Project Report: League of Legends Champion Analysis and Role Prediction

Author: Caner AKCASU, (sd1)
Date: June 4, 2025

Abstract: This report details the process of analyzing a dataset of League of Legends champions with the primary objective of predicting a champion's role based on their attributes. The project encompasses data loading and initial exploration, extensive exploratory data analysis (EDA) covering numerical and categorical feature distributions and correlations, comprehensive data cleaning and preprocessing, model training using a RandomForestClassifier with hyperparameter optimization via GridSearchCV, and finally, model evaluation using standard classification metrics. All visualizations generated during the analysis are systematically saved for review.

---

1. Introduction

The project aims to leverage machine learning techniques to analyze League of Legends champion statistics. The core goal is to build a predictive model capable of classifying champions into their predefined roles based on various in-game attributes. This involves a systematic workflow from data ingestion to model deployment and evaluation. The script is designed to perform these steps, identifying 'Role' as the target variable for a classification task.

---

2. Data Loading and Initial Exploration

- **Data Source:** The analysis utilizes a CSV file named champions.csv. This dataset was obtained from Kaggle and is available at: https://www.kaggle.com/datasets/cutedango/league-of-legends-champions. For this project, the file was loaded from the local path specified in the script.
- **Initial Checks:** Upon successful loading, the script performs several initial exploratory steps:
    - Displays the first five rows of the dataset (df.head()).
    - Prints dataset information (df.info()), including column data types and the presence of missing values.
    - Generates a statistical summary of numerical features (df.describe()).
    - Counts and displays the number of missing values for each column (df.isnull().sum()).

- **Target Variable:** The script explicitly defines 'Role' as the target column and 'classification' as the problem type.

---

3. Exploratory Data Analysis (EDA)

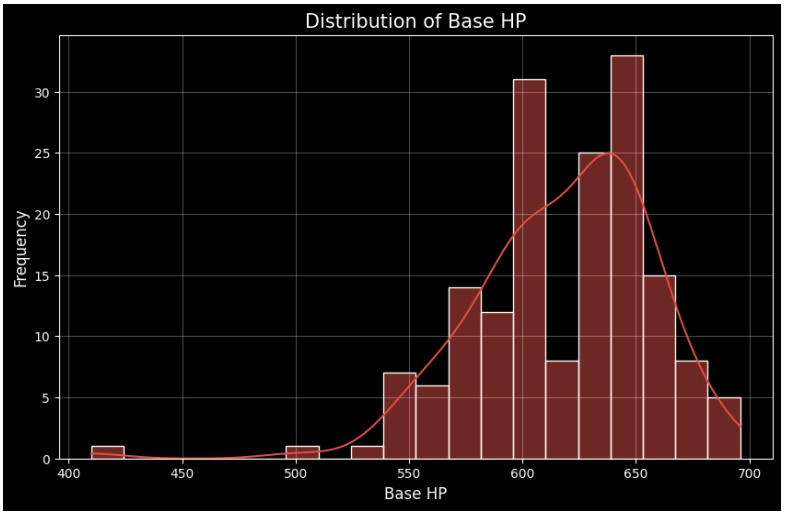
The EDA phase focuses on understanding the characteristics of the dataset's features and their relationships.

3.1. Numerical Feature Analysis

Histograms with Kernel Density Estimates (KDE) were generated for various numerical features to visualize their distributions. A correlation matrix was also generated to understand linear relationships between these features.

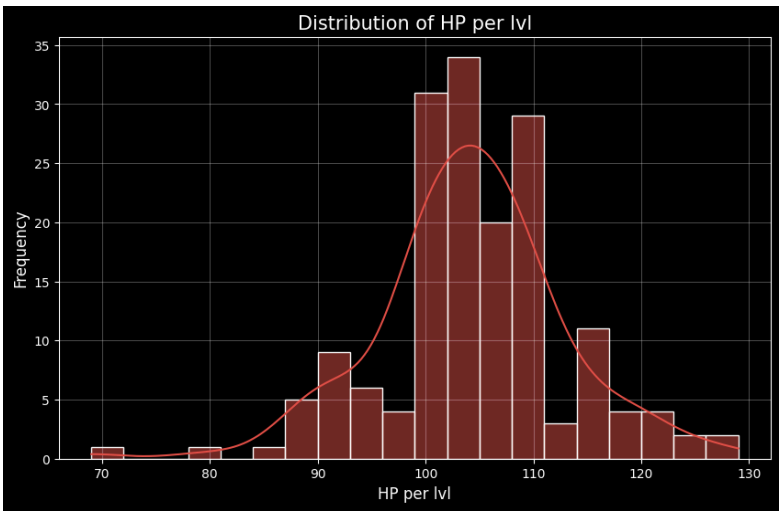
3.1.1. Histograms of Numerical Features

- **Distribution of Base HP**
    - **Image:**



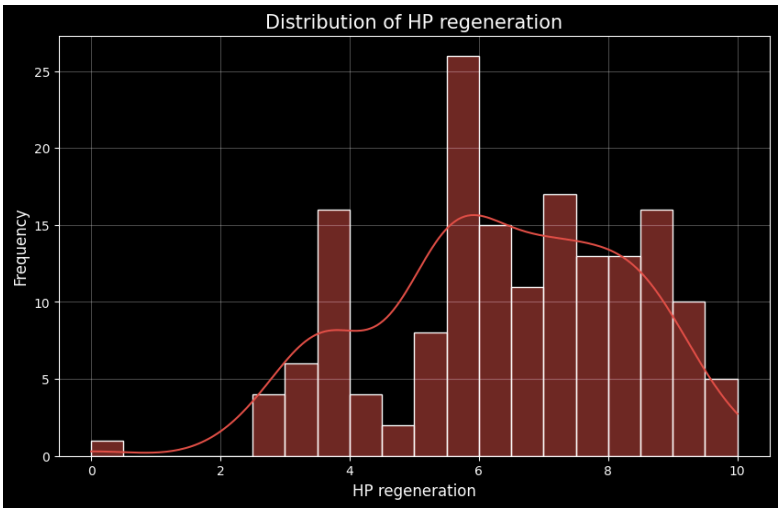
    - **Description:** This histogram illustrates the distribution of champions' base health points (HP) at level 1.
    - **Analysis:** Most champions have base HP values ranging from approximately 550 to 670, with significant clusters around 600-620 and 630-650. This suggests different starting durabilities, a key aspect distinguishing roles like tanks (higher HP) from squishier damage dealers.

