Basketball Analytics: Regression and Classification for Player Scoring Performance

Akshar Sharma

University of North Carolina at Charlotte

ITSC 3156-001: Introduction to Machine Learning

Professor Minwoo ‘Jake’ Lee

December 9, 2024

Introduction

Problem Statement

The primary problem tackled in this project is predicting and classifying NBA players' performance based on historical player statistics and demographic information. Specifically, my project focuses on two key objectives: predicting the average points scored per game (pts) using regression and identifying high-performing players using classification models. This analysis aims to provide insights into player evaluation, enabling teams and coaches in today's league to make data-driven decisions regarding player recruitment, training focus, and performance optimization.

Motivation and Challenges

Basketball is a data-rich sport where player performance metrics directly impact team strategies and outcomes. Understanding and predicting player performance is very important for team managers, scouts, coaches, analysts, fans, etc. to make informed decisions. Accurately predicting player statistics, such as points scored per game, provides a competitive edge in roster management, talent recruitment, game preparation, and much more. However, this task comes with significant challenges. The variability in player performance due to factors like injuries, team dynamics, or playtime allocation introduces complexity. Additionally, handling missing or non-standardized data, like draft-related details and categorical features, requires detailed preprocessing and feature engineering to ensure accurate model predictions.

How Can We Approach This Problem?

To address this problem, I used a data-driven approach that combined data analysis, feature engineering, and machine learning models. The dataset was first cleaned and preprocessed to ensure consistent and complete information. I selected key features based on their relevance to player performance, and used techniques like one-hot encoding and normalization to handle categorical and numerical variables. Two machine learning algorithms were implemented: linear regression to predict points scored and logistic regression to classify players as high or low performers. The models were evaluated based on performance metrics such as MSE and R2 score for regression, and accuracy score, precision score, recall score, and F1 score for classification. This approach balances predictive accuracy and model interpretability, making it practical for real-world basketball analytics.

