Introduction and Problem Statement:

In this project, we focus on analyzing the performance data of Starcraft players in ranked games. My goal is to develop a model that can accurately predict a player's rank based on the information provided in the dataset.

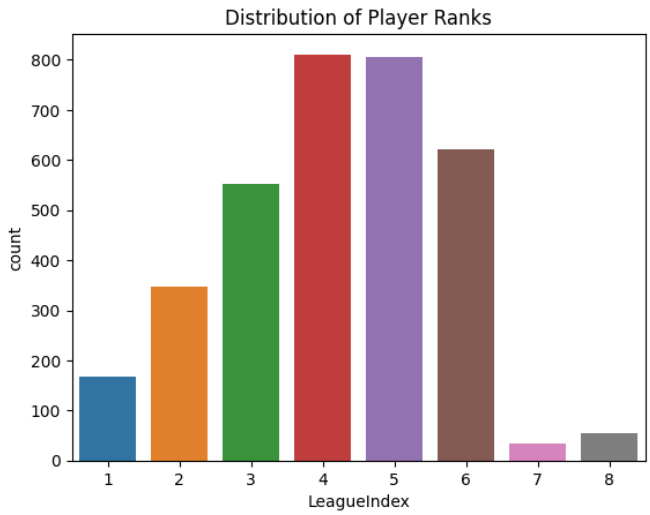
The dataset we have at our disposal contains a wide range of features related to player demographics, gaming behavior, and in-game performance. These include variables such as age, hours spent playing per week, action per minute (APM), unique hotkeys used, and many others. By leveraging this rich set of information, we aim to build a predictive model that can accurately classify a player into the appropriate rank or league.

The ability to predict a player's rank has several practical implications. It can help gaming platforms and esports organizations in player matchmaking, skill-based ranking systems, and identifying high-potential players for professional leagues. Additionally, understanding the key factors that contribute to a player's rank can provide valuable insights into the game's mechanics and strategies, allowing for continuous improvement of gameplay experience.

By addressing this challenge, we aim to unlock the potential of player performance analysis and contribute to the understanding of factors that influence player rankings in online competitive gaming environments.

Data Description and EDA

The dataset consists of 3395 rows with 20 features. Our target variable is ‘LeagueIndex’ which refers to the rank of the player. Here is the graph showing the distribution of player ranks:



Current dataset is imbalanced, and certain rank categories (especially Rank7 and Rank8) have fewer samples.

Missing Values

After conducting a checking test for the missing values (non-integer or non-float values), I found that there are three variables which contain missing values:

```
Variable 'Age' contains 55 non-integer or non-float values.
Variable 'HoursPerWeek' contains 56 non-integer or non-float values.
Variable 'TotalHours' contains 57 non-integer or non-float values.
```

More specifically, all the values for the Age feature were missing for the LeagueIndex = 8. The missing value issue may cause issues in our prediction model since the highest rank (8) is possibly what the stakeholders want to pay attention to.

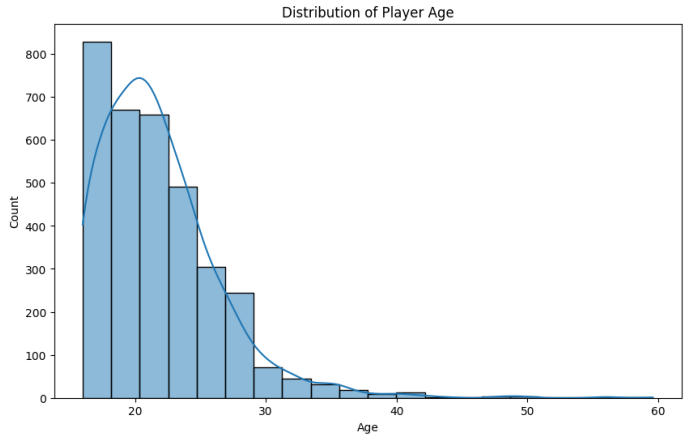
Since the data distributions for different LeagueIndex (Rank) are different, simply using imputation techniques such as removing the missing values or filling in with ‘mean’ or ‘median’ values of the rest of the data are not appropriate or favorable here. The approach that I used is by using 3 separate random forest models to predict the missing values in the three columns. The approach has the advantage of preservation of information, reduced bias, improving accuracy, and generalizability. The imputed values can be influenced by various factors, leading to more realistic estimations.

