ML Team 51 Final Report

NBA Player Performance Prediction

Team Members: Aadi Katta, Ishaan Kothari, Rana Myneni, Saahir Shaik, Varun Vudathu

Introduction

We will be aiming to predict player performance for points, rebounds, and assists in an upcoming game by using past player data from the NBA API, a client package that accesses the official data for NBA.com. The dataset provides every possible statistic about a player's performance since the modern stat-keeping era, so we can choose which measures we feel will best support our model. There is also the option to create our own customized statistics based off of more basic ones. A paper written by Matthew Houde talks about the process of writing a model for sports data, discussing which statistics are most important for winning. We can use this to guide our statistical research when diving into the data ourselves, before we preprocess it [1].

Problem

Predicting player performance is extremely hard to do with the multitude of statistics available. Our model will aim to consider as many of those variables as it can when making a decision about how a player will play in a game given their past performance. Previous studies have shown that machine learning techniques, such as Support Vector Machines and other approaches, are effective in predicting basketball game outcomes and handling the non-linear nature of player performance patterns [2]. Ultimately, we can use these statistics to correctly choose which players will play well or not, and use that information to make correct betting choices or add a new dimension to the modern statistical era, thereby improving sports as a whole.

Preprocessing Method #1

We used Standard Scalar to standardize features in the dataset. Standard Scalar is a tool used for standardizing features in datasets, transforming the features of the data such that they have a mean of 0 and a standard deviation of 1. This can help with various parts of the machine learning process. For instance, very large or small numbers can lead to numerical instability in machine learning models, and by applying standard scalar, we can reduce the drastic swings in the data. This effectively limits the power an outlier has in the learning process because it gets standardized, making sure our model(s) don't get confused or thrown off track when converging to a particular policy or learning parameters. Standardization and data preprocessing techniques are crucial for achieving stable predictions and improving the accuracy of sports analytics models, as demonstrated in recent work [3].

Preprocessing Method #2

We used Principal Component Analysis to reduce the dimensionality of our dataset. PCA is a technique that transforms the original set of features into a smaller set of uncorrelated variables called principal components, which capture the maximum variance in the data. This helps simplify the dataset by focusing on the most significant features while discarding noise and redundant information [4]. By applying PCA, we can help decrease issues related to multicollinearity among features and reduce the risk of overfitting in our models. PCA also helped speed up the training of our models, as with a multitude of statistics in our data beyond just points, rebounds and assists, the model would have to train on all these features, even if they weren't as informative.

Supervised Machine Learning Algorithm #1

The ML Algorithm we use is the MLPRegressor from Scikit-learn. MLPRegressor involves a neural network that does regression on the data. This model is a multi-layer perceptron and we have two strong reasons for why we wanted this to be our first model: the model is a supervised learning model and it can be used to solve complex nonlinear problems. We know that, for NBA players, performance can vary a lot for each game at the highest competitive level, making predictions incredibly nonlinear. One player might have a great three weeks and then goes into a slum the next few weeks. While there might be a subtle pattern into the makings of player performance, that pattern is not linear, where our model comes in. Also, we also knew we had to make use of a supervised learning model since all our data would come from previous games, which already has the results of the game and the player performance. Since neural networks are great supervised learning models, ubiquitous in machine learning applications, we decided to make sure our model was based on a neural network. With those two reasons, we decided to use MLPRegressor.

Supervised Machine Learning Algorithm #2

The next ML algorithm we implemented is the RandomForestRegressor from Scikit-learn. RandomForestRegressor is an ensemble learning method that constructs multiple decision trees during training and outputs the mean prediction of the individual trees. We chose this model because it is a supervised learning model, and it can handle complex datasets with high dimensionality and nonlinear relationships. In the context of NBA player performance, numerous factors influence a player's game statistics, and these factors can interact in complex, nonlinear ways. Random forests are effective in capturing these patterns because they consider various subsets of features and can model interactions between them. Also, as a supervised learning model, RandomForestRegressor is fitting since we have historical data with known outcomes for player performances. The ensemble nature of random forests helps in reducing overfitting, a common issue when dealing with high-variance datasets like sports statistics. By averaging the results of multiple decision trees, the model improves predictive accuracy and robustness. These reasons led us to believe that RandomForestRegressor would be a strong model for predicting NBA player performance.