Data/Environment

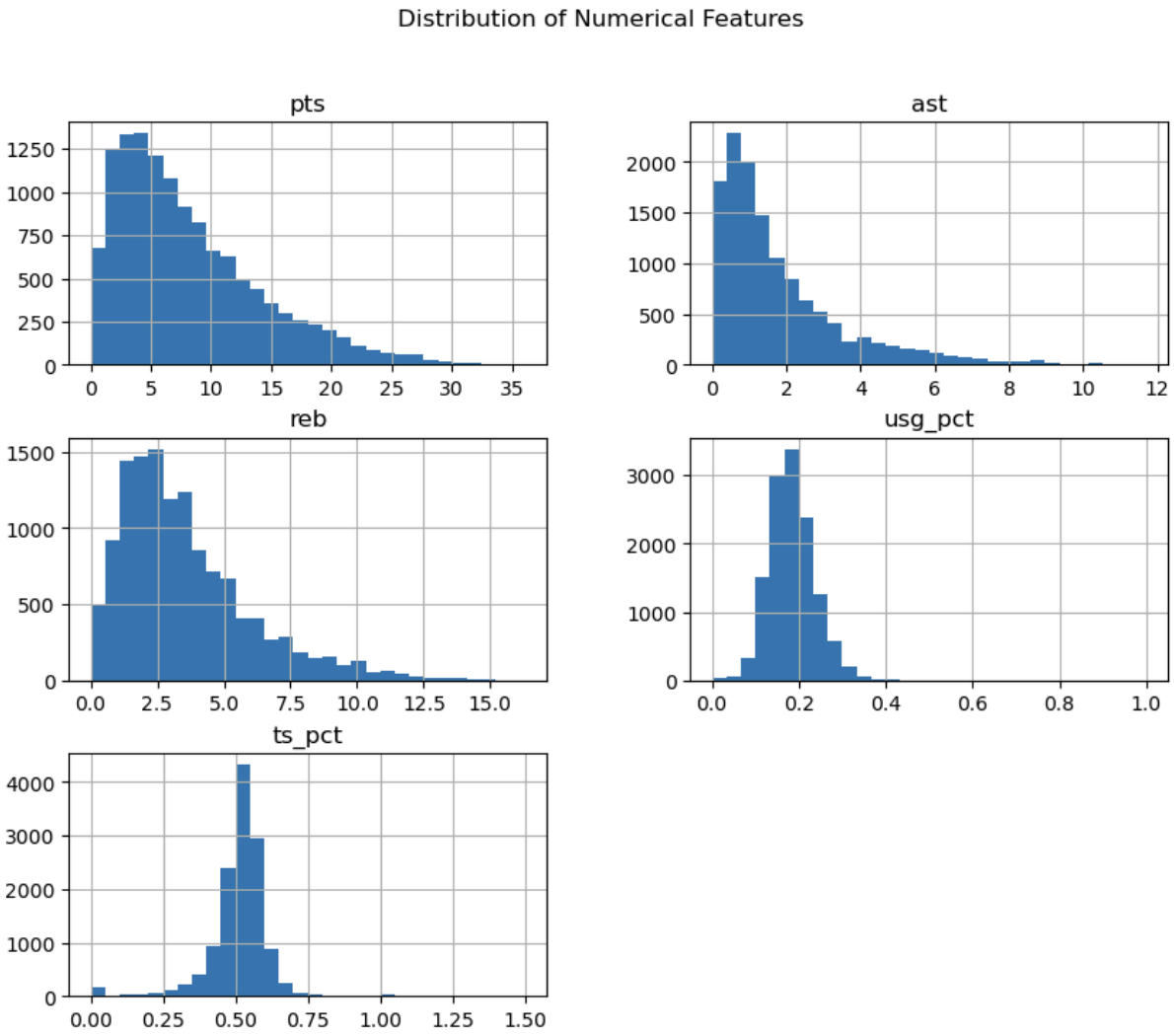
The dataset used for this project contains detailed NBA player statistics spanning multiple seasons. It has 12,844 rows and 22 columns, with features categorized into demographic attributes, performance metrics, and some advanced analytics. Additionally, contextual information such as college, country, and draft_year provides background on each player's journey to professional basketball. This dataset is highly relevant as it contains key information which can influence player performance and enable regression and classification tasks.

```
   Unnamed: 0       player_name team_abbreviation   age  player_height  \
0           0  Randy Livingston               HOU  22.0         193.04
1           1  Gaylon Nickerson               WAS  28.0         190.50
2           2      George Lynch               VAN  26.0         203.20
3           3    George McCloud               LAL  30.0         203.20
4           4      George Zidek               DEN  23.0         213.36

   player_weight               college country draft_year draft_round  ...  \
0      94.800728       Louisiana State     USA       1996           2  ...
1      86.182480  Northwestern Oklahoma     USA       1994           2  ...
2     103.418976        North Carolina     USA       1993           1  ...
3     102.058200         Florida State     USA       1989           1  ...
4     119.748288                  UCLA     USA       1995           1  ...

    pts  reb  ast  net_rating  oreb_pct  dreb_pct  usg_pct  ts_pct  ast_pct  \
0   3.9  1.5  2.4         0.3     0.042     0.071    0.169   0.487    0.248
1   3.8  1.3  0.3         8.9     0.030     0.111    0.174   0.497    0.043
2   8.3  6.4  1.9        -8.2     0.106     0.185    0.175   0.512    0.125
3  10.2  2.8  1.7        -2.7     0.027     0.111    0.206   0.527    0.125
4   2.8  1.7  0.3       -14.1     0.102     0.169    0.195   0.500    0.064

    season
0  1996-97
1  1996-97
2  1996-97
3  1996-97
4  1996-97

[5 rows x 22 columns]
```

