Patrick Quinn

ADAN 8888

13 October 2024

Week 6 Assignment

Model Approach and Justification

For this week's task, I chose a Random Forest model to predict fantasy football points. This decision is based on the Random Forest's strength in handling high-dimensional data, its ability to handle interactions between variables, and its natural capacity to work well with both continuous and categorical data. Given the variety of features, including Total Yards, Touchdown Efficiency, and Fantasy Points per Game, Random Forest is an appropriate model to capture the complexities and relationships within the dataset. Moreover, its ability to provide robust predictions while minimizing overfitting through bagging makes it a sound choice for this task.

The complexity of the Random Forest model lies in its ensemble structure, which combines multiple decision trees. Each tree contributes to the overall prediction, which adds complexity as the number of estimators increases. In addition, while Random Forests do not require as extensive parameter tuning as some other models, hyperparameters such as the number of trees, tree depth, and minimum samples for splitting do influence performance, as discussed below.

Hyperparameter Tuning

For hyperparameter tuning, I chose to evaluate three key parameters: the number of estimators, maximum tree depth, and minimum samples required to split a node. The number of estimators controls the number of trees in the forest, which directly affects the model’s performance and computational cost. Larger forests tend to generalize better but also increase the risk of overfitting. The maximum depth of the trees controls how deep the trees can grow before stopping. Deeper trees can capture more intricate patterns in the data but might lead to overfitting. Lastly, the minimum samples for splitting a node help control the model's complexity by limiting how finely the trees can split. Evaluating these parameters allowed me to find the best balance between model complexity and prediction accuracy.

Model Performance Metrics

For evaluating model performance, I used two metrics: Root Mean Squared Error (RMSE) and R^2 (coefficient of determination). RMSE is a popular choice for regression tasks as it provides a straightforward interpretation of the average error magnitude. Since my goal is to predict fantasy points, having a metric that gives a clear view of the average prediction error in the same unit (points) is essential. R^2, on the other hand, gives an indication of how well the model explains the variance in the target variable. Since Random Forest can be prone to overfitting, especially when the number of estimators is high or the trees grow deep, R^2 helps ensure that the model not only fits the training data well but also generalizes effectively to unseen data.

Calculating Metrics

To assess performance, I calculated RMSE and R^2 on both the training and validation datasets for three variations of the model: the base Random Forest model, the model with hyperparameters tuned via GridSearchCV, and a Random Forest model with Principal Component Analysis (PCA) applied. For each variation, the training and validation RMSE and R^2 were calculated. These metrics provide insights into how well the model fits the training data and how accurately it generalizes to the validation set.

Analysis of Training and Validation Metrics

As expected, the base Random Forest model produced strong results on the training dataset, with relatively low RMSE and high R^2. However, on the validation set, there was a slight increase in RMSE and a reduction in R^2, indicating some overfitting. The second variation, with hyperparameter tuning using GridSearchCV, improved the validation performance by reducing overfitting while maintaining solid results on the training set. By selecting the optimal values for the number of estimators, maximum depth, and minimum samples for splitting, the GridSearchCV-tuned model provided a better balance between complexity and performance. The third variation, where PCA was applied before training the Random Forest, did not significantly improve performance over the GridSearchCV-tuned model. In fact, in some cases, PCA reduced the model's ability to capture relevant patterns, as the dimensionality reduction may have discarded useful features.

Comparison of Variations

When comparing the three model variations, it becomes clear that while all performed well, the model with hyperparameter tuning consistently achieved better validation results than both the base model and the PCA-augmented model. Although the PCA model reduced the feature space, it did not lead to a significant improvement in performance, likely due to the Random Forest model's inherent ability to handle many features without requiring dimensionality reduction.

Selection of the Best Model

Among the three variations, the best model for this week's task is the Random Forest model with hyperparameters tuned via GridSearchCV. This model provided the best balance between training and validation performance, with the lowest validation RMSE and the highest R². By optimizing key hyperparameters, I was able to improve the model's generalization capabilities while keeping its complexity manageable.

Table of Metrics

Model	**Training RMSE**	**Validation RMSE**	**Training R^2**	**Validation R^2**
Base	18.4871920002891	30.64079352395721	0.995085923425118	0.988958846237974
Grid Search	16.9481953265351	30.67567504124581	0.995870029614116	0.988933693402881
PCA + Grid Search	30.30089734988608	159.2664810371110	0.9867988733775771	0.7016931607936405