Supervised Machine Learning Algorithm #3

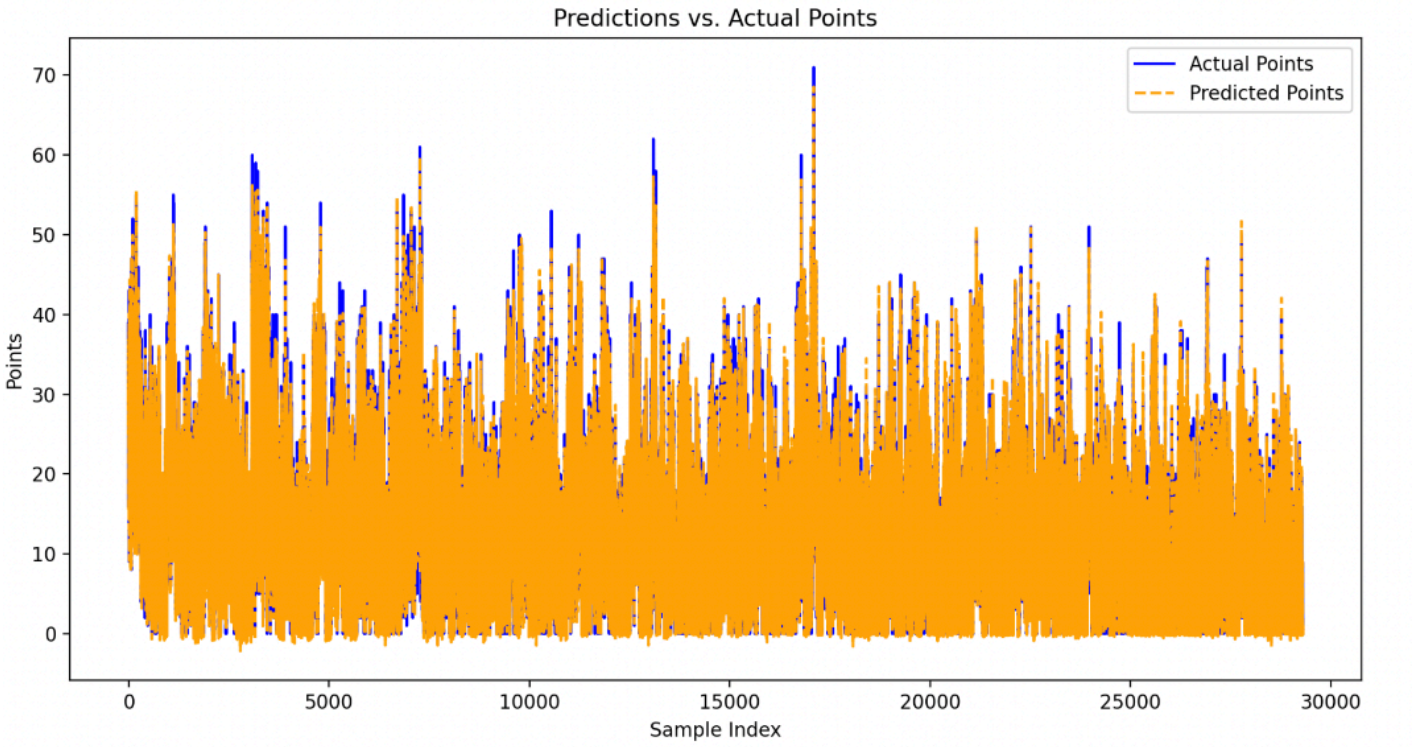
The last ML algorithm we utilized is Ridge Regression from Scikit-learn. Ridge Regression is a type of linear regression that includes L2 regularization, which adds a penalty equal to the square of the magnitude of coefficients. Since it is a supervised learning model, it is suitable for regression problems, and it can also handle multicollinearity among features. In our dataset, many player statistics are highly correlated, which can cause ordinary least squares regression models to have a high degree of numerical instability. Ridge Regression helps decrease this issue by shrinking the coefficients, thus reducing model complexity and preventing overfitting. However, since Ridge Regression is a linear model, we transformed the data by adding polynomial features. By incorporating the Polynomial Features transformer from Scikit-learn, we generated higher-degree terms and interaction terms of the original features. Raising the data to a higher dimension allowed Ridge Regression to still capture “linear” trends in the data. With this approach, we aimed to determine whether a linear model with polynomial terms could achieve an error similar to the more complex nonlinear models like Random Forest and MLP