This is what the imputed data from random forest looks like:

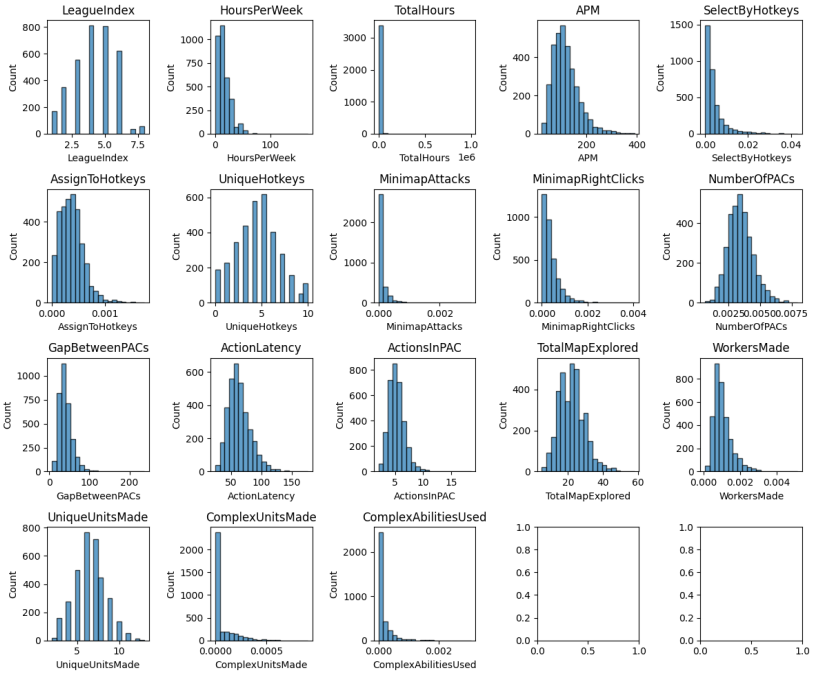
3346	10018	8	24.00	23.50	1136.26	211.0722
3347	10021	8	24.00	24.30	1002.99	189.5778
3348	10022	8	30.00	29.40	1151.64	210.5088
3349	10023	8	32.00	31.36	4373.46	248.0118
3350	10024	8	32.00	33.14	61020.00	299.2290
3351	10025	8	28.00	27.72	1284.17	179.9982

2) Data Distribution (Skewed Data)

Here is the graph showing the distribution of Player Age:



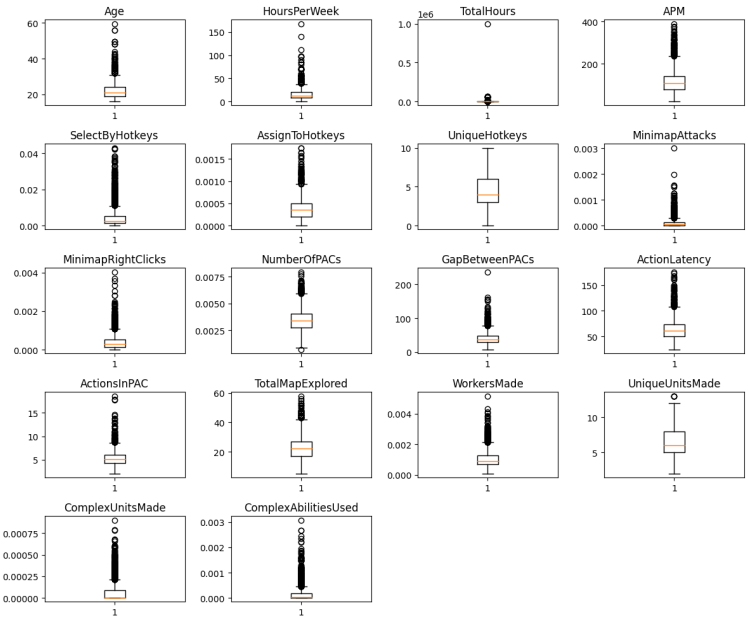
From the graph we can see that the ‘Age’ is skewed and needs to normalize before the model training. Let's see if other features also need normalizing before model training.