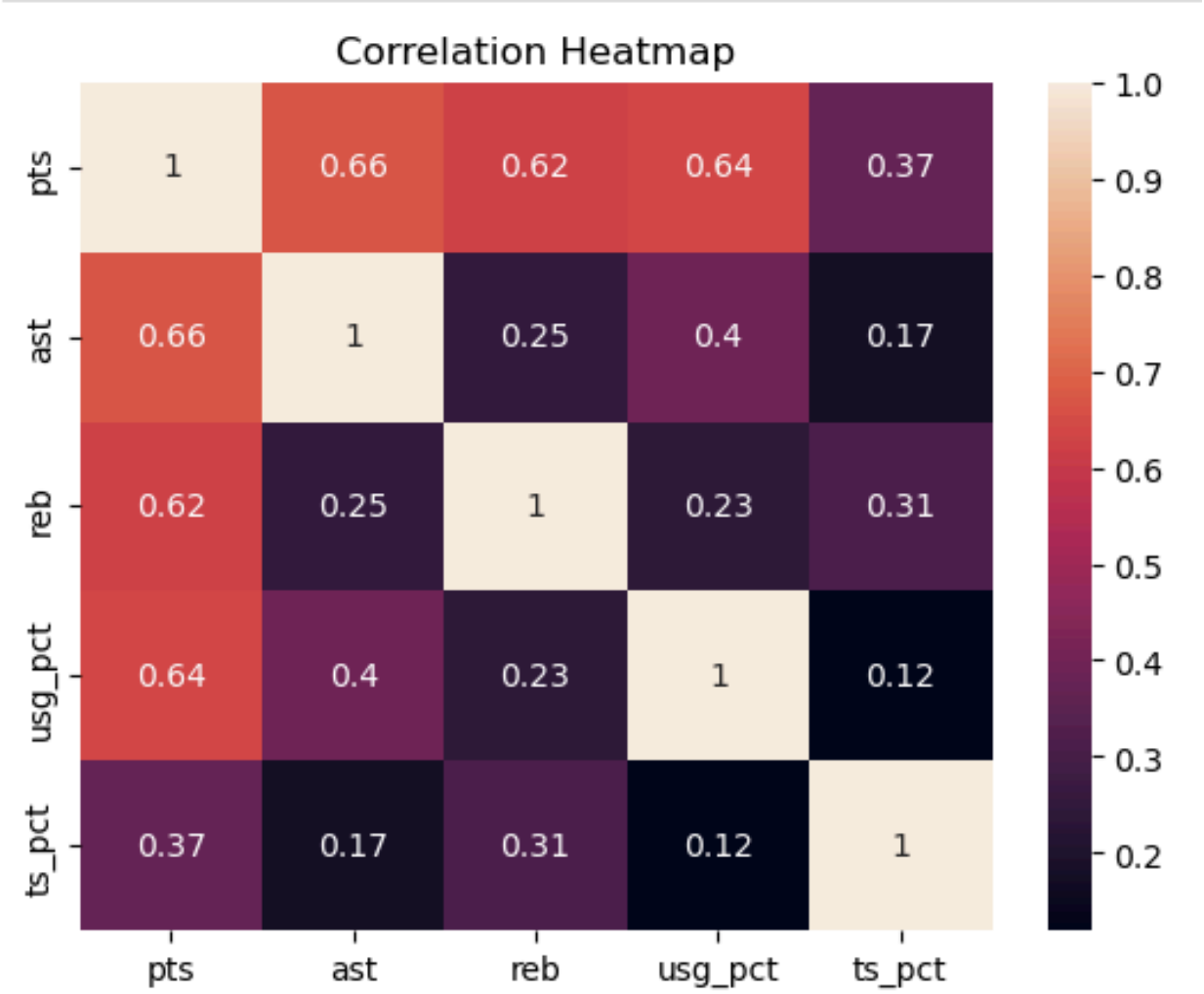
Data Visualization and Analysis

I used the following visualizations to understand the data distribution and relationships.