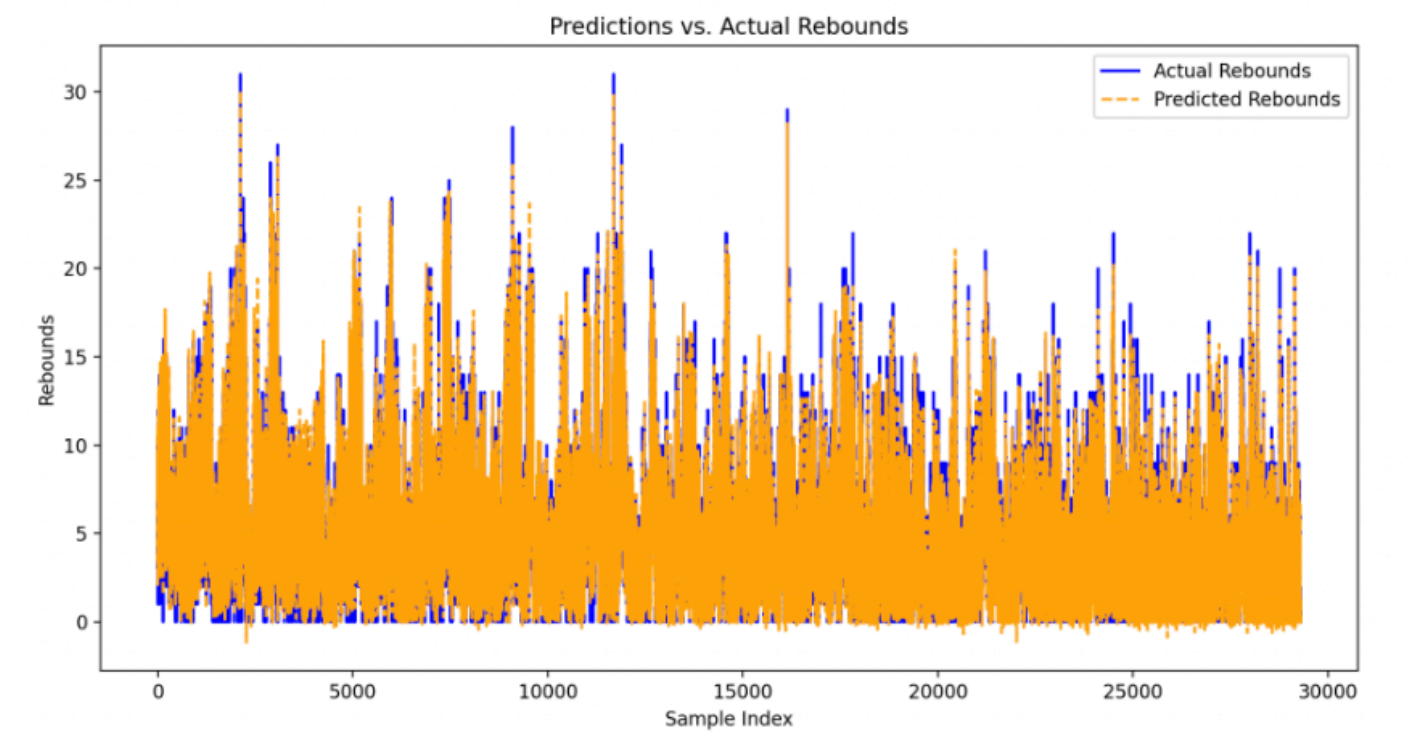
Results and Discussion

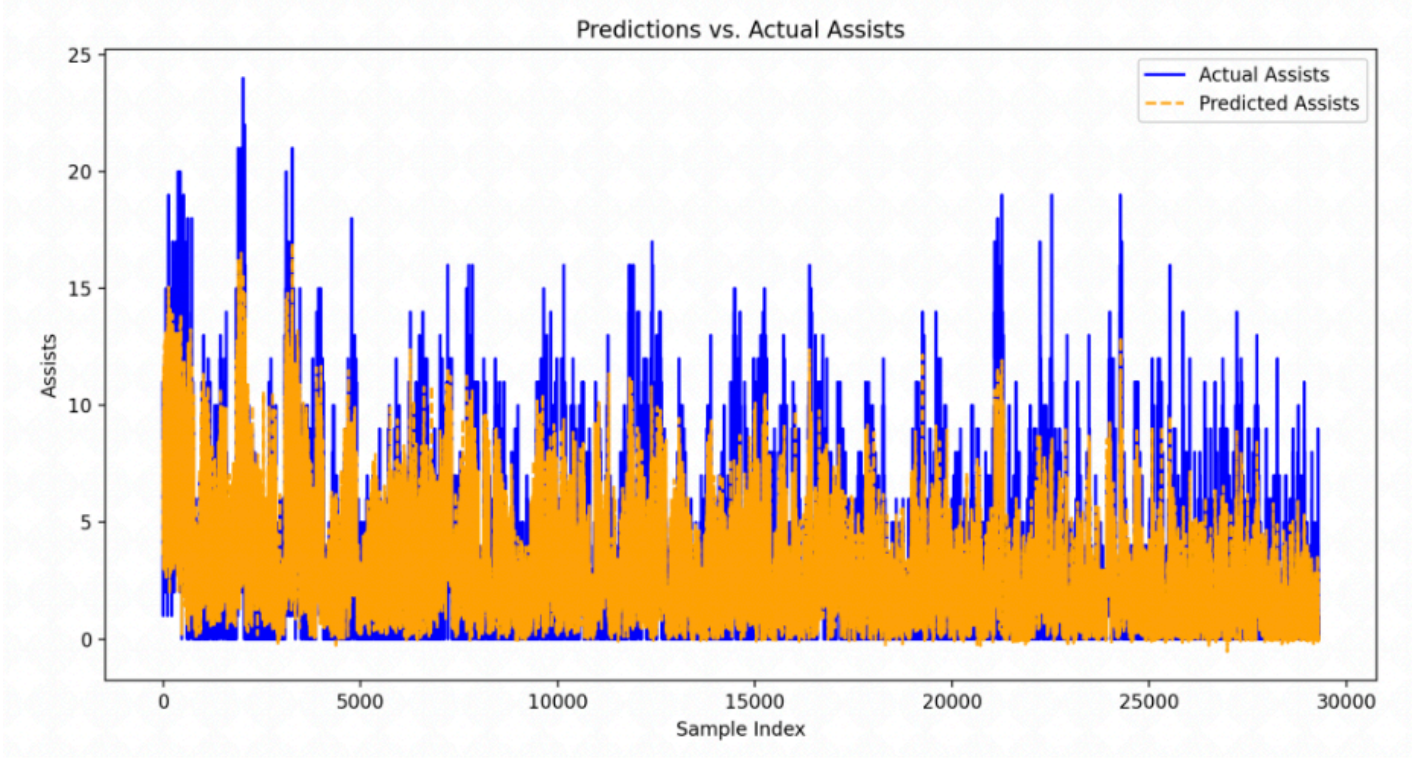
With the improved preprocessing method of PCA and the denormalized values of RMSE, we can truly see the predictive power of the three models we developed, MLP Regressor, Random Forest Regressor, and Ridge Regressor.

For our new MLP Regressor, we used the notes taken from our past report like feeding it more features, adding more layers, and tuning the other hyperparameters to obtain better results. For example, we added features like minutes played, field goals made, field goals attempted, offensive rebs, defensive rebs, plus minus, etc. to provide the model with more information to base its prediction on. With those new additions, we can see a tremendous improvement in the RMSE, our statistic of choice. Before these improvements, the RMSE value was very high meaning the predictive values to the actual values were off, but now the RMSE values are 1.2871, 0.7086, 1.8097 for points, rebounds, and assists, respectively. At first

glance, they may all look good, but truly the points and rebounds are the only ones that make sense. That is because points vary at a large scale from 0-70, however rebounds and assists vary from 0-15, which is a much smaller scale, meaning a difference of 1.2871 isn't as bad for points as 1.8097 is for assists. The visualizations below show the differences between the actual values and the predicted values for all of the labels.

