Comprehensive Report

1. Exploratory Data Analysis and Preprocessing

a. Dataset Overview

- The Dataset, `Hitters.csv`, was loaded and examined.
- Initial Dataset shape: `(263, 20)`
- Displayed the first few rows of the dataset using `df.head()`.

b. Handling Missing Values

- Checked for missing values using `df.isna().sum()`.
- Removed rows with missing values using `df = df.dropna()`.
- Reset index after dropping missing values: `df = df.reset_index(drop=True)`.

c. Column Mapping for Categorical Features

- Mapped Categorical Features (`League`, `Division`, `NewLeague`) to numerical values.
- Displayed the mapping for each categorical column.

d. Data Statistics

- Displayed general information about the dataset using `df.info()`.
- Described the statistical summary of the dataset using `df.describe()`.
- Computed the correlation matrix using `df.corr()`.

2. Principal Component Analysis (PCA)

a. Standardization

Feature Standardization

- **Separation of Features and Target Variable:**
    - We separated the features `X` and target variable `y` to prepare for the Principal Component Analysis (PCA) process.
- **Standardization of Features:**
    - Features were standardized to ensure a consistent scale across variables.
    - Standardized features: `X_standardized = (X - X.mean()) / X.std()`.
- **Why Standardize Features Before PCA?**
    - Standardizing features is crucial for PCA because it ensures that all variables contribute equally to the analysis.
    - PCA is sensitive to the scale of the variables, and standardization helps prevent dominance by variables with larger scales.

    - It facilitates a more accurate representation of the covariance structure and aids in identifying the principal components effectively.

Target Variable Standardization (Not Performed)

- We did not standardize the target variable `y` in this context.
- **Reason:**
    - Standardizing the target variable is unnecessary for PCA.
    - PCA focuses on capturing variance in the features, and the scale of the target variable does not impact this process.
    - Standardizing the target could distort the interpretability of the regression coefficients when interpreting the original feature space.
- **Summary:**
    - Standardizing features ensures a meaningful PCA outcome, while the target variable remains unstandardized to maintain interpretability in subsequent regression analyses.