- **Distribution of HP per lvl**
    - **Image:**



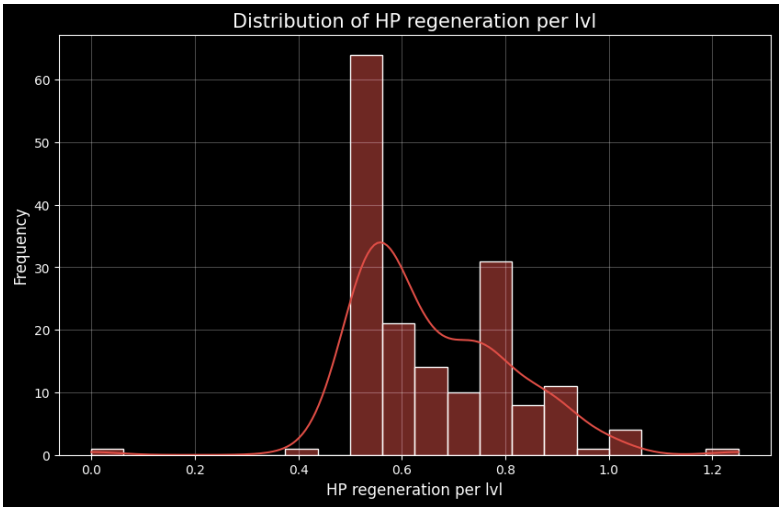
- **Description:** This histogram shows the amount of health points champions gain per level.
- **Analysis:** Most champions gain between 95 and 115 HP per level, with the highest frequencies around 100-105 HP and 105-110 HP. This health scaling is crucial for survivability and varies by intended role.

- **Distribution of HP regeneration**
  - **Image:**

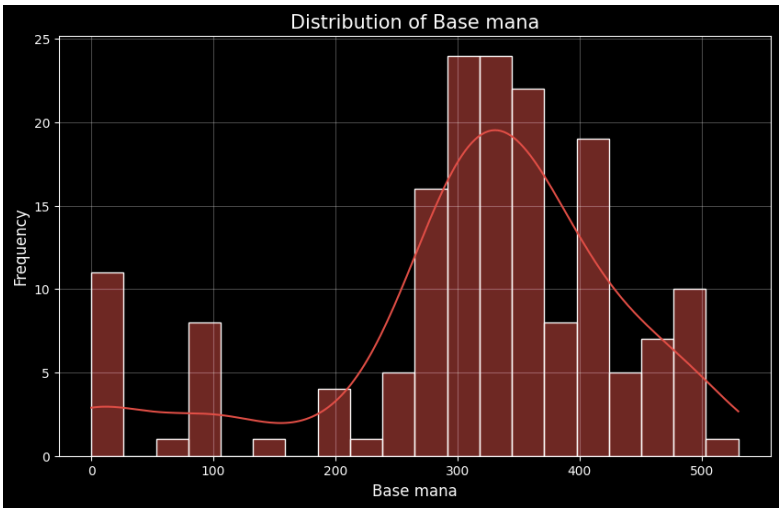


- **Description:** This histogram displays the distribution of base health regeneration per 5 seconds for champions.
- **Analysis:** Base HP regeneration values are spread out, with common values ranging from 3 to 9 HP per 5 seconds. There are noticeable peaks around 3-4, 6, and 8 HP regeneration. Roles expecting to take frequent damage might have higher base regeneration.

- **Distribution of HP regeneration per lvl**
  - **Image:**



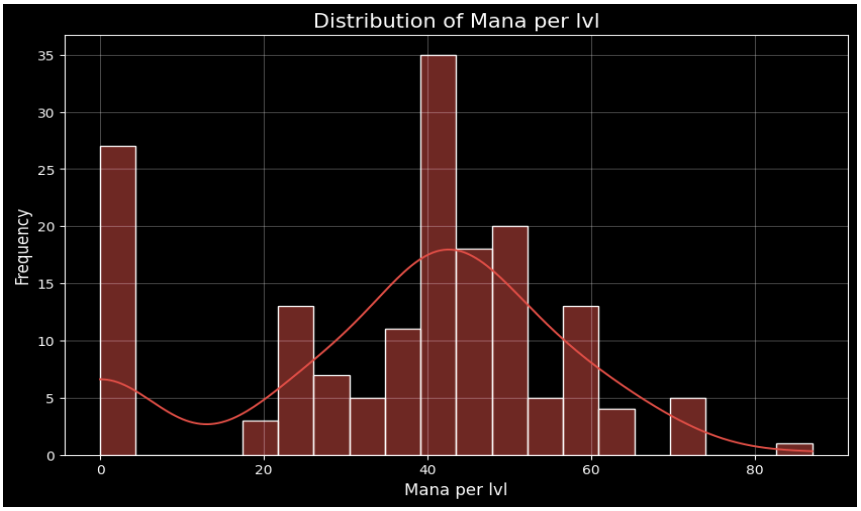
  - **Description:** This graph shows the additional HP regeneration per 5 seconds that champions gain per level.
  - **Analysis:** The distribution is concentrated at lower values, with a prominent peak around 0.5 to 0.6. Another smaller peak is visible around 0.75-0.8.
- **Distribution of Base mana**
  - **Image:**



- **Description:** This histogram shows the frequency distribution of champions' base mana pools at level 1.
- **Analysis:** There's a notable group of champions with 0 mana (manaless champions). For mana-using champions, values are spread out, with common clusters around 280-350 and 400-450 mana. This clearly distinguishes mana-reliant casters.

- **Distribution of Mana per lvl**

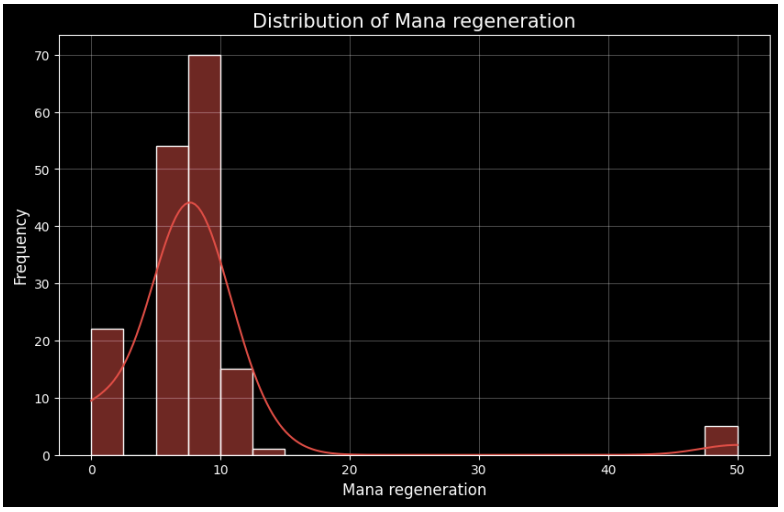
  - **Image:**



  - **Description:** This histogram displays the amount of mana champions gain per level.
  - **Analysis:** A significant number of champions gain 0 mana per level (manaless champions). For mana-using champions, the gain per level commonly ranges from 20 to 70, with a notable peak around 40-50 mana per level.

- **Distribution of Mana regeneration**

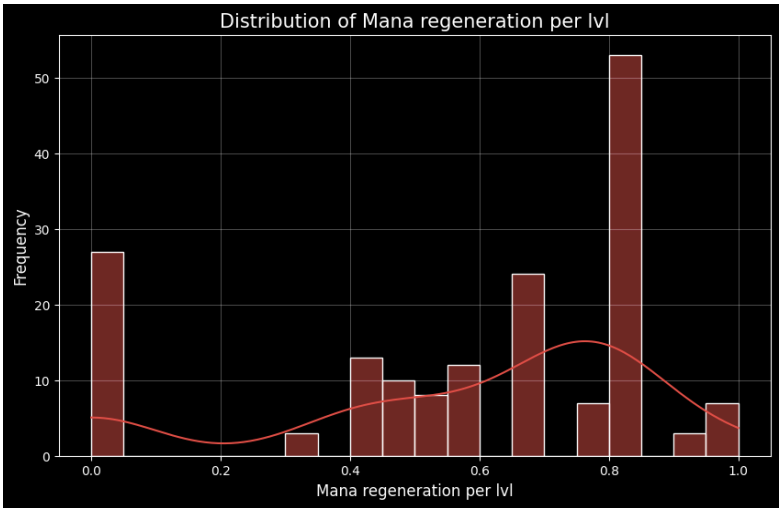
  - **Image:**



- **Description:** This plot shows the distribution of base mana regeneration per 5 seconds.
- **Analysis:** Many champions have base mana regeneration values between 6 and 10, with a strong peak around 8-10. Some champions have 0 mana regeneration (likely manaless ones or those with unique resource mechanics). There's also a very small group with exceptionally high mana regeneration (around 50), which could be an outlier or a very specific champion type.