Histograms: The distributions of numerical features such as pts, reb, ast, usg_pct, and ts_pct show significant skewness. For instance, most players score fewer than 10 points per game, while only a few elite players exceed 20 points.



Correlation Heatmap: A heatmap revealed strong positive correlations between pts (points scored) and features like ast (assists) and reb (rebounds). This indicates that well-rounded players who contribute to different aspects of the game tend to score more points.



From the visualizations, I noticed and analyzed a few things:

- Most players have moderate to low contributions, while a small percentage of them dominate
- The correlation analysis confirms that assists and rebounds are strong predictors of scoring ability (pts), making them valuable features for predictive modeling

Data Preprocessing

1. *Feature Engineering*
    a. Creating a Binary Classification Target (High_Scorer):
        i. The median points scored (pts) was calculated for all players
        ii. A new binary column, High_Scorer, was created where players scoring above the median were assigned a value of 1 (high scorers), and those scoring below or equal to the median were assigned a value of 0
        iii. This feature engineering step introduced a target variable for the classification model to predict player performance categories effectively
2. *Handling Missing Values*
    a. Dropping Rows with Critical Missing Values:
        i. Rows with missing values in pts and High_Scorer were dropped to ensure the integrity of the regression and classification tasks
    b. Replacing Non-Numeric Strings with Missing Values:
        i. Columns such as draft_round and draft_number contained non-numeric strings ("Undrafted"). These were replaced with NaN
        ii. Following that, draft_round and draft_number were converted to numeric types
    c. Replacing Missing Values:
        i. Numeric columns, such as age, player_height, player_weight, draft_year, gp, and advanced metrics, had their missing values replaced with the median value of the column

ii. For categorical columns, such as college, country, season, and team_abbreviation, missing values were filled with the most frequent (mode) value

3. *Encoding Categorical Variables*
    a. Categorical features such as college, country, season, and team_abbreviation were transformed into a format compatible with machine learning models using One-Hot Encoding
        i. The new encoded columns were merged with the existing numerical data to form the final feature set
4. *Splitting Data*
    a. The dataset was split into training and testing sets for both regression and classification tasks:
        i. For regression, the target variable was pts (average points scored per game)
        ii. For classification, the target variable was High_Scorer (binary high/low scorer classification)
        iii. An 80-20 split was applied to separate training and testing data, ensuring adequate data for training while preserving a testing set for evaluation
5. *Scaling Numerical Features*
    a. To ensure uniform contribution of all numerical features to the model, MinMaxScaler was applied:
        i. Many columns were scaled to a range of [0, 1]

ii. This normalization step prevented features with larger ranges from disproportionately influencing the model during training

6. *Final Data Preparation*

   a. After preprocessing, the following datasets were prepared:

      i. Regression Data:

         1. X_train_reg_final and X_test_reg_final for feature matrices
         2. y_train_reg and y_test_reg for target values (points scored)

      ii. Classification Data:

         1. X_train_clf_final and X_test_clf_final for feature matrices
         2. y_train_clf and y_test_clf for target values (High_Scorer)

Methods

Linear Regression

Linear Regression was chosen as the algorithm for predicting players' average points per game. It is a supervised learning technique that models the relationship between a dependent variable and one or more independent variables by fitting a straight line to the data.

1. *Model Details*
    a. The algorithm assumes a linear relationship between the target variable (pts) and the features
2. *Why Linear Regression?*
    a. Simple to understand
    b. Effective for continuous target variables when there is a linear relationship with predictors.
3. *Implementation*
    a. Features were preprocessed using normalization and one-hot encoding
    b. The model was trained on the training set (X_train_reg_final and y_train_reg) and evaluated on the test set (X_test_reg_final and y_test_reg)

Logistic Regression

Logistic Regression was utilized to classify players as high scorers (High_Scorer = 1) or low scorers (High_Scorer = 0). Unlike linear regression, logistic regression predicts the probability of a binary outcome.