b. Eigenvalue and Eigenvector Calculation

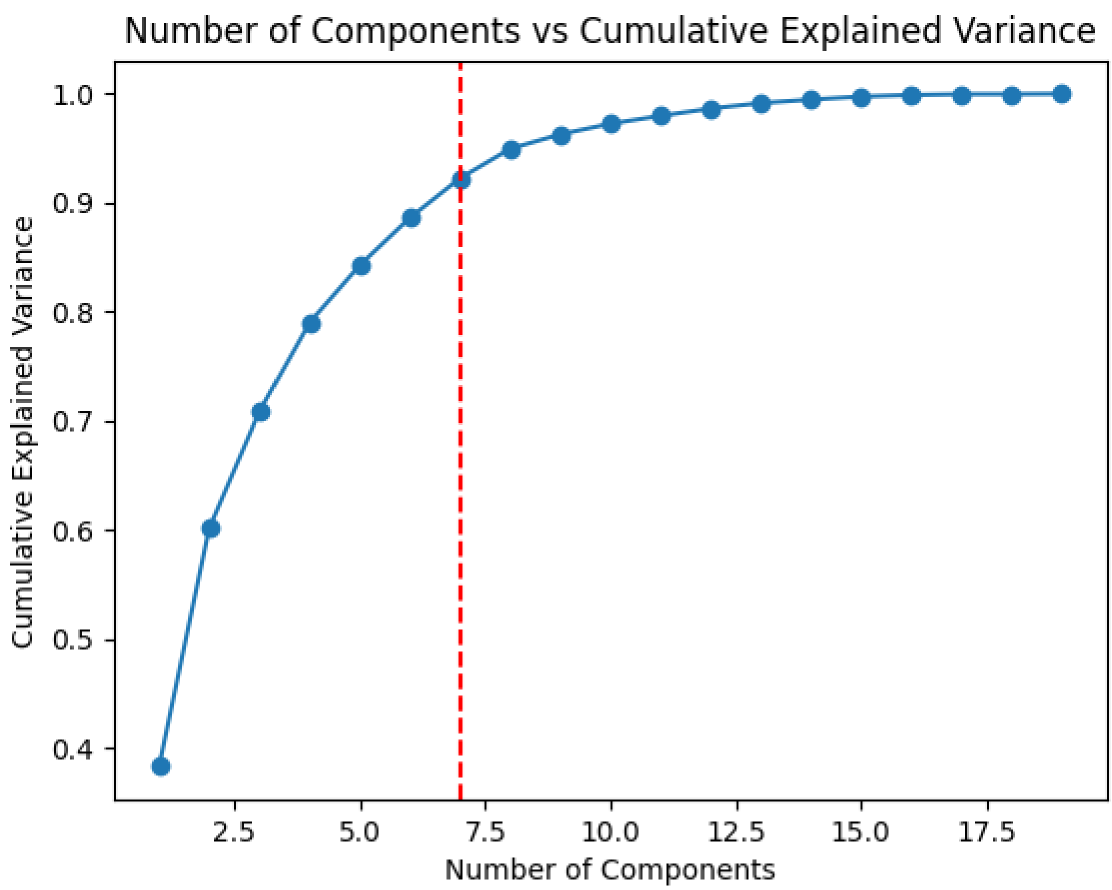
- Calculated the covariance matrix `covariance_matrix` using `np.cov(X_standardized, rowvar=False)`.
- The covariance matrix provides insights into the relationships between different features by quantifying their joint variability.
- Obtained `eigenvalues` and `eigenvectors` using `eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)`.
- Eigenvalues represent the amount of variance captured by each principal component and Eigenvectors indicate the direction in which the data varies the most.
- Sorted `eigenvalues` and corresponding `eigenvectors` in descending order.
- The eigenvalues represent the variance explained by each principal component, and sorting helps prioritize the components with higher variance.
- Eigenvalues play a crucial role in PCA, as they quantify the amount of information (variance) retained in each principal component.
- Eigenvectors provide the direction of maximum variance, aiding in the interpretation of principal components.
- By examining these values, we gain insights into the intrinsic structure of the data and can determine the optimal number of principal components to retain for dimensionality reduction.

c. Explained Variance and Cumulative Explained Variance

- Calculated Explained variance for each component.
- Explained variance represents the proportion of the total variance in the dataset that is captured by each individual principal component. It serves as a measure of how much information each component retains from the original data.
- Computed cumulative explained variance by summing up the explained variance values across all components.

- Cumulative explained variance provides insights into the total information retained as we consider an increasing number of principal components.
- Useful for determining the minimum number of components required to retain a significant amount of information.
- Determined the number of components explaining at least 90% of the variance.
- These metrics guide the decision-making process in selecting an appropriate number of principal components.

d. Number of Components vs Cumulative Explained Variance



- Plotted the relationship between the number of components and explained variance.
- Identified the number of components for at least 90% variance (which is determined to be the stable number of components).

Model Training

Train-Test Split

- Set a random seed and split the dataset into training and testing sets.
- Fraction of data used for training: `train_fraction = 0.8`. Train and Test are 80/20 split.
- The train-test split is a critical step in model development, supporting the evaluation of model performance on unseen data. It helps validate the model's generalization capabilities, guards against overfitting, and allows for reproducibility by setting a random seed.