The Histogram Result showed that The following features need normalization before model training: 'Age', 'HoursPerWeek','SelectByHotkeys', 'AssignToHotkeys', 'UniqueHotkeys', 'MinimapAttacks', 'NumberOfPACs', 'GapBetweenPACs', 'ActionLatency', 'TotalMapExplored', 'WorkersMade', 'UniqueUnitsMade', 'ComplexUnitsMade'.

3) Outliers

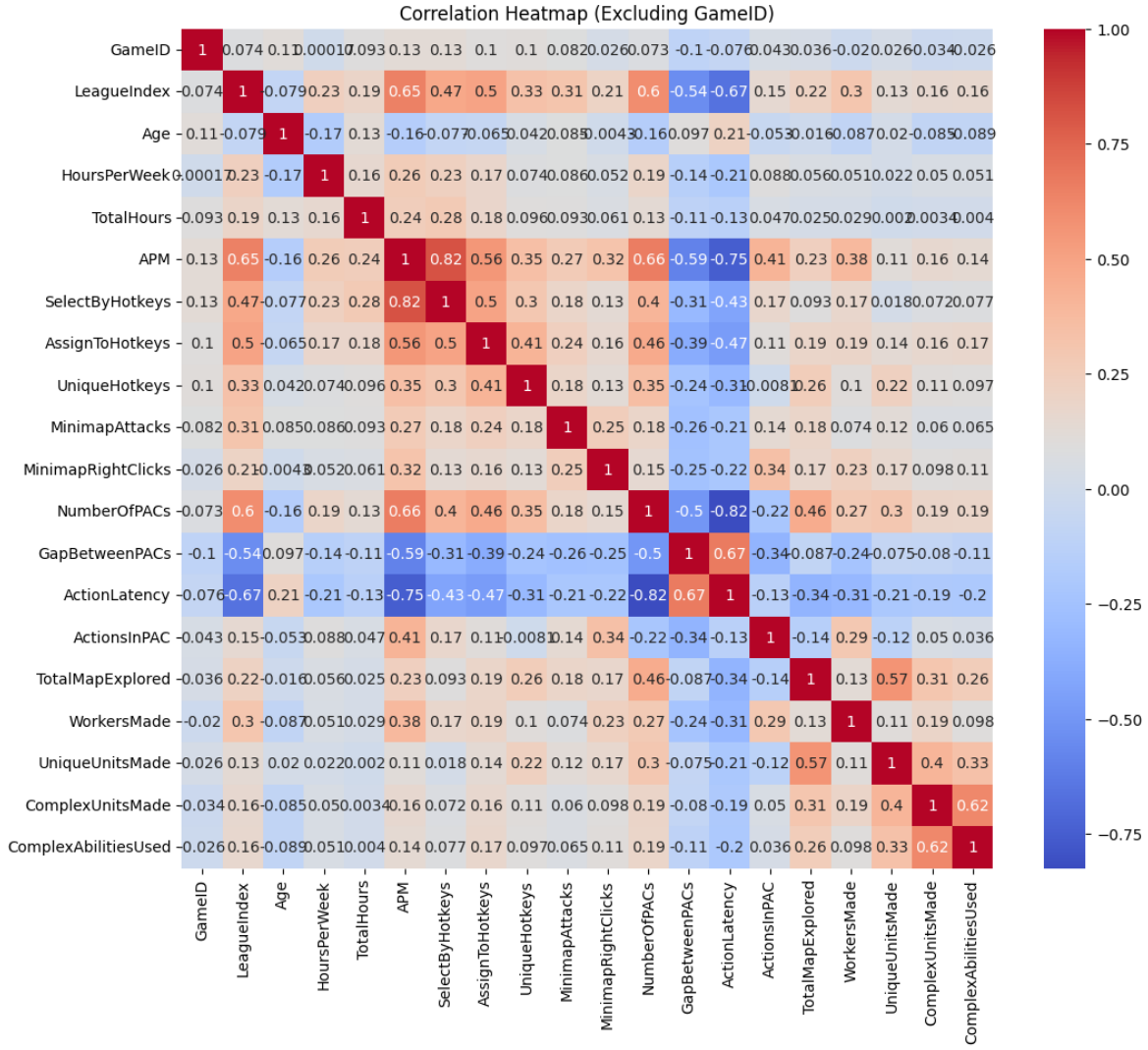
Here is the graph showing the outliers for each feature:



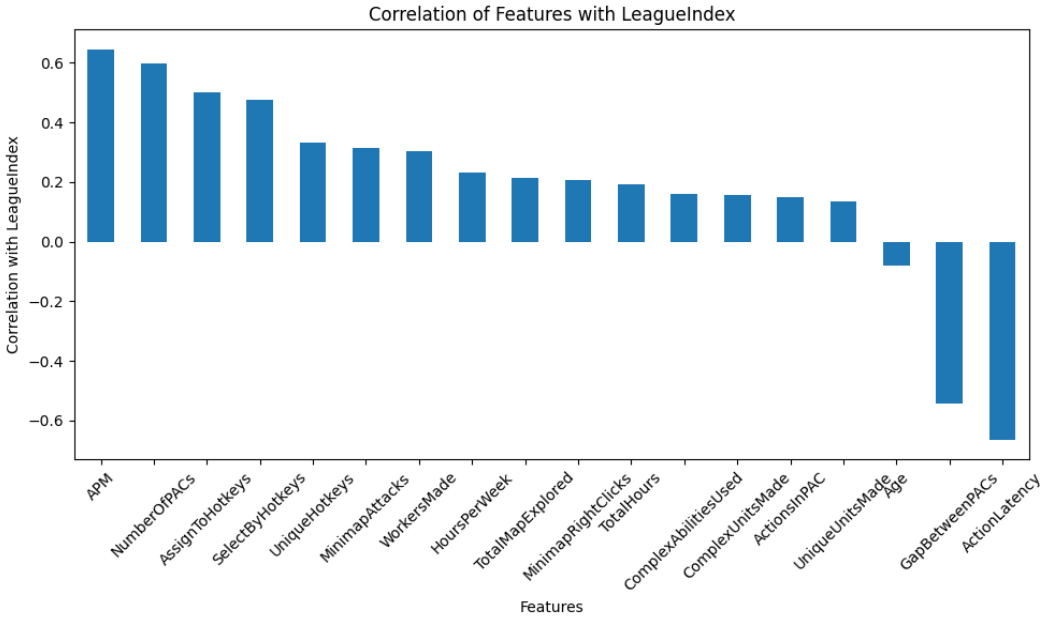
In order to remove the outliers that are too far away from the mean, I used z-score and applied the corresponding threshold (set to 5) to get rid of the outliers. The z-score is a measure of how many standard deviations a data point is from the mean. Using the z-scores, I identify the outliers by filtering the DataFrame for data points with z-scores above the threshold.