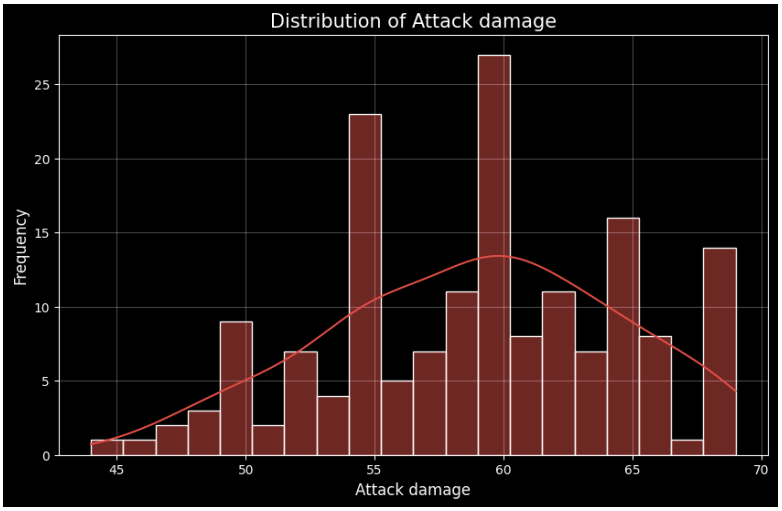
- **Distribution of Mana regeneration per lvl**

  - **Image:**



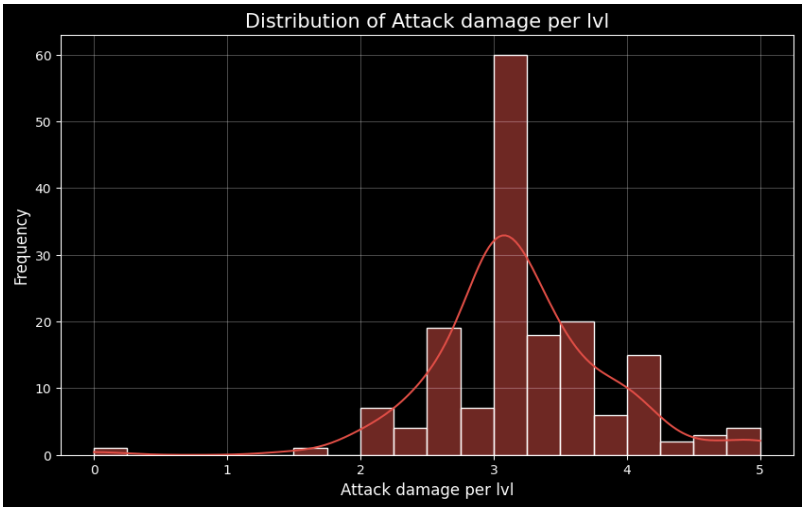
  - **Description:** This histogram illustrates the additional mana regeneration per 5 seconds champions gain per level.
  - **Analysis:** A large group of champions gains 0 mana regeneration per level. For others, the gain is typically between 0.4 and 1.0, with the most common value being around 0.8.

- **Distribution of Attack damage**
    - **Image:**



    - **Description:** This histogram illustrates the distribution of base attack damage values for champions at level 1.
    - **Analysis:** Base attack damage is spread from approximately 45 to 70. The distribution has multiple peaks, suggesting natural groupings (e.g., mages/supports with lower base AD, fighters/marksmen with higher).

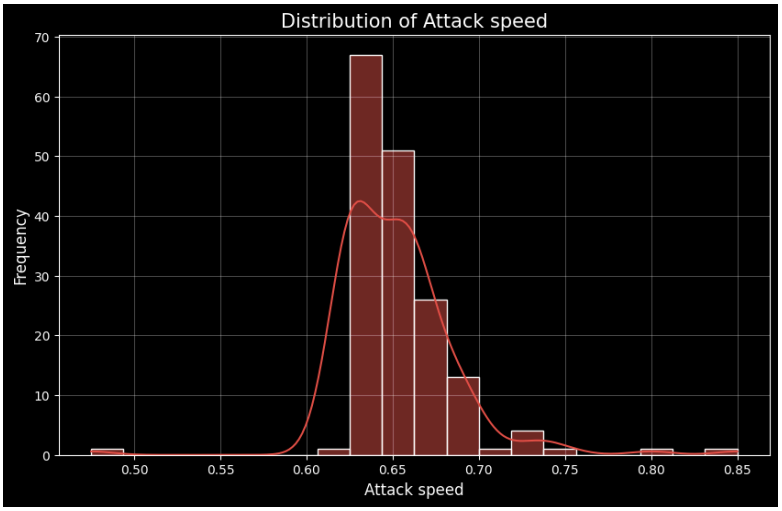
- **Distribution of Attack damage per lvl**
    - **Image:**



- **Description:** This graph shows the frequency distribution of how much additional attack damage champions gain per level.
- **Analysis:** Most champions gain between 2.5 and 4 attack damage per level, peaking around 3-3.5. This scaling is critical for damage output progression.

- **Distribution of Attack speed**

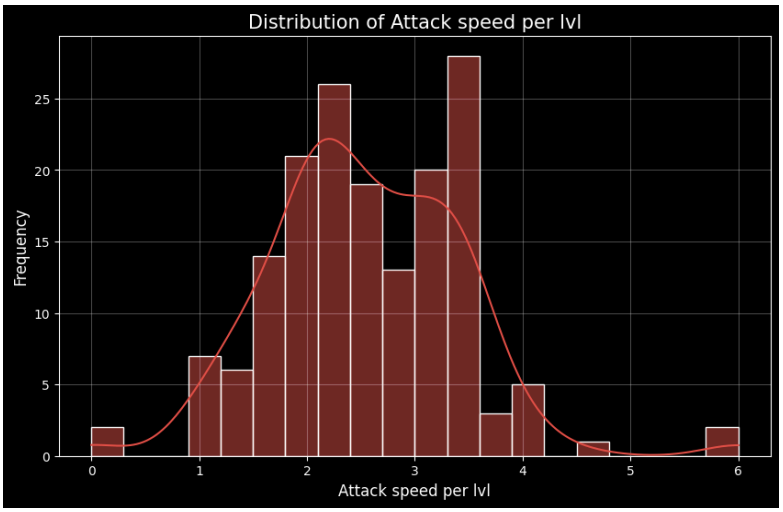
  - **Image:**



  - **Description:** This histogram displays the distribution of champions' base attack speed values.
  - **Analysis:** The majority of champions have a base attack speed between approximately 0.62 and 0.69, with common values clustering around 0.625 and 0.65.