1. *Model Details*
    a. Logistic Regression uses the sigmoid function to map predicted values to probabilities between 0 and 1
    b. The decision boundary is set at 0.5, meaning players with a probability greater than 0.5 are classified as high scorers
2. *Why Logistic Regression?*
    a. Well-suited for binary classification problems.
    b. Interpretable coefficients that indicate the contribution of each feature to the probability of being a high scorer
3. *Implementation*
    a. Features were preprocessed using normalization and one-hot encoding
    b. The model was trained on the classification dataset (X_train_clf_final and y_train_clf) and evaluated on the test set (X_test_clf_final and y_test_clf)
4. *Confusion Matrix*
    a. The confusion matrix showed the number of true positives, true negatives, false positives, and false negatives.

Results

1. Experimental Setup

a. Training and Testing Split

i. The dataset was divided into training (80%) and testing (20%) subsets for both regression and classification tasks.

b. Evaluation Metrics:

i. For regression, performance was evaluated using:

1. Mean Squared Error (MSE): Measures the average squared differences between actual and predicted values.
2. R^2 Score: Indicates the proportion of variance in the target variable explained by the model.

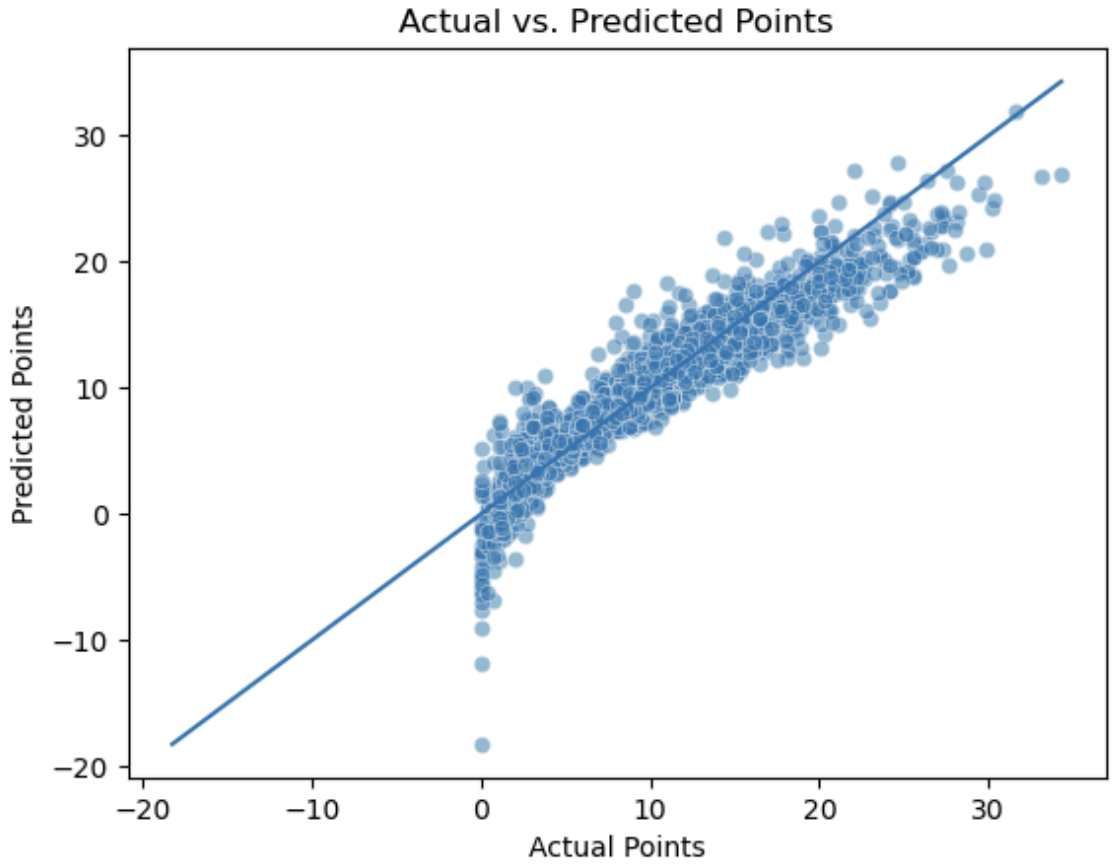
ii. For classification, the following metrics were used:

1. Accuracy: The percentage of correct predictions.
2. Precision: The proportion of true positives among predicted positives.
3. Recall: The proportion of true positives among actual positives.
4. F1 Score: The harmonic mean of precision and recall, balancing both metrics.

2. Test Results

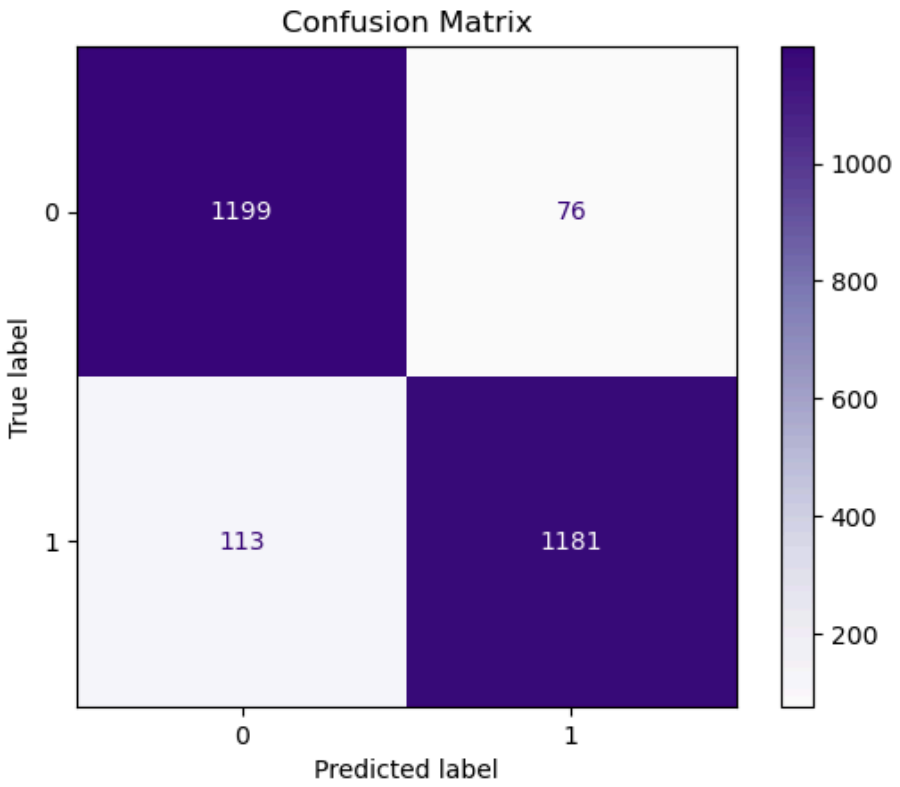
a. Regression:

```
Linear Regression MSE: 3.92730818180357
Linear Regression R² Score: 0.8928014363644116
```



b. Classification:

```
Logistic Regression Classifier Accuracy: 0.9264305177111717
Logistic Regression Classifier Precision: 0.939538583929992
Logistic Regression Classifier Recall: 0.9126738794435858
Logistic Regression Classifier F1 Score: 0.9259114072912583
```




```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.94      0.93      1275
           1       0.94      0.91      0.93      1294

    accuracy                           0.93      2569
   macro avg       0.93      0.93      0.93      2569
weighted avg       0.93      0.93      0.93      2569
```