Feature Engineering

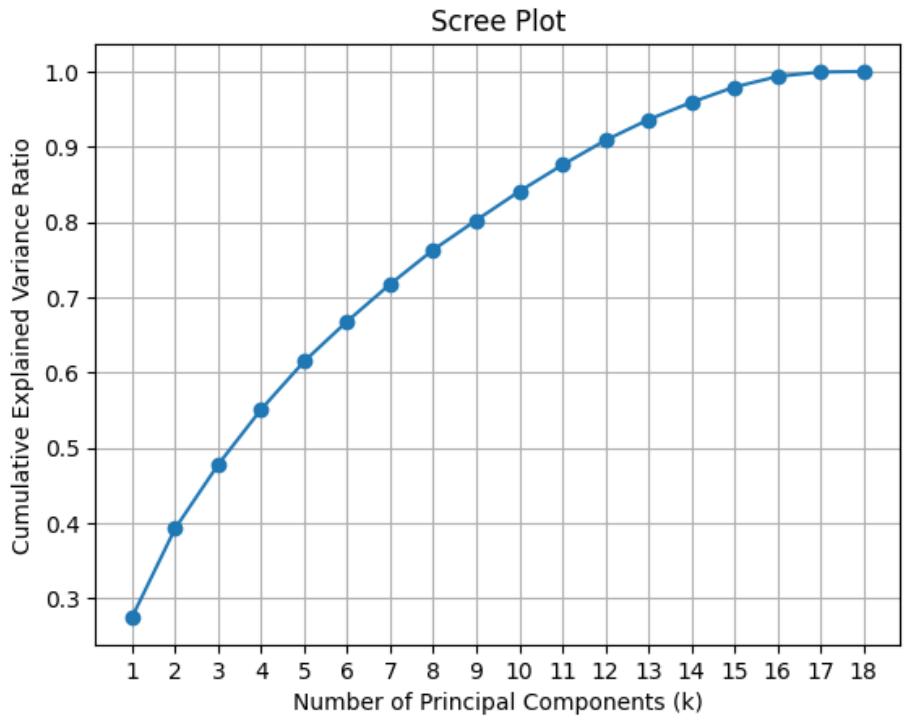
In order to select the features that contribute most to the prediction task and help reduce dimensionality and improve model interpretability, I firstly compute and create the heatmap of feature correlation to see which features are the most correlated with the target feature (LeagueIndex). Here is the heatmap:



Then create a graph showing the correlation of features with LeagueIndex:



From the plot, we can see that APM, NumberOfPACs, AssignToHotkeys, and SelectByHotkeys are the 4 most positively correlated features with LeagueIndex, while ActionLatency and GapBetweenPACs are the top 2 most negatively correlated features with LeagueIndex. To further explore the features needed for the model, I conducted a PCA for feature selection. Here is the output:



From the Plot above, we can see that K = 12 when Cumulative explained variance ratio reaches 0.9. This means that the first 12 principal components contribute significantly to explaining the variability in the dataset. From my analysis, removing additional features beyond the 12 principal components may not be necessary since they contribute less to the explained variance and we'd better keep most of the features.

Modeling Approach

Logistic Regression
A widely used classification algorithm that models the relationship between the independent variables and the binary or multi-class outcome. I chose logistic regression as it is a simple and interpretable model, suitable for binary or multi-class classification tasks. It allows us to examine the impact of individual features on the probability of belonging to a particular rank category.

Random Forest
An ensemble learning method that combines multiple decision trees to make predictions. It works by creating a forest of randomly sampled trees and aggregating their predictions.
I selected random forest due to its ability to handle complex relationships, capture nonlinear patterns, and provide feature importance rankings.

XGBoost
A gradient boosting framework that utilizes an ensemble of weak prediction models, such as decision trees, to create a strong predictive model. I opted for XGBoost because it has shown exceptional performance in many machine learning competitions and is highly effective in handling complex datasets. XGBoost provides mechanisms for handling missing values, regularization, and controlling model complexity.

Neural Network
A powerful deep learning model inspired by the structure of the human brain. It consists of interconnected layers of artificial neurons that learn hierarchical representations from the data. I included a neural network model to take advantage of its ability to capture intricate patterns and relationships in the data.

Parameter Tuning
Employed techniques such as cross-validation and grid search to find the optimal combination of hyperparameters for each model. Cross-validation helps assess the generalization performance of the model, while grid search systematically explores different hyperparameter values.