- **Distribution of Attack speed per lvl**

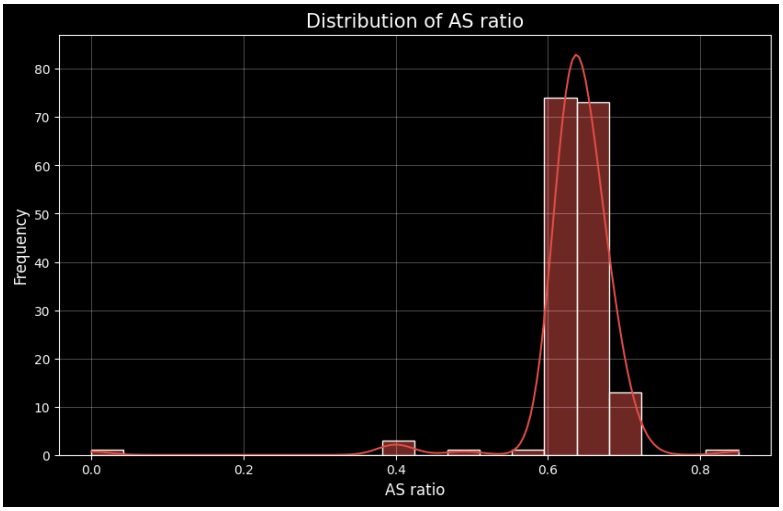
  - **Image:**



- **Description:** This plot shows the frequency distribution of the percentage increase in attack speed champions gain per level.
- **Analysis:** Attack speed gain per level varies, with common values falling between 1.5% and 4%. There are distinct peaks around 2-2.5% and another around 3.5%.

- **Distribution of AS ratio (Attack Speed Ratio)**

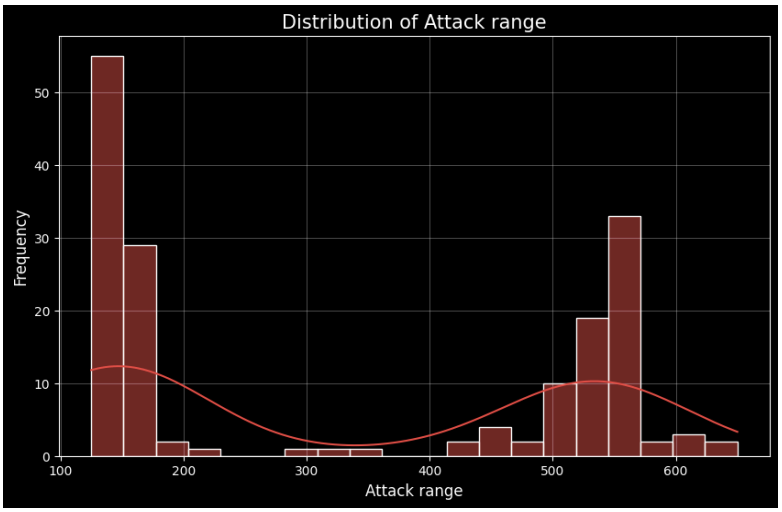
  - **Image:**



  - **Description:** This plot displays the distribution of champion attack speed ratios.
  - **Analysis:** Most champions have an AS ratio concentrated around 0.6 to 0.7, with a prominent peak around 0.625-0.65. This suggests a common baseline for attack speed scaling.

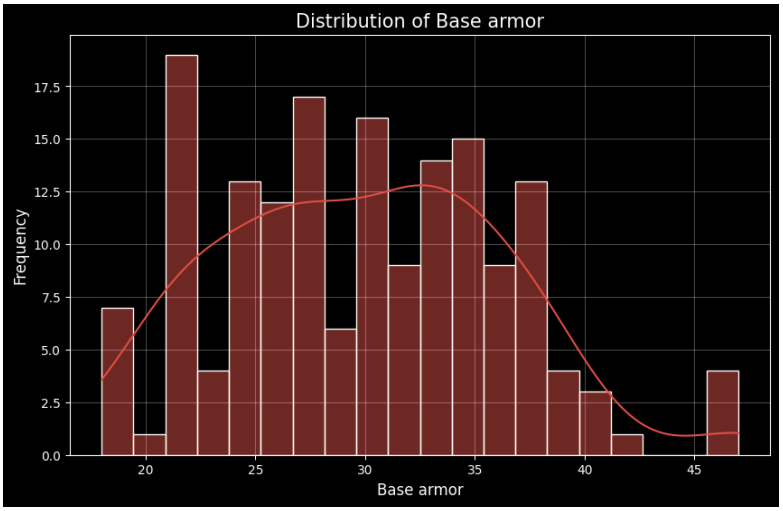
- **Distribution of Attack range**

  - **Image:**



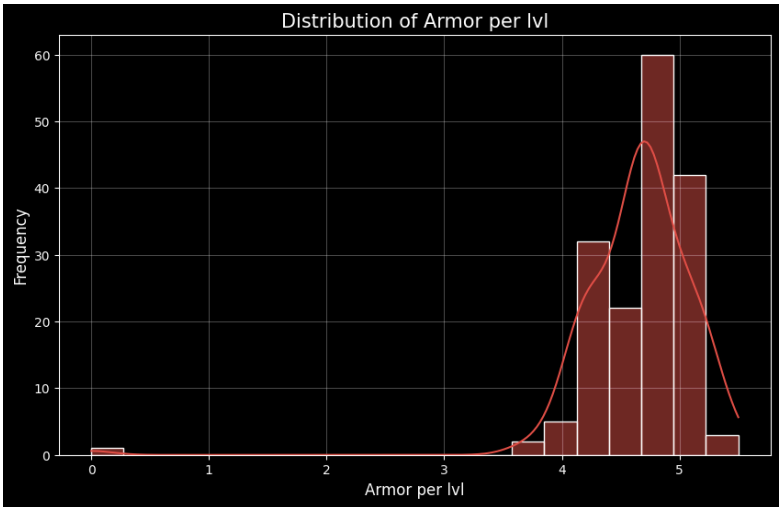
    - **Description:** This histogram shows the distribution of champions' basic attack ranges.
    - **Analysis:** The distribution is clearly bimodal: a large cluster with low attack range (125-175, melee champions) and another significant cluster with higher ranges (500-600, ranged champions). This is a fundamental role differentiator.
- **Distribution of Base armor**

    - **Image:**



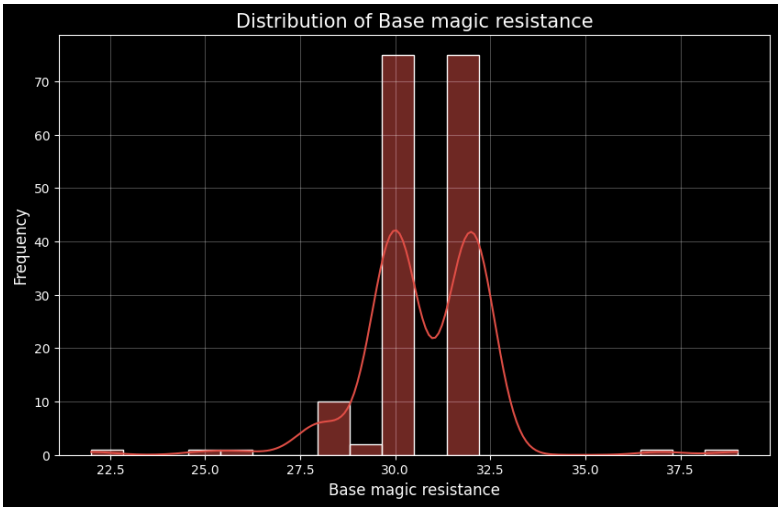
    - **Description:** This histogram presents the distribution of champions' base armor values at level 1.
    - **Analysis:** Base armor values are spread, primarily from around 18 to 40. The distribution appears somewhat multimodal. Tanks and fighters generally start with higher base armor.