Linear Regression Model

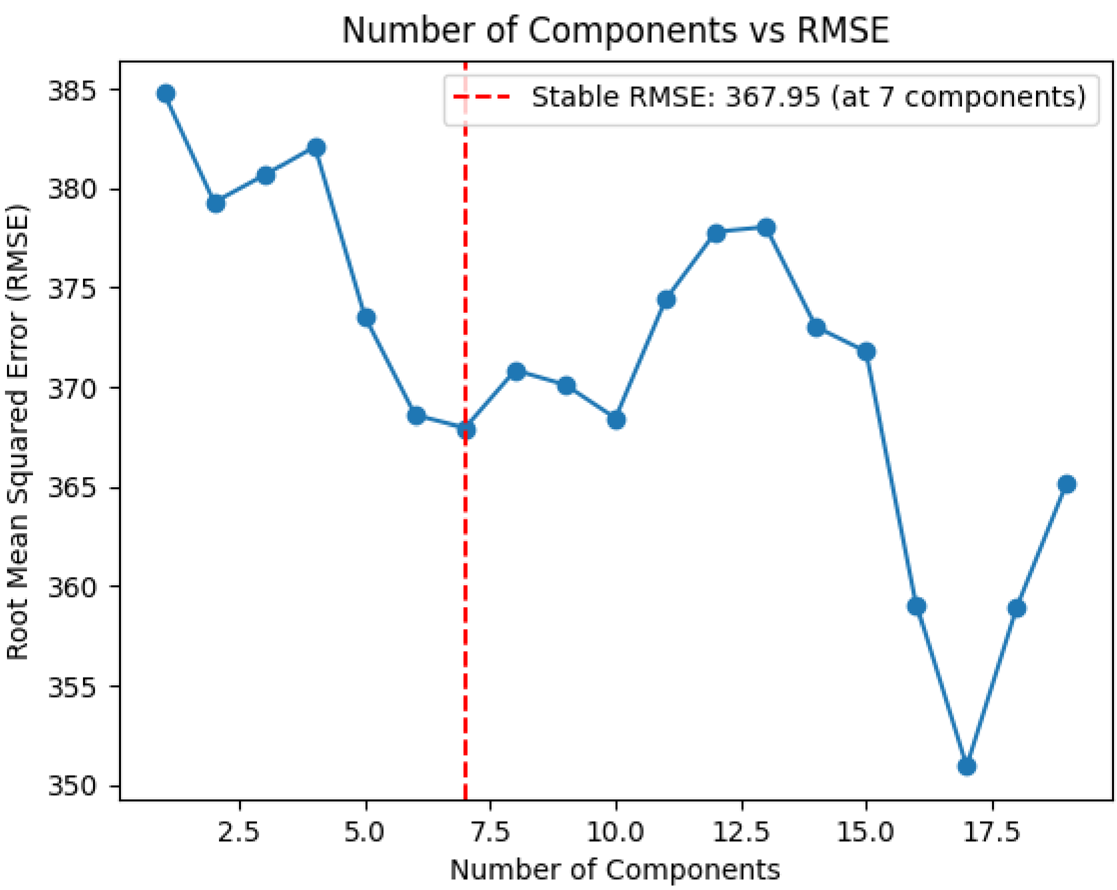
- Defined functions for linear regression model training, prediction, Mean Squared Error (MSE), and Mean Absolute Error (MAE).

Model Evaluation with PCA

- Applied PCA to training and testing sets with varying numbers of components.
- Calculated and stored RMSE values for each component.

Graphical Analysis

Number of Components vs RMSE



- Plotted the RMSE values for different numbers of components.
- Identified the stable RMSE point and marked it with a red dashed line.

Testing the Most Efficient Model

Optimal Number of Components

- Chose the optimal number of components based on the stable RMSE point.

Model Prediction

- Projected data onto the optimal number of components.
- Fitted linear regression using gradient descent.
- Made predictions for a specific data point.

- Displayed the predicted y value: `169.08245822253235`.

Conclusion and Analysis

Interpretation of the Graph

- Analyzed the number of components vs RMSE graph to understand the trade-off between model complexity and accuracy.
- Identified the optimal number of components marked by the stable RMSE point.

Significance of Selecting an Appropriate Number of Components

- Emphasized the importance of finding the right balance between model simplicity and predictive accuracy.
- Discussed the significance of avoiding underfitting and overfitting.

Analysis of the Predicted Value (y_pred)

- Highlighted the importance of analyzing the predicted value in the context of the specific application.

Accuracy Assessment

- Calculated Mean Absolute Error (MAE) for a comprehensive evaluation.
- Displayed the MAE value: `236.30859972742817`.