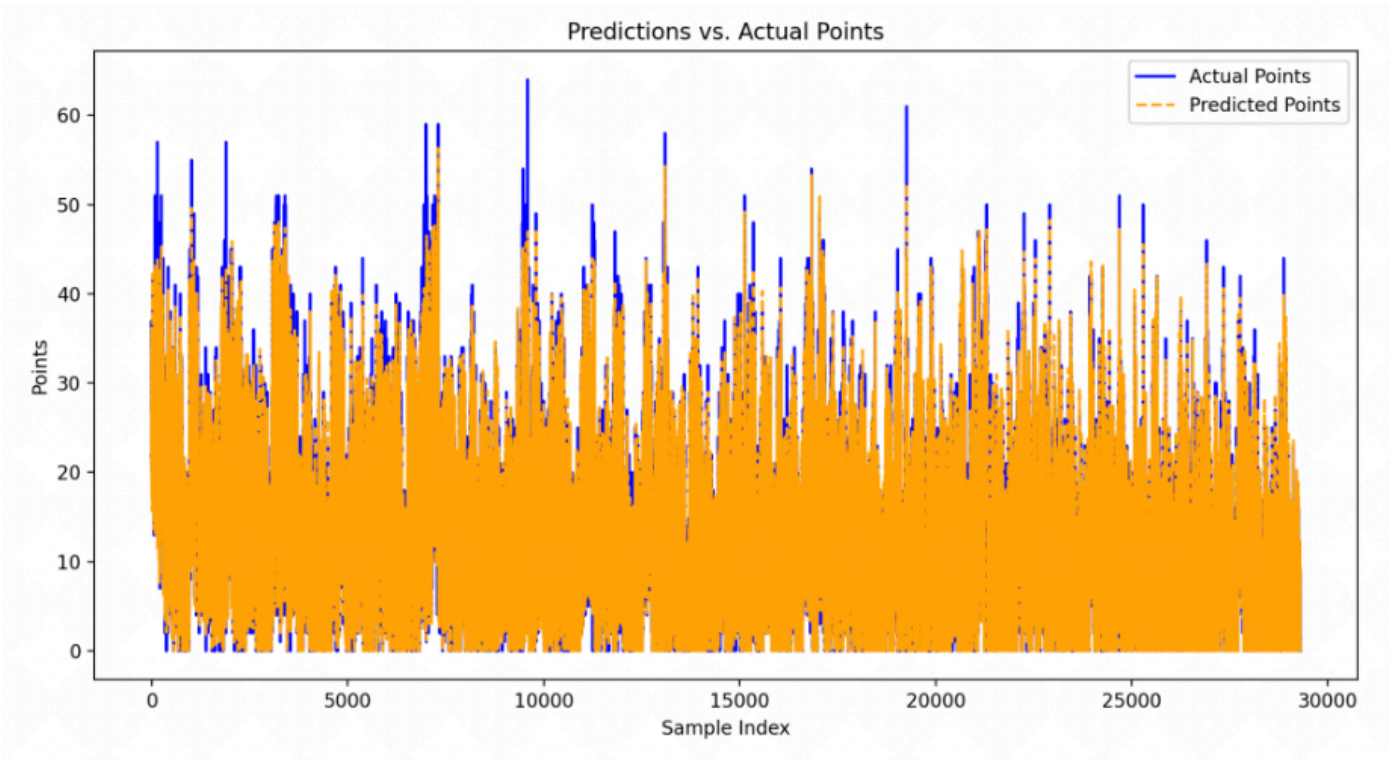


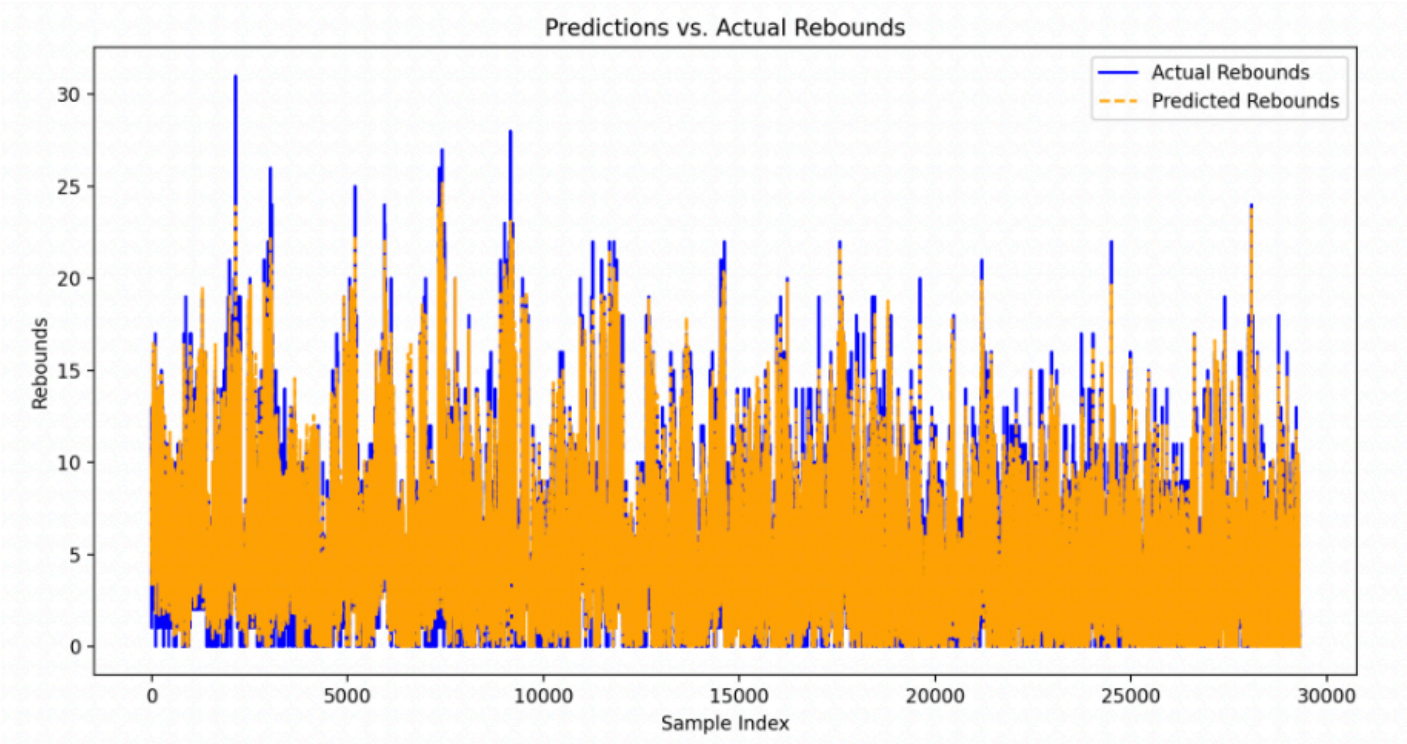


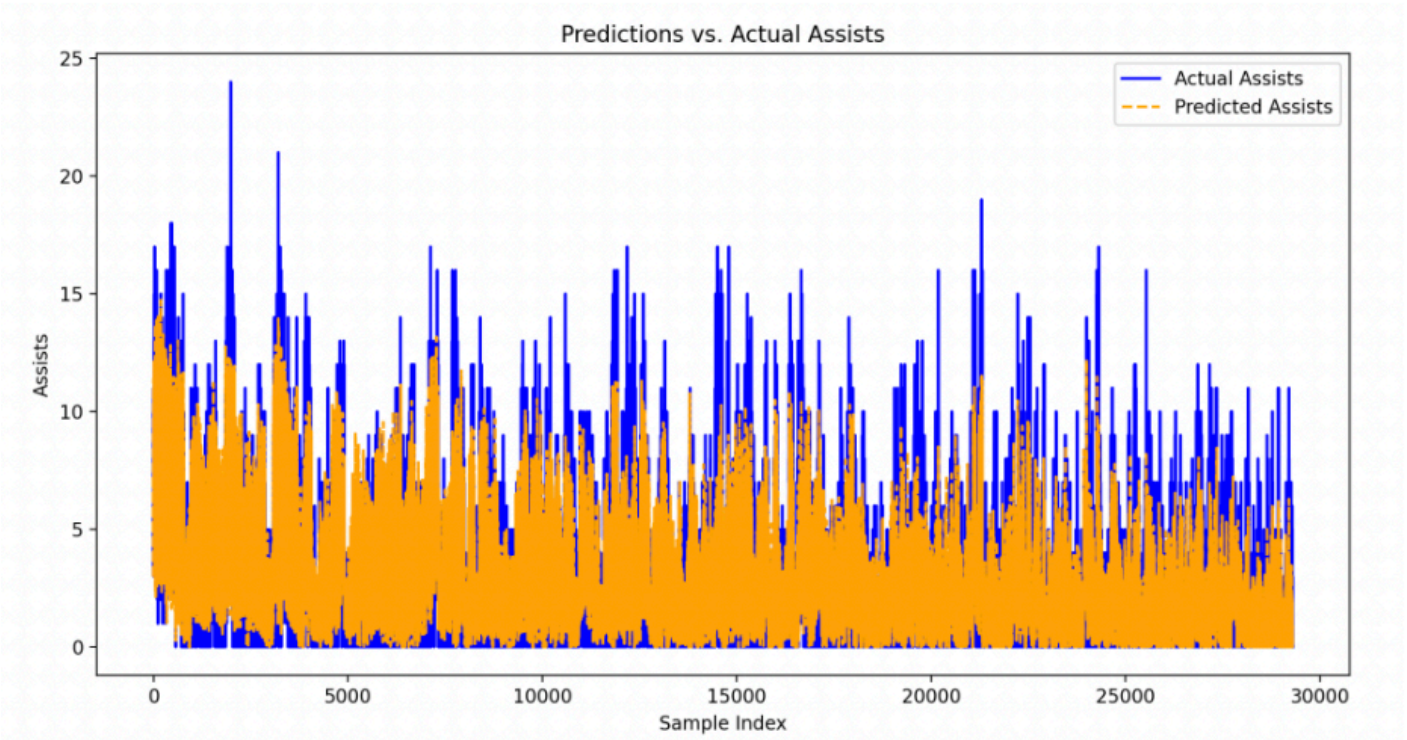


In comparison to the other models described, the MLP Regressor performed the best. Even if it was by a small margin, the ability to capture nonlinear trends and having a neural network architecture allowed for it to be more scalable and flexible to the data given without overfitting the data to the training data.

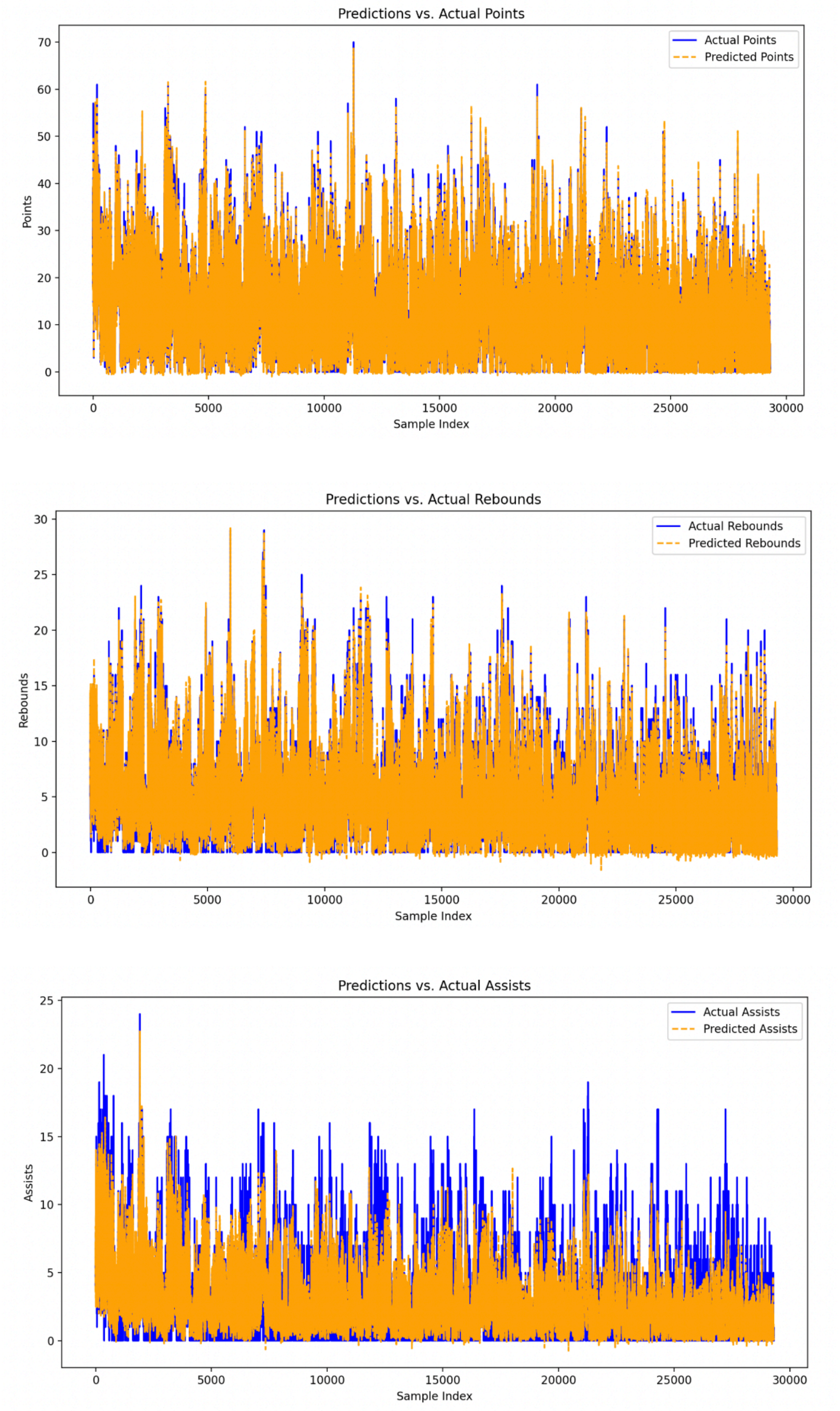
With the Random Forest Regressor model developed, we hoped to eliminate the problem with the assists label by allowing the model to randomly select features and subsets from the dataset. With the added processing similarly used in the MLP Regressor and the idea of creating diverse trees to mitigate the overfitting problem, we still resulted in pretty similar RMSE values in 1.4336, 0.7955, and 1.8448 for points, rebounds, and assists, respectively. The RMSE values are a little worse compared to the MLP Regressor in all three labels. A reason for this could be the diversity and size of our dataset. RF relies heavily on these two things and without these things all the trees basically result in the same outcome, causing worse metrics. Another reason that the values could be worse is due to not tuning the hyperparameters as much as we did with the MLP Regressor. The visualizations below show the differences between the actual values and the predicted values for all of the labels.







