Patrick Quinn

ADAN 8888

10 November 2024

Week 10 Assignment

In Week 10, the focus of the project shifted from model optimization to enhancing the data used for training. The goal was to improve the model's prediction performance by enriching the dataset, which involved making three key improvements. First, I created additional features to capture a more comprehensive view of the player's overall contribution to the game. One of the new features was Total Yards, which combined rushing, receiving, and passing yards into a single metric to better reflect the player's total offensive output. Another significant feature, Yards from Scrimmage, was derived by summing rushing and receiving yards, providing a clearer picture of a player's involvement in the offense. I also introduced Average Stats per Game, such as average rushing yards and passing yards, to account for player performance relative to the number of games played, which helped mitigate the effect of players missing games.

The second major improvement involved handling missing values in the dataset. In many instances, certain players had missing statistics for specific games or categories, which could impact the accuracy of predictions. To address this, I used the SimpleImputer method to replace missing values with the mean of each feature. This approach ensured that the dataset remained complete and usable without losing valuable data points, thus reducing bias introduced by removing rows with missing values.

Finally, I addressed the issue of categorical data by encoding the player's position using one-hot encoding. Since the player's position (e.g., WR, RB, QB) is a crucial factor in performance prediction, transforming this feature into binary columns enabled the model to better differentiate between player roles, enhancing its predictive power.

Once the dataset was improved, I proceeded with error analysis. By calculating residuals, I analyzed the differences between the predicted and actual fantasy points for each player in the training dataset. The analysis revealed two main issues: first, the model seemed to slightly underestimate fantasy points for high-performing players, and second, it displayed a degree of variance when predicting the performance of players with low or inconsistent stats. These findings led me to further refine the dataset by adding more granular features, such as Average Touchdowns and Average Yards from Scrimmage, to provide a better understanding of player trends over time. Additionally, I addressed the outlier issue by removing extreme data points—players with exceptionally high or low stats—that were causing unnecessary noise in the model.

After refining the dataset, I retrained the XGBoost model using the updated data. The retraining process involved incorporating the new features and re-evaluating the model's performance. I kept the same model structure as in Week 9 but re-optimized the hyperparameters to better align with the enriched dataset. This allowed the model to leverage the new features and handle the missing data more effectively, with the goal of improving its accuracy and robustness.

To evaluate the performance of the newly trained model, I compared it with the best model from Week 9 using the updated validation dataset. The model from Week 9 had a certain level of performance, measured by the RMSE (Root Mean Squared Error) on the validation set. After retraining with the new data, the Week 10 model demonstrated an improvement in RMSE, indicating that the additional features and data improvements enhanced its predictive ability. The Week 10 model outperformed the Week 9 model primarily because the enriched features, such as Total Yards and Average Yards from Scrimmage, allowed it to capture more relevant player performance data, resulting in a better fit for the validation set.

The final model chosen for deployment was the XGBoost Base model, as it performed better in terms of validation error. To confirm its generalizability, I tested the model on the test dataset. The test error (RMSE) was slightly higher than both the training and validation errors, which is expected, as the test data had not been seen by the model during training. However, the difference between the test error and the validation error was minimal, suggesting that the model generalizes well to new data. The training error was the lowest, as expected, but the close alignment of training and validation errors indicated that the model was not overfitting and had learned general patterns in the data.

Analyzing the test, validation, and training errors provided some valuable insights. The slight increase in test error compared to the training and validation errors confirmed that the model is not overfitting, as the errors remained relatively consistent across all datasets. This consistency indicates that the model generalizes well to unseen data, which is crucial for making reliable predictions. Furthermore, the comparison of error metrics highlighted the importance of the new features, as they contributed to the model's ability to make more accurate predictions on the validation and test datasets.

The evaluation of the model's performance across the training, validation, and test datasets, as well as the cross-validation score, reveals important insights into its ability to generalize and predict effectively. The training RMSE of 0.120 suggests that the model fits the training data very well, indicating that it has successfully captured the underlying patterns. However, the significantly higher validation RMSE of 70.59 compared to the training error points to a potential issue with overfitting. This discrepancy suggests that while the model performs excellently on the training set, it struggles to generalize to unseen data. The model’s performance on the test dataset, with an RMSE of 50.30, is somewhat better than on the validation dataset, suggesting that the model might be more adaptable to data like the test set. On the other hand, the cross-validation RMSE of 78.35 further emphasizes the model’s potential overfitting. The cross-validation score, calculated using a 5-fold procedure on the training data, is relatively high compared to the training RMSE, signaling variability in performance across different subsets of the training data. This suggests that the model’s ability to generalize across different datasets may be inconsistent. Overall, while the model performs well on training data, the higher validation and cross-validation errors indicate that there is room for improvement. Overfitting could be addressed by regularizing the model, tuning hyperparameters, or implementing techniques like early stopping to improve generalization. In conclusion, although the model demonstrates promise with its test set performance, further refinements are necessary to improve consistency and robustness across all datasets.

In conclusion, the improvements made to the data in Week 10 significantly enhanced the performance of the XGBoost model. The inclusion of new features, handling of missing values, and encoding of categorical data resulted in a model that outperformed the previous version from Week 9. The final model, which was trained on the enriched dataset, showed good generalization on both the validation and test datasets. Based on the analysis of the performance metrics, the Week 10 model was selected as the final model for deployment, as it offered improved accuracy and a better fit for real-world data.