- **Distribution of Armor per lvl**
    - **Image:**



    - **Description:** This histogram shows the frequency distribution of armor gained by champions each time they level up.
    - **Analysis:** The majority gain between 4 and 5.5 armor per level, peaking around 5. Tankier roles likely have higher values.

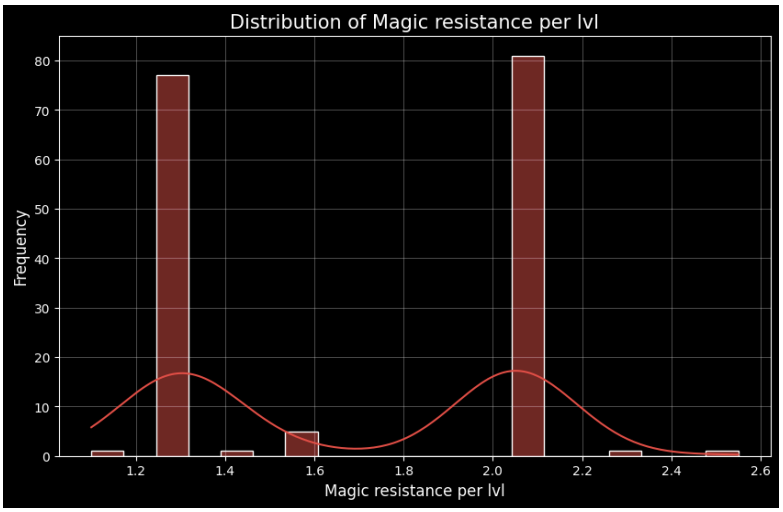
- **Distribution of Base magic resistance**
    - **Image:**



- **Description:** This plot displays the distribution of champions' base magic resistance at level 1.
- **Analysis:** A very large proportion of champions has base magic resistance values tightly clustered around 30 and 32. This suggests base magic resistance is quite standardized.

- **Distribution of Magic resistance per lvl**

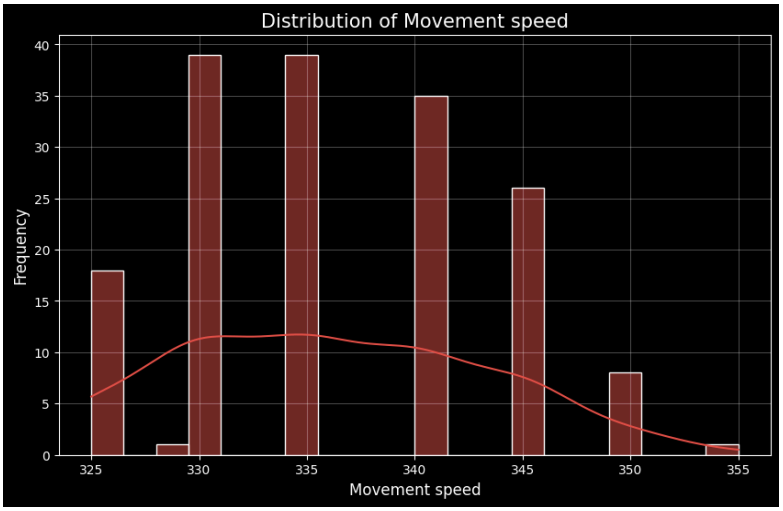
  - **Image:**



  - **Description:** This histogram shows the additional magic resistance champions gain per level.
  - **Analysis:** The distribution is distinctly bimodal, with large groups of champions gaining around 1.25-1.3 MR per level and another large group gaining around 2.05-2.1 MR per level. A smaller number gain around 1.55 MR per level. This indicates different patterns of defensive scaling against magic damage.

- **Distribution of Movement speed**

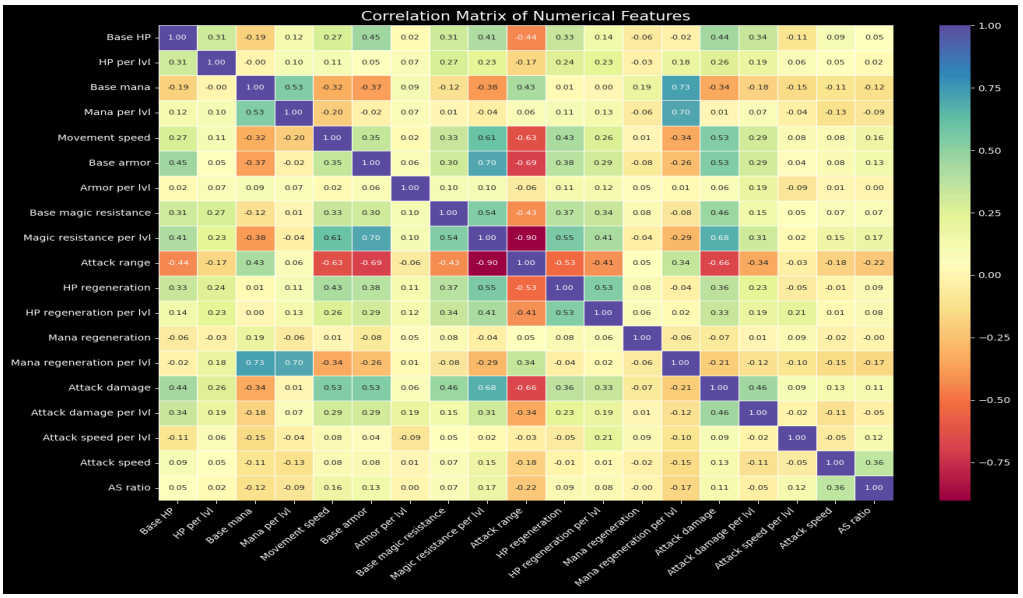
  - **Image:**



- **Description:** This histogram shows the distribution of champions' base movement speed.
- **Analysis:** Movement speed values are clustered at specific points: 325, 330, 335, 340, 345, and 350 are the most common, with 330, 335 and 340 being particularly frequent. This suggests standardized tiers of base mobility.

