## Decoding NBA Greatness

### Overview

This project aimed to analyze and predict NBA player performance metrics, specifically focusing on the Player Impact Estimate (PIE). The analysis involved data processing, outlier handling, multiple regression models, unsupervised learning techniques, and the development of machine learning models such as Random Forest and Neural Networks. The final analysis resulted in identifying the most important features that influence player performance and optimizing model performance through feature engineering and hyperparameter tuning.

### Data Processing

The dataset consisted of multiple NBA playoff CSV files that were merged into a single DataFrame. After processing, the player performance metrics, including offensive and defensive ratings, assist percentages, rebound percentages, and shooting efficiency (TS%), were used as features for model building. Key steps in data processing included:

- Outlier Detection and Removal: Outliers were identified using the Interquartile Range (IQR) method and removed to ensure model accuracy.
- Handling Skewness: Yeo-Johnson transformation was applied to correct for skewness in the data.
- Feature Engineering: Additional features, such as interaction terms (e.g., `OFFRTG/DEFRTG`) and polynomial features (e.g., `MIN_SQ` and `USG_SQ`), were created to improve model performance.

### Exploratory Data Analysis (EDA)

- Correlation Heatmap: A heatmap was created to visualize the relationships between various performance metrics. `EFG%` was dropped to avoid multicollinearity issues.
- Distribution Plots: Histograms and box plots were used to explore the distribution of each feature and visualize the spread of data points.

### Models Developed

<u>Model 1: Multiple Linear Regression (Baseline)</u>
**Techniques Applied**:

- A basic Ordinary Least Squares (OLS) regression model was used to predict Player Impact Estimate (PIE) based on NBA player performance metrics.

- The independent variables (features) included several player performance statistics like TS%, AST%, MIN, and DEFRTG.
- The relationship between the dependent variable (PIE) and independent variables was modeled as a linear combination of these features.

**Model Assumptions**:

1. **Linearity**: There is a linear relationship between the dependent variable (PIE) and the independent variables.
2. **Independence**: The observations are independent of one another.
3. **Homoscedasticity**: The variance of residuals (errors) is constant across all levels of the independent variables.
4. **Normality of Residuals**: The residuals (errors) should be normally distributed.
5. **No Multicollinearity**: There should not be a high correlation between independent variables. However, through the correlation heatmap, some multicollinearity was identified and addressed in Model 2.

**Analysis**:

- This baseline regression model explained a substantial portion of the variance in PIE, with an **$R^2$ of 0.84**.
- Residual analysis showed that most assumptions were reasonably met, but potential multicollinearity was a concern, leading to adjustments in Model 2.

<u>Model 2: Multiple Linear Regression (Optimized)</u>
**Techniques Applied**:

- To address multicollinearity, the feature AST/TO was removed, as it was found to be highly correlated with other features like AST% and TO RATIO.
- Further residual analysis and Variance Inflation Factor (VIF) analysis were applied to ensure that multicollinearity was minimized.

**Model Assumptions**:

- The same assumptions as Model 1 applied, but with an added focus on addressing multicollinearity.

**Analysis**:

- While the $R^2$ score remained the same (0.84), the optimization provided better residual diagnostics.

- By addressing multicollinearity, the model's assumptions were more aligned with OLS regression, making the model more reliable.

Model 3: Random Forest with PCA

- **Techniques Applied**:
  - **Principal Component Analysis (PCA)**: Dimensionality reduction was applied to reduce the number of features while retaining most of the variance in the data.
  - **Random Forest Regressor**: A Random Forest algorithm was applied to the PCA-transformed dataset. Random Forest is an ensemble learning method that constructs multiple decision trees and outputs the mean prediction of the individual trees.
  - **Hyperparameter Tuning**: Parameters like `n_estimators`, `max_depth`, and `min_samples_leaf` were optimized using GridSearchCV.
- **Model Assumptions**:
  - **No Assumptions on Data Distribution**: Unlike linear regression, Random Forest makes no assumptions about the linearity or normality of the data.
  - **No Multicollinearity Concerns**: Random Forest can handle correlated features well, but PCA was applied to reduce dimensionality and address potential multicollinearity issues.
- **Analysis**:
  - **MSE**: 5.59
  - **$R^2$**: 0.611
  - PCA was expected to reduce noise, but it removed too much relevant information. As a result, this model performed poorly compared to the Random Forest model without PCA, and thus it was not retained as a final model.

Model 4: Random Forest Without PCA (Final)
**Techniques Applied**:

- **Random Forest Regressor**: This model used the full dataset without PCA for dimensionality reduction.
- **Feature Importance**: After fitting the model, feature importance was extracted, and features like TS%, USG%, DREB%, and AST% emerged as the top contributors to PIE.

- **Hyperparameter Tuning**: Similar to Model 3, hyperparameters were optimized using GridSearchCV, which found that n_estimators=200 and max_depth=20 provided the best results.