The last model implemented was the Ridge Regressor model. Using a L2 regularization, we eliminated the overfitting of data by penalizing large coefficients. Once again the idea of using Ridge Regression was to handle highly correlated data better to potentially improve the model performance. Lot of statistics in the NBA are very correlated like field goals attempted and field goals made or plus minus score to the overall stats of the player. Having a model that could handle this could potentially improve the RMSE. However, it resulted in almost the same values as the other two models, having values of 1.3474, 0.7493, and 1.8587 for points, rebounds, and assists, respectively. The values were a little worse than the MLPRegressor, but a little better than the Random Forest Regressor. The visualizations below show the differences between the actual values and the predicted values for all of the labels.



The reason for not having the best results could be due to the lack of complexity, meaning it can't handle non linear data as well as the MLPRegressor can.

Thoughts on Results, Comparison, and Next Steps:

Using unique models in order to try and find different results was a challenging task because the diverse models we used ended up having very similar accuracy for our different categories. Although this is not necessarily unexpected, we wished to have more diversity in accuracy in the models so we could find a definite model that performed significantly better than others. Each model performed so well because we applied the dimensional reduction strategy of PCA, helping the models figure out the most important distinctions between features, and our access to such a large dataset of previous games.

The MLP algorithm came on top via the RMSE results, while all of them were relatively close in prediction results. One key difference between the algorithms was that the random forest algorithm ended up taking significantly more time to train than the ridge regression or the MLP to achieve nearly the same results. The implication can be that random forest isn't as good a regression algorithm as the others for this particular problem: a reason for this could be that decision trees themselves are discrete, making it hard to have a decision for every single possible point. This required us to increase the complexity of each tree in the ensemble, therein increasing the time it took to train. The models overall performed great compared to our initial tests with MLP.

For our next steps, while we had lots of samples, we needed to get more data for different features, such as assists so we can improve our results for more than good results on points and assists. For all our models, we consistently have underperforming predictions for assists for this reason. Also, currently, our predictions come close to the exact result but on average our results have a margin of three. We'd like to reduce this margin to near one since that would be a more practical use of the model for actual predictions. Also, we can incorporate live game data and update our predictions in real time as that would be a more practical use of our predictions.

Gantt Chart

Refer to the Gantt Chart linked here.

Contribution Table

	Name	Checkpoint Contributions
0	Aadi Katta	Help with some models and presentation
1	Ishaan Kothari	Helped write results/discussion section
2	Rana Myneni	Implemented models, wrote report, completed presentation
3	Saahir Shaik	Helped record presentation
4	Varun Vudathu	Implemented PCA, helped with results/discussion

References

[1] M. Houde, "Predicting the Outcome of NBA Games," Honors Thesis, Bryant University, Smithfield, RI, USA, Apr. 2021. [Online]. Available: https://digitalcommons.bryant.edu/honors_data_science/1

[2] S. Jain and H. Kaur, "Machine learning approaches to predict basketball game outcome," 2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall), Dehradun, India, 2017, pp. 1-7, doi: 10.1109/ICACCAF.2017.8344688.

[3]K. Apostolou and C. Tjortjis, "Sports Analytics algorithms for performance prediction," 2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA), Patras, Greece, 2019, pp. 1-4, doi: 10.1109/IISA.2019.8900754.

[4] Andrzej Maćkiewicz, Waldemar Ratajczak, “Principal components analysis (PCA)”, Computers & Geosciences, Volume 19, Issue 3, 1993, Pages 303-342, ISSN 0098-3004, https://doi.org/10.1016/0098-3004(93)90090-R.