3.1.2. Correlation Matrix of Numerical Features

- **Image:**



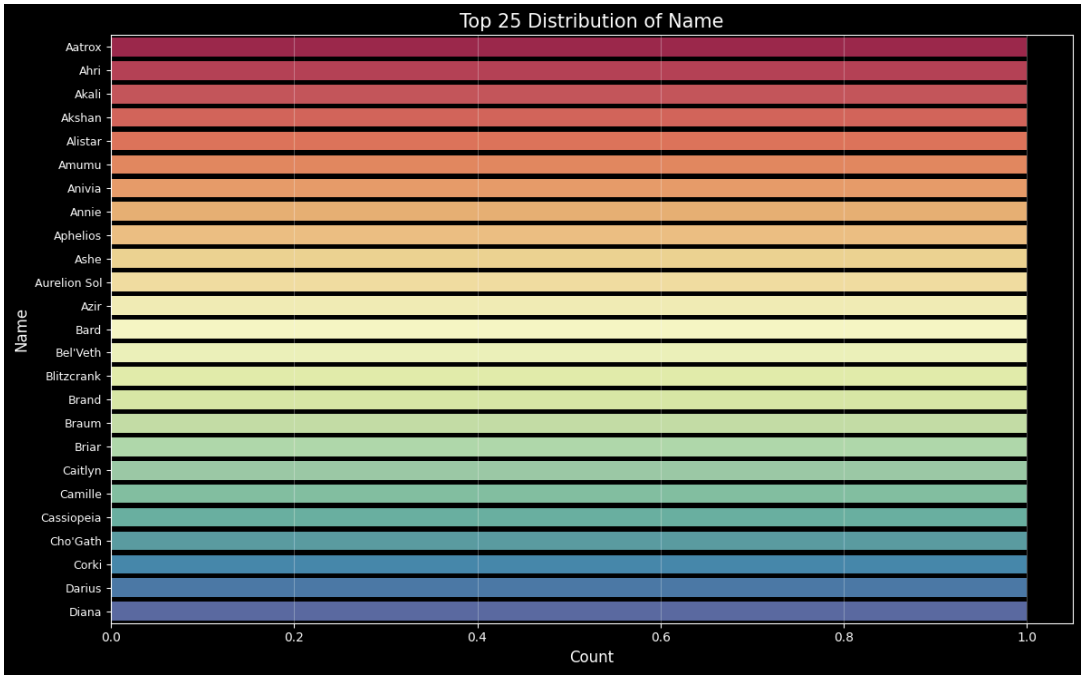
- **Description:** This heatmap visualizes the Pearson correlation coefficients between pairs of numerical features.
- **Analysis:**
  - Strong positive correlations include Magic resistance and Magic resistance per lvl (0.70).
  - Attack range shows strong negative correlations with stats like Base armor (-0.69).

- These correlations highlight interdependencies in champion design (e.g., ranged champions being less tanky by default).

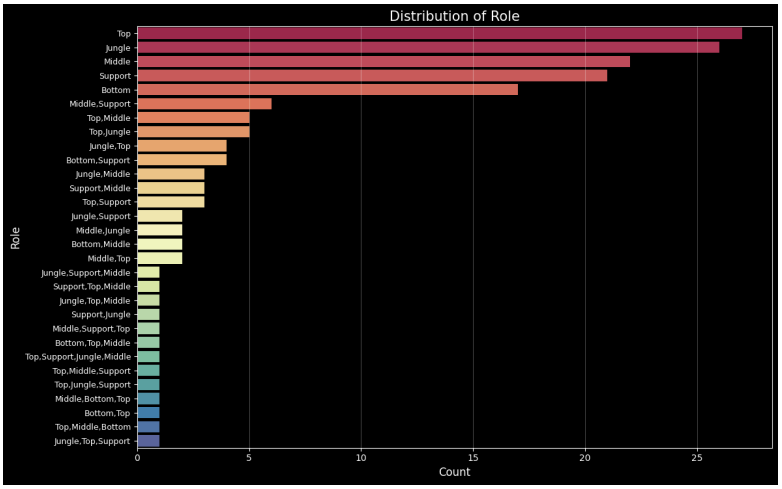
3.2. Categorical Feature Analysis

Count plots were generated for categorical features to display the frequency of each unique category.

- **Distribution of Name**
  - **Image:**

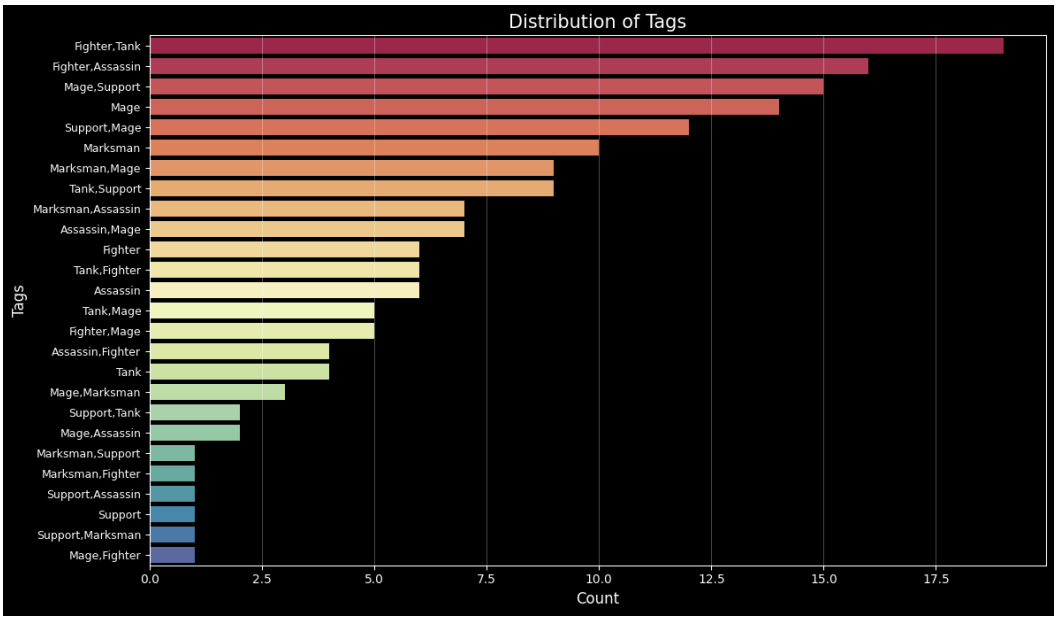


  - **Description:** This plot shows the count for the top 25 champion names in the dataset.
  - **Analysis:** Each champion name in this top 25 list appears exactly once. This confirms that 'Name' is a unique identifier for each champion within this subset (and likely the full dataset, as expected for champion names). For modeling, this feature would typically be dropped or used only for labeling, not as a predictive feature due to its high cardinality and lack of generalizable patterns.
- **Distribution of Role (Target Variable)**
  - **Image:**



- **Description:** This bar chart displays the frequency of each champion role (or combination of roles) as defined in the dataset. This is the target variable for the classification task.
- **Analysis:** The most common roles are singular: "Top", "Jungle", "Middle", "Support", and "Bottom", each having over 20 champions. There are also many combination roles like "Middle,Support", "Top,Middle", etc., but these are less frequent. The least common roles have only one champion. This class imbalance, especially with many rare role combinations, might pose a challenge for the classification model and could necessitate strategies like class weighting or careful stratified sampling.