**Model Assumptions**:

- **No Assumptions on Data Distribution**: Like other decision tree-based models, Random Forest does not require assumptions about linearity, normality, or homoscedasticity.
- **Robust to Multicollinearity**: Unlike linear models, Random Forest is robust to multicollinearity and does not suffer performance degradation due to correlated features.
- **Overfitting Protection**: Random Forest models can overfit to the training data, but this risk was mitigated through cross-validation and hyperparameter tuning.

**Analysis**:

- **MSE**: 2.86
- **$R^2$**: 0.802
- This model achieved the best performance among the Random Forest models. The use of the full feature set, combined with proper hyperparameter tuning, allowed this model to accurately predict PIE.

Model 5: Random Forest with Feature Engineering

**Techniques Applied**:

- New interaction features were added, such as OFFRTG/DEFRTG, to capture the relationship between offensive and defensive performance.
- Polynomial features were introduced for variables like MIN and USG%.
- Categorical binning was applied to AGE to categorize players into prime and veteran groups.

**Model Assumptions**:

- Same as for the Random Forest model, with no distributional assumptions.

**Analysis**:

- **MSE**: 2.85
- **$R^2$**: 0.801

- Despite the feature engineering, this model did not significantly outperform Model 4, suggesting that the additional features did not add much value to this prediction task.

Model 6: Neural Network with Adam Optimizer

- **Techniques Applied**:
  - A feedforward neural network was applied to the scaled dataset.
  - **Architecture**: The model contained three hidden layers with ReLU activation functions and L2 regularization to prevent overfitting.
  - **Optimizer**: The Adam optimizer was used for backpropagation, with mean_squared_error as the loss function.
  - **Hyperparameter Tuning**: The number of layers, neurons per layer, and learning rate were tuned using RandomizedSearchCV.
- **Model Assumptions**:
  - **No Distributional Assumptions**: Neural networks make no assumptions about the distribution or linearity of the data.
  - **Sufficient Data**: Neural networks generally perform best when there is a large amount of data, as they require substantial training to capture complex patterns.
  - **Overfitting Protection**: Overfitting is a common issue with neural networks, and this was mitigated using regularization techniques like L2 regularization.
- **Analysis**:
  - **MSE**: 1.98
  - **$R^2$**: 0.862
  - The Neural Network model outperformed all other models with the lowest MSE and highest $R^2$ score, making it the most accurate predictor of PIE. However, it is more complex and less interpretable compared to the Random Forest model.

Model 7: Neural Network with Adam Optimizer (with Decay)

- **Techniques Applied:**
  - **Feedforward Neural Network**: A neural network with three hidden layers was implemented to predict the Player Impact Estimate (PIE). The model included ReLU activation functions in the hidden layers.
  - **Regularization**: L2 regularization was applied to the network to help prevent overfitting.

    - **Optimizer**: The Adam optimizer was used for backpropagation, enhanced with decay to reduce the learning rate over time. Mean squared error was used as the loss function for optimization.
    - **Dropout & Batch Normalization**: Dropout was added between layers to prevent overfitting, while Batch Normalization was applied to improve training stability and convergence.
    - **Hyperparameter Tuning**: The model's architecture, including the number of neurons per layer, learning rate, regularization strength, and batch size, was tuned using RandomizedSearchCV. The search explored different configurations to find the best performing model.
- **Model Assumptions:**
    - **No Distributional Assumptions**: Neural networks are non-parametric models and do not assume any specific data distribution or linear relationships.
    - **Sufficient Data**: Neural networks generally require large datasets to perform optimally, as they aim to capture complex, non-linear patterns in the data.
    - **Overfitting Protection**: Overfitting was addressed using multiple techniques, including L2 regularization, dropout layers, and the Adam optimizer with decay.
- **Analysis:**
    - **Mean Squared Error (MSE)**: 2.03
    - **$R^2$ Score**: 0.859

The best-performing models are:

- Model 2 (Linear Regression): A robust model with simple interpretability.
- Model 4 (Random Forest Without PCA): Achieved the best balance between performance and interpretability, with an $R^2$ score of 0.802.
- Model 6 (Neural Network): Outperformed all other models with the highest $R^2$ score (0.862), but with increased complexity and less interpretability.

**Conclusion and Recommendations**

Based on the techniques and performance, **Model 4 (Random Forest without PCA)** is recommended for its balance between accuracy and interpretability. **Model 6 (Neural Network)** is the most accurate but is more challenging to interpret and may not be suitable for cases where interpretability is critical. Finally, **Model 2 (Optimized Linear Regression)** remains a solid, interpretable option that performs well but may not

capture non-linear relationships as effectively as the tree-based models or the neural network.

Further improvements can be made through additional feature engineering or by testing more complex neural network architectures.