Evaluation

Evaluation Metrics

Accuracy, precision, recall, and F1 score were used to assess the performance of the models. Accuracy measures the overall correctness of predictions, while precision focuses on the proportion of correctly predicted instances for a specific class. Recall measures the proportion of true positive instances identified correctly, and the F1 score provides a balanced measure of precision and recall.

Logistic regression:

	precision	recall	f1-score	support
1	0.31	0.19	0.23	27
2	0.30	0.28	0.29	69
3	0.49	0.32	0.39	114
4	0.36	0.49	0.42	157
5	0.41	0.42	0.41	161
6	0.57	0.57	0.57	115
7	0.00	0.00	0.00	6
8	0.60	1.00	0.75	3
accuracy			0.42	652
macro avg	0.38	0.41	0.38	652
weighted avg	0.42	0.42	0.41	652

Random Forest:

	precision	recall	f1-score	support
1	0.30	0.11	0.16	27
2	0.35	0.35	0.35	69
3	0.36	0.32	0.33	114
4	0.38	0.46	0.42	157
5	0.43	0.45	0.44	161
6	0.56	0.53	0.54	115
7	0.00	0.00	0.00	6
8	1.00	0.67	0.80	3
accuracy			0.42	652
macro avg	0.42	0.36	0.38	652
weighted avg	0.41	0.42	0.41	652

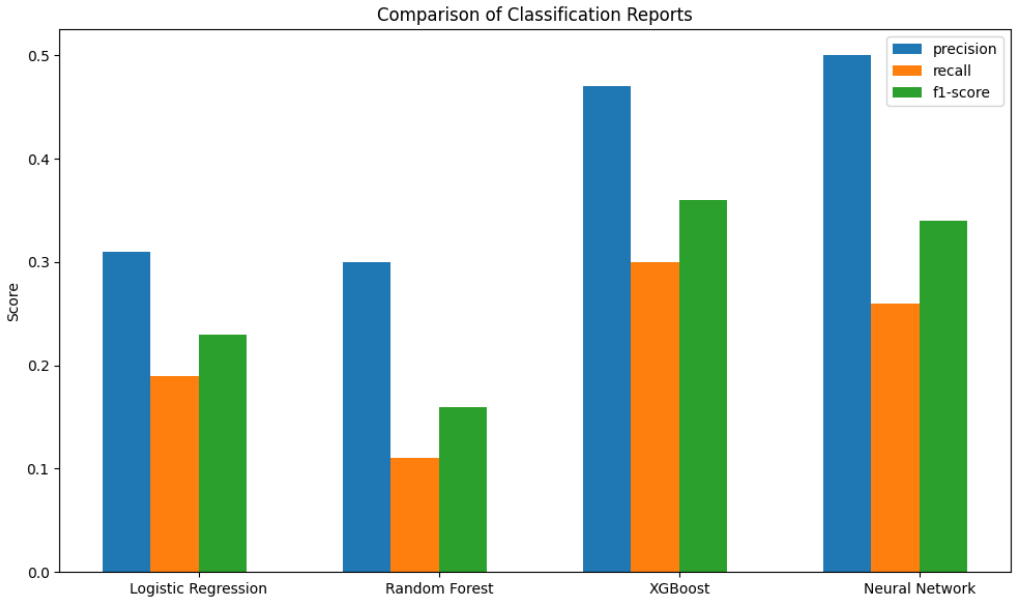
XGBoost:

	precision	recall	f1-score	support
0	0.47	0.30	0.36	27
1	0.31	0.28	0.29	69
2	0.31	0.24	0.27	114
3	0.37	0.47	0.42	157
4	0.44	0.43	0.43	161
5	0.52	0.57	0.54	115
6	0.00	0.00	0.00	6
7	0.60	1.00	0.75	3
accuracy			0.41	652
macro avg	0.38	0.41	0.38	652
weighted avg	0.40	0.41	0.40	652