- **Distribution of Tags**
  - **Image:**

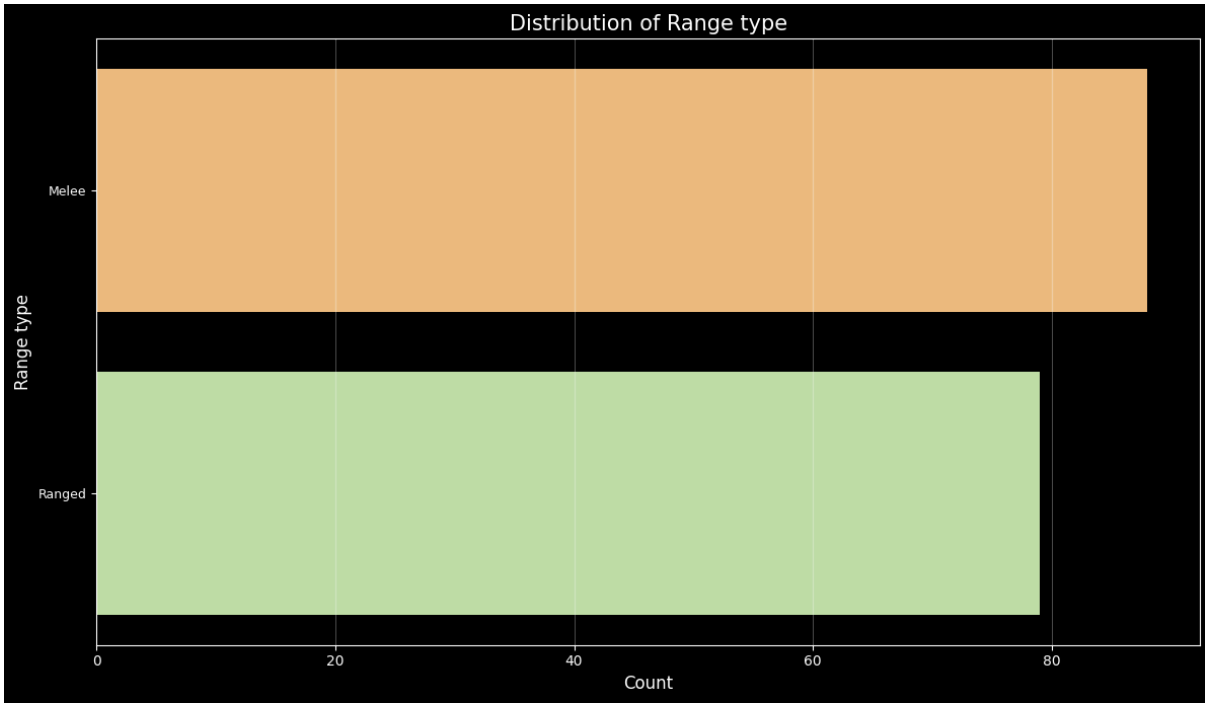


  - **Description:** This plot shows the frequency of different champion tags or tag combinations (e.g., "Fighter,Tank", "Mage").

- **Analysis:** "Fighter,Tank" is the most common tag combination, followed by "Fighter,Assassin" and "Mage,Support". Single tags like "Mage" and "Marksman" are also quite frequent. This feature provides a more granular classification than 'Role' and could be very informative for predicting the primary role, as tags often dictate a champion's playstyle and core strengths.

- **Distribution of Range type**

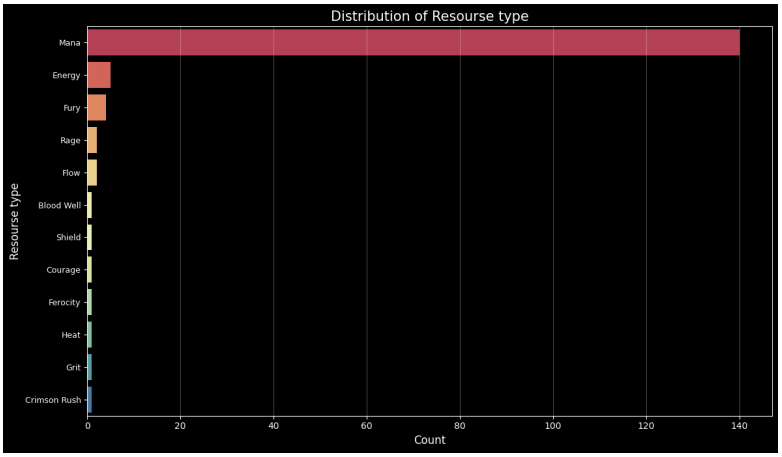
  Image:



  - **Description:** This bar chart shows the count of champions classified as "Melee" versus "Ranged".
  - **Analysis:** The dataset contains a slightly higher number of "Melee" champions (around 88-90) compared to "Ranged" champions (around 78-80). This is a fundamental binary classification that strongly correlates with 'Attack range' and will likely be a very important feature for role prediction.

- **Distribution of Resourse type (Resource Type)**

  - **Image:**



- **Description:** This plot displays the frequency of different resource types champions use for their abilities (e.g., Mana, Energy, Fury).
- **Analysis:** "Mana" is by far the most common resource type, used by over 140 champions. "Energy" is the next most common, but significantly less so (around 5-7 champions). Other resource types like "Fury", "Rage", "Flow", "Blood Well", "Shield", "Courage", "Ferocity", "Heat", "Grit", and "Crimson Rush" are used by very few champions each (typically 1-3). The presence of "None" or "Manaless" in the base mana distribution aligns with these alternative resource types. This feature will be crucial for distinguishing different types of casters and other champions.

---

4. Data Cleaning and Preprocessing

- **Missing Value Imputation:** Numerical features imputed with median, categorical with mode. Target column 'Role' imputed with mode.
- **Feature and Target Preparation:**
  - X (features) and y (target 'Role') defined.
  - Target 'Role' is LabelEncoded.
  - Rare classes (<2 samples) in y removed.
- **Preprocessing Pipelines:**
  - Numerical: SimpleImputer (median) -> StandardScaler.
  - Categorical: SimpleImputer (mode) -> OneHotEncoder.
  - ColumnTransformer applies these.

---

5. Data Splitting

- Data split into 80% training, 20% testing (random_state=42).
- Stratification by y is used.

---

6. Model Training and Optimization

- **Model:** RandomForestClassifier (random_state=42).
- **Hyperparameter Tuning:** GridSearchCV with Pipeline (preprocessor + model).
    - Params: n_estimators ([50, 100]), max_depth ([10, 20, None]), min_samples_leaf ([1, 2, 4]).
    - Scoring: 'accuracy'.

---

7. Model Evaluation

The trained RandomForestClassifier model, optimized via GridSearchCV, was evaluated on the held-out test set to assess its performance in predicting champion roles. Standard classification metrics were computed, and a confusion matrix was generated to provide a detailed view of its predictive accuracy across different roles.

7.1 Classification Report Summary (Suggestion: Add this subsection if you have the report)

(If you have the classification report from your script (with precision, recall, F1-score per class), you would summarize it here or present it as a table. For example:)

"The classification report, presented in Table [Your Table Number, e.g., 7.1], summarizes the precision, recall, and F1-score for each predicted champion role. Overall, the model achieved an accuracy of [Insert Accuracy Score, e.g., XX.X%] on the test set.

