional factors such as fatigue and morale in order to fully capture the situational dynamics influencing team success. Additionally, we would like to explore time-sensitive models such as Recurrent Neural Networks (RNNs) which can capture more sequential data, in turn, taking into account the order of games. Whilst Graph Neural Networks (GNNs) inter team relationships and how these influence performance over the course of a season.

Finally, when expanding this model to inter-conference competitions, investigating interactions between conferences becomes important. Differences in conference strength can significantly influence how predictive certain team statistics certain team statistics are. For future work, we aim to analyze how the relative strength of different conferences impacts the predictive power of individual team metrics. By incorporating conference-specific features we can refine the model's ability to predict outcomes in inter-conferences matchups. This could involve making a custom metric for conference strength and exploring how those influence team-level statistics and in turn offering a more accurate prediction for cross-conference tournaments and head to heads.

6 Figures

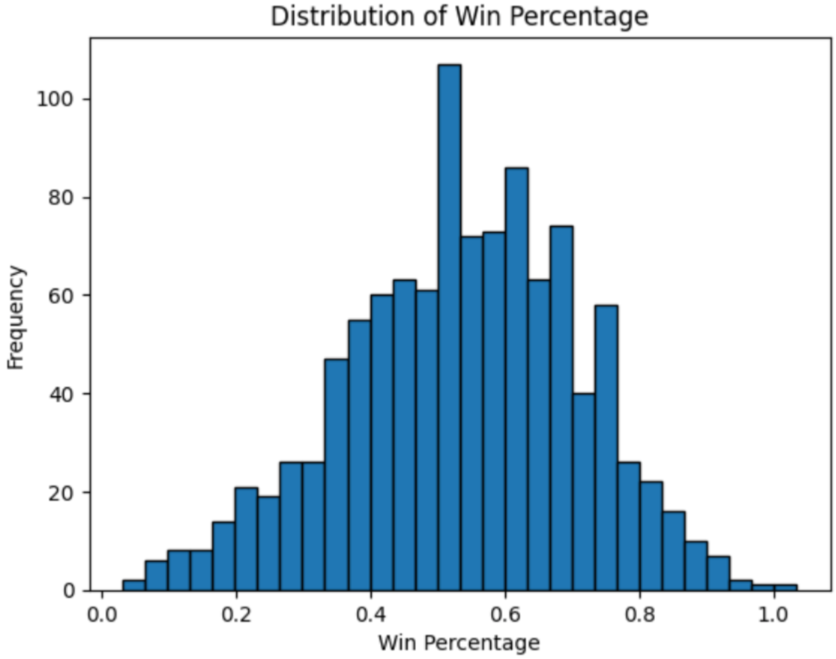


Figure 1: Distribution of win percentages.

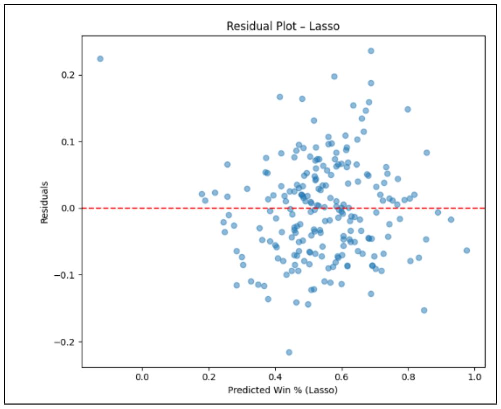


Figure 2: Residual Plot for Lasso Regression.