Neural Network:

	precision	recall	f1-score	support
1	0.50	0.26	0.34	27
2	0.27	0.29	0.28	69
3	0.42	0.35	0.38	114
4	0.41	0.46	0.44	157
5	0.44	0.40	0.42	161
6	0.52	0.63	0.57	115
7	0.00	0.00	0.00	6
8	1.00	1.00	1.00	3
accuracy			0.43	652
macro avg	0.45	0.43	0.43	652
weighted avg	0.43	0.43	0.42	652

Here is a comparison of the results from different models:



Results and Interprets:

In each result report, there are following metrics for each class (rank category):
- Precision: The proportion of correctly predicted instances for a specific class out of all instances predicted as that class.
- Recall: The proportion of correctly predicted instances for a specific class out of all instances belonging to that class.
- F1-score: A harmonic mean of precision and recall, providing a balanced measure of the model's accuracy for a specific class.
- Support: The number of instances in the test set belonging to each class.

Comparing the accuracy values for each model:
- Logistic Regression: Accuracy = 0.42
- Random Forest: Accuracy = 0.42
- XGBoost: Accuracy = 0.41
- Neural Network: Accuracy = 0.43

Based on the accuracy metric, the Neural Network model achieved the highest accuracy of 0.43 and the precision, recall, and F1-score for most classes show improvement compared to the previous models. Therefore, ***Neural Network*** would be the optimal model for predicting player LeagueIndex or Rank for this dataset.

Communicate with Non-tech Stakeholders (some guidance)

Identify Data Gaps
"The current dataset shows promise, but we have identified some areas where additional data could enhance our understanding. Specifically, we have observed missing information in the 'Age', 'HoursPerWeek', and 'TotalHours' columns for players with a rank of 8, which corresponds to the highest rank (Professional leagues). As this top rank is of significant interest to us, it is crucial that we collect this information accurately and comprehensively in future data collection efforts. By ensuring complete and reliable data, especially for players in the highest rank, we can further improve the accuracy and insights of our analysis."

Feature Enhancement
"The current dataset contains a rich set of features that contribute significantly to predicting and representing a player's skill and ranking. However, to provide even deeper insights and enhance the predictive power of our model, we can consider incorporating additional features.

1) Player Interactions: Exploring features related to player interactions, such as social interactions and team participation, can provide valuable information. Variables like the number of friends, in-game messages, team membership, or participation in tournaments or leagues can shed light on a player's engagement, collaboration, and competitive spirit. These features can help us better understand the social dynamics and teamwork aspects that may influence a player's ranking.

2) Gameplay Strategies: Investigating features that describe a player's gameplay strategies, decision-making, and tactical choices can be highly informative. Variables like early aggression, defensive play, utilization of specific units or abilities, or strategic decision points can provide insights into a player's approach and adaptability during gameplay. By considering these features, we can gain a deeper understanding of a player's strategic thinking and their ability to adapt to different game scenarios.

By incorporating these additional features into our analysis, we can unlock further insights into the factors that contribute to a player's skill and ranking. These features will enhance the predictive power of our model and provide a more comprehensive understanding of the nuances that influence player performance in the Starcraft game."

Balancing Sample Distribution

"The current dataset shows an imbalance in the distribution of samples across different rank categories, with fewer samples available for the GrandMaster and Professional leagues. To ensure the accuracy and reliability of our model in predicting ranks across all categories, future data gathering efforts should prioritize collecting more data for these underrepresented categories.

By gathering additional data for the GrandMaster and Professional leagues, we can balance the sample distribution and provide the model with a more comprehensive representation of players across all ranks. This will enhance the model's ability to make accurate predictions and provide valuable insights for players at every level.

By addressing the sample imbalance, we can ensure that our model captures the intricacies of each rank category, from the lower leagues to the highly competitive ones. This approach will enable us to provide more equitable and reliable predictions, benefiting players across the entire spectrum of Starcraft ranks."

The Link of the Github Repository:

https://github.com/RobertChen9520/Assessment_Evil-Geniuses.git