(Insert Table of Classification Report here if available)

Table [e.g., 7.1]: Classification Report for RandomForestClassifier on Test Data"

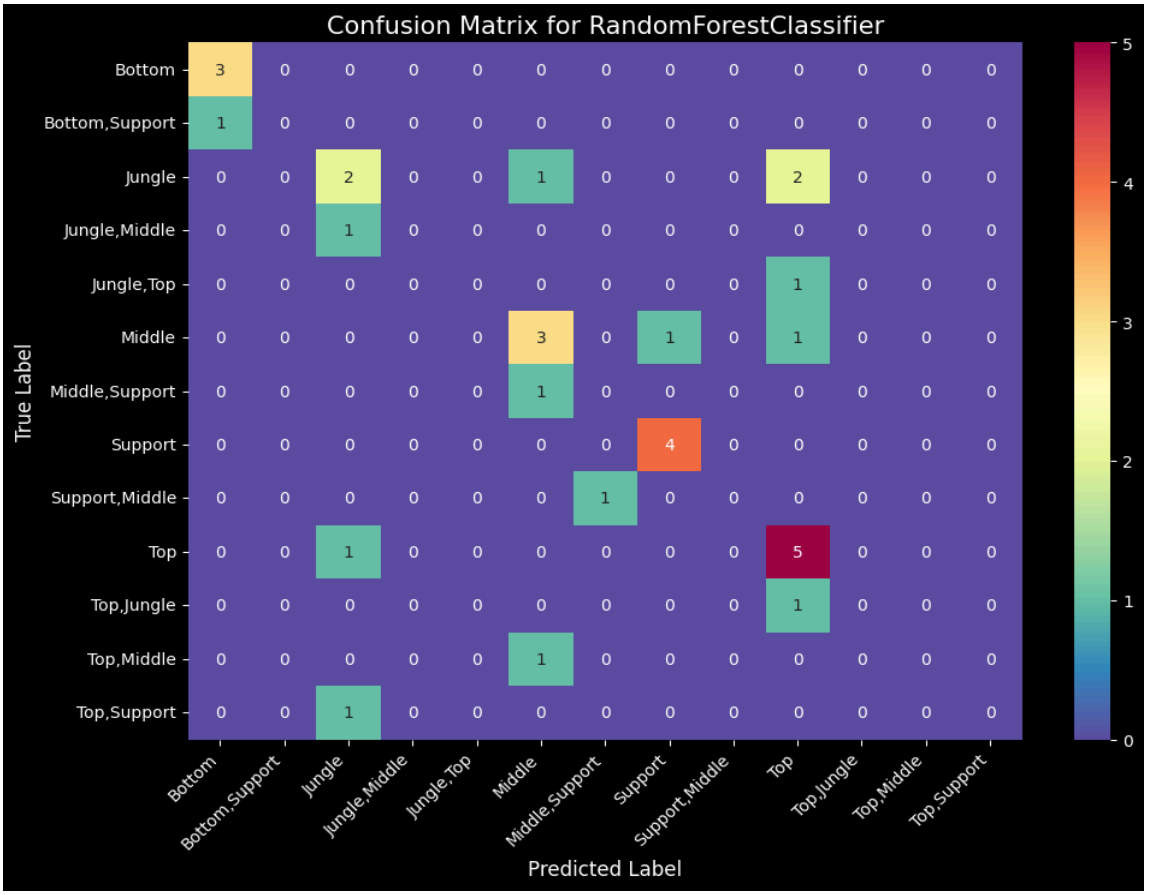
7.2 Confusion Matrix Analysis

"To further understand the model's performance on a per-class basis, a confusion matrix was generated (Figure [Your Figure Number, e.g., 7.1]). This matrix is crucial for identifying which roles the model predicts accurately and where misclassifications commonly occur.

7.2.1 Enhancement in Label Representation for Interpretability

A key aspect of this evaluation is the interpretability of the class labels within the confusion matrix. While the underlying model operates on numerically encoded labels (as initially described in the raw output, e.g., 'true label 0'), the visualization presented in Figure [e.g., 7.1] has been configured to display the original, human-readable text-based role names (e.g., 'Top', 'Jungle,Middle'). This was achieved through post-prediction mapping using the fitted

LabelEncoder instance, ensuring that the model's performance can be directly and more easily interpreted in the context of the actual champion roles.



(Confusion_Matrix_for_RandomForestClassifier.png)

7.2.2 Interpreting the Confusion Matrix

The confusion matrix in Figure [e.g., 7.1] is structured with:

- **Rows** representing the actual (True) champion roles from the test set.
- **Columns** representing the champion roles as predicted by the RandomForestClassifier model.
- **Diagonal cells** indicating the number of champions correctly classified into their respective roles. For example, the cell at the intersection of the 'Top' row and 'Top' column shows the number of champions that are actually 'Top' and were correctly predicted as 'Top'.
- **Off-diagonal cells** showing the instances that were misclassified. For example, a value in the 'Top' row and 'Jungle' column would indicate the number of champions whose true role is 'Top' but were incorrectly predicted as 'Jungle'.

7.2.3 Analysis of Predictive Performance (Based on your image)

The confusion matrix (Figure [e.g., 7.1]) reveals several key aspects of the model's performance:

- **Well-Predicted Roles:** The model demonstrates reasonable success in identifying certain primary roles. For instance, the role 'Support' was correctly predicted for 4 champions, and the role 'Top' was correctly predicted for 5 champions. The diagonal entries for 'Bottom' (3) and 'Middle' (3) also indicate correct classifications for these roles.
- **Common Misclassifications:**
    - The model exhibits some confusion between closely related or overlapping roles. For example, champions with a true role of 'Jungle' were misclassified as 'Jungle,Middle' (1 instance) and 'Top,Jungle' (2 instances).
    - A champion with the true role 'Middle' was misclassified as 'Middle,Support' (1 instance).
    - The model also misclassified one 'Top,Jungle' champion as 'Jungle,Top' (1 instance).
- **Performance on Hybrid Roles:** The dataset contains numerous hybrid roles (e.g., 'Bottom,Support', 'Jungle,Middle', 'Middle,Support'). The matrix shows that while some hybrid roles have correct predictions on the diagonal (e.g., 'Middle,Support' has 1 correct prediction when it was the true label, 'Bottom,Support' also has 1), these roles are also involved in misclassifications, either being mispredicted as another role or other roles being mispredicted as them. This highlights the inherent difficulty in classifying champions that fit into multiple role definitions.
- **Impact of Class Imbalance:** As identified in the EDA (Section 3.2, specifically the 'Distribution of Role' analysis), there is a significant class imbalance, with many roles (especially combined roles) having very few instances. This can affect the model's ability to learn and accurately predict these less frequent roles, which is reflected in lower counts (both correct and incorrect) for many of the rarer role combinations in the matrix. For example, roles like 'Top,Middle,Bottom' or 'Jungle,Top,Support' appear very infrequently and thus have limited representation in the test outcomes.

---

8. Conclusion

The project successfully implemented a pipeline for League of Legends champion role prediction. EDA revealed distinct patterns in champion statistics and clear differentiators like attack range and resource type. A RandomForestClassifier, tuned with GridSearchCV, was trained and evaluated. The confusion matrix provides insights into its classification performance on various roles. The class imbalance in the 'Role' feature, identified during EDA, is a key consideration for interpreting model performance and for future improvements.