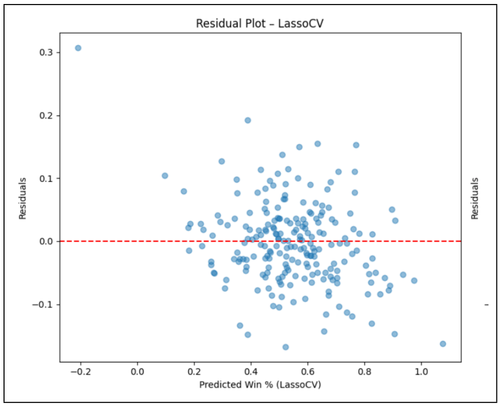


Figure 4: Residual Plot for LassoCV Regression

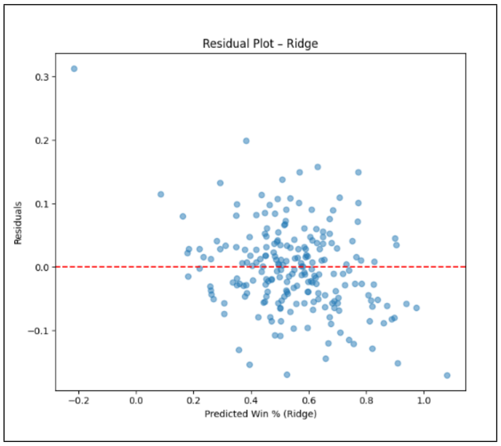


Figure 6: Residual Plot for Ridge Regression

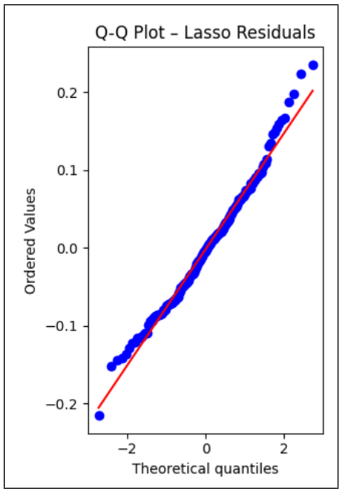


Figure 3: QQ Plot for Lasso Regression

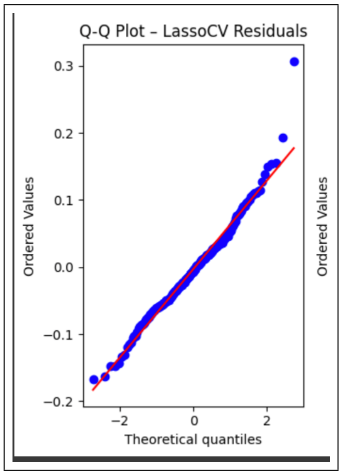


Figure 5: QQ Plot for LassoCV Regression

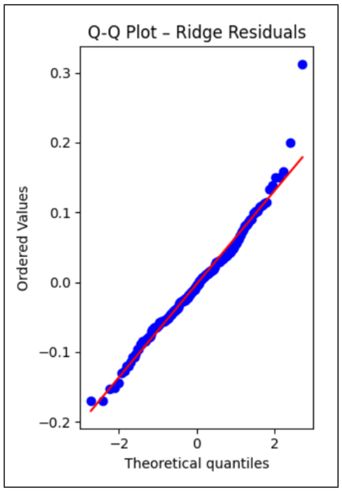


Figure 7: QQ Plot for Ridge Regression

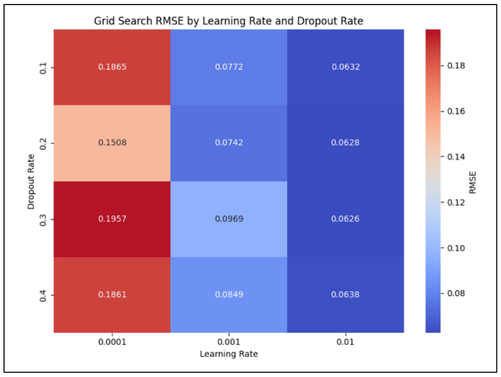


Figure 8: Grid Search based on RMSE for learning rate and dropout rate.

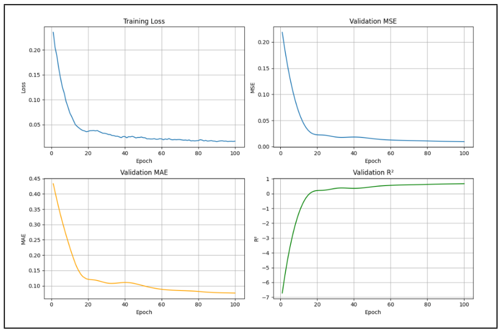


Figure 9: Line graphs showing the various training processes for Neural Networks.

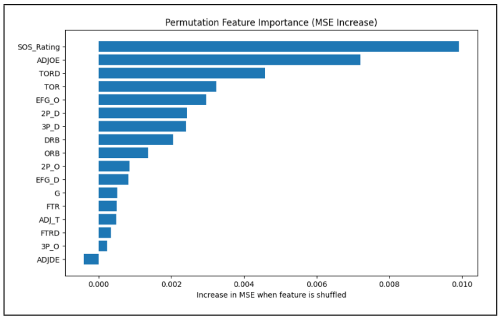
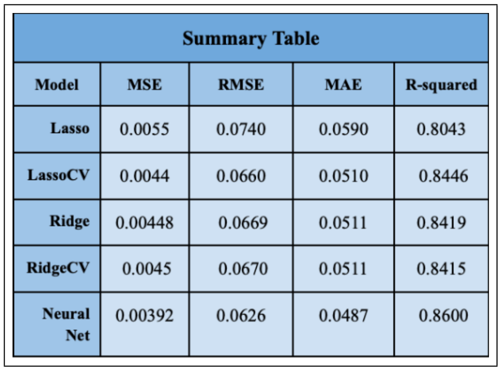


Figure 10: Feature Importance Based on Neural Networks.



Summary Table

Model	MSE	RMSE	MAE	R-squared
Lasso	0.0055	0.0740	0.0590	0.8043
LassoCV	0.0044	0.0660	0.0510	0.8446
Ridge	0.00448	0.0669	0.0511	0.8419
RidgeCV	0.0045	0.0670	0.0511	0.8415
Neural Net	0.00392	0.0626	0.0487	0.8600

Figure 11: Summary table of results from each model.

References

[1] Chua, G. J. (2023). ”Statistical analysis and predictive modeling in basketball: Unveiling key variables for championship success” (Order No. 30531176). Available from ProQuest Dissertations & Theses Global. (2832999785). Retrieved from http://libproxy.lib.unc.edu/login?url=https://www.proquest.com/dissertations-theses/statistical-analysis-predictive-modeling/docview/2832999785/se-2

[2] Lopez, Michael J., and Gregory J. Matthews. ”Building an NCAA men’s basketball predictive model and quantifying its success.” Journal of Quantitative Analysis in Sports, vol. 11, no. 1, 2015, pp. 5–12. De Gruyter, https://doi.org/10.1515/jqas-2014-0058.

[3] Magel, R., & Unruh, S. (2012). Predicting NCAA Tournament outcomes using basic statistics. Journal of Sports Analytics, 3(4), 142-157. Retrieved from https://www.scirp.org/journal/paperinformation?paperid=35927

[4] Ruiz, F. & Perez-Cruz, F. (2015). A generative model for forewarning outcomes in college basketball. Journal of Quantitative Analysis in Sports, 11(1), 39-52. https://doi.org/10.1515/jqas-2014-0055

[5] West, Brady T. ”A Simple and Flexible Rating Method for Predicting Success in the NCAA Basketball Tournament.” Journal of Quantitative Analysis in Sports, vol. 2, no. 3, 2006. https://doi.org/10.2202/1559-0410.1099.