3. Observations and Discussion

a. *Regression Observations:*

   i. The model performs well in predicting pts, with a high R-squared score of 0.89, indicating that most of the variance in player points is explained by the features

   ii. Outliers may be slightly impacting the accuracy, as seen in the scatterplot

b. *Classification Observations:*

   i. The high accuracy of 92.6% demonstrates the model's strength in identifying high and low scorers

   ii. Precision of 93.9% suggests that the model rarely misclassified low scorers as high scorers

   iii. Recall of 91.3% indicates it successfully identifies the majority of true high scorers

c. *Experiments to Support Analysis:*

   i. The scatterplot and confusion matrix directly validate the reliability of the models

   ii. The classification report reinforces the model's balanced performance

Conclusion

1. Concluding Remarks

This project tackled the challenge of predicting and classifying NBA players' performance using historical data. By using Linear Regression, I accurately predicted players' average points scored, achieving an R-squared score of 0.89. Similarly, Logistic Regression did well in classifying high scorers, achieving an F1 score of 92.6%. These results highlight the potential of machine learning models to provide insights in sports analytics.

2. Lessons Learned

I learned about the importance of data preprocessing, feature engineering, model evaluation, and lots more. Addressing missing values, handling categorical variables, and normalizing numerical features were important steps in ensuring the models' success. Furthermore, understanding the strengths and limitations of each model allowed me to use my methods in order to fit the problem requirements.

3. Challenges and Solutions

During the project, I also faced several challenges, particularly in handling missing data and non-standardized features. For example, columns like draft_round and draft_number included non-numeric entries, such as "Undrafted," which required transformation. Similarly, skewed data distributions, especially in the target variable pts, required normalization and feature selection to avoid model biases. By addressing these issues, I was able to receive a reliable outcome.

I’m excited to see the application of machine learning in sports analytics in the future and would definitely love to keep working on more projects like this one. It was a very rewarding learning experience.

References

GeeksforGeeks. "Metrics for Machine Learning Models." *GeeksforGeeks*, 03 July 2024, https://www.geeksforgeeks.org/metrics-for-machine-learning-model/. Accessed 3 Dec. 2024.

GeeksforGeeks. "ML Feature Scaling – Part 2." *GeeksforGeeks*, 21 Dec. 2023, https://www.geeksforgeeks.org/ml-feature-scaling-part-2/. Accessed 5 Dec. 2024.

GeeksforGeeks. "ML – One Hot Encoding." *GeeksforGeeks*, 2 Nov. 2024, https://www.geeksforgeeks.org/ml-one-hot-encoding/. Accessed 5 Dec. 2024.

GeeksforGeeks. "Regression Metrics." *GeeksforGeeks*, 29 July 2024, https://www.geeksforgeeks.org/regression-metrics/. Accessed 3 Dec. 2024.

GeeksforGeeks. "Sklearn Classification Metrics." *GeeksforGeeks*, 18 Oct. 2023, https://www.geeksforgeeks.org/sklearn-classification-metrics/. Accessed 4 Dec. 2024.

Justinas. "NBA Players Data." *Kaggle*, https://www.kaggle.com/datasets/justinas/nba-players-data/data. Accessed 2 Dec. 2024.

Matplotlib Documentation. https://matplotlib.org/. Accessed 8 Dec. 2024.

NumPy Documentation. https://numpy.org/. Accessed 3 Dec. 2024.

Pandas Documentation. https://pandas.pydata.org/. Accessed 4 Dec. 2024.

Scikit-learn Documentation. https://scikit-learn.org/stable/. Accessed 8 Dec. 2024.

Seaborn Documentation. https://seaborn.pydata.org/. Accessed 6 Dec. 2024.

Stack Overflow. https://stackoverflow.com/. Accessed 8 Dec. 2024.

Acknowledgment & Source Code

Libraries such as Scikit-learn, NumPy, Pandas, Matplotlib, and Seaborn were used in this project for providing tools for data analysis and machine learning. Additionally, resources like GeeksforGeeks, Stack Overflow, and Kaggle were used in guiding the implementation and supplying the dataset for this project.

Source Code:

https://github.com/aksharrsharma/MLFinalProject