Introduction

The goal of this project is to use machine learning models to predict the league index (ranging from 1 to 8) of a player based on their gaming behavior. The league index is a numeric representation of a player's skill level in Starcraft II, with 1 being the lowest and 8 being the highest, which represents a professional level player. This league index is often used in eSport tournaments to categorize players based on their skill level and to create fair matches.

The data used in this project includes a variety of in-game behavioral attributes such as the player's action latency, actions per minute, total minutes played, and many more. This wide range of attributes captures different aspects of a player's gameplay behavior, offering an intricate look into their gaming style and strategies.

Throughout the course of this project, I utilized multiple predictive models, including Logistic Regression, Random Forest Classifier, and Neural Networks, to analyze this complex dataset and predict player skill level. Each model was chosen for its unique strengths and combined in an ensemble to yield more robust and accurate predictions.

Metrics

Given the nature of the Starcraft player performance data, it's evident that there could be a class imbalance - certain leagues may be overrepresented compared to others. This imbalance has implications on the predictive model we build; our model may become biased towards the dominant classes (leagues), leading to higher misclassification rates for less represented leagues.

This model will ideally be utilized by gaming platforms, e-sports analysts, and perhaps players themselves, seeking to better understand and predict player performance. Therefore, the costs associated with misclassifying a player's league ranks are substantial. Misclassifying a player's league rank can lead to skewed matchmaking, unfair competition, and an overall poor gaming experience.

Predicting a lower-ranked player as high-ranked (false positive) can create a daunting and discouraging experience for the player, while predicting a high-ranked player as low-ranked (false negative) can give them undue advantage and disrupt the competitive balance.

Thus, we strive for a model with low false positive and false negative rates. To ensure our model is best-suited for classifying player league ranks, we'll employ a dual metric approach, prioritizing both overall accuracy and the F1 score. The F1 score, in particular, will provide a balanced measure of our model's precision and recall, making it a crucial metric in the face of our class imbalance problem. We aim for a model that performs well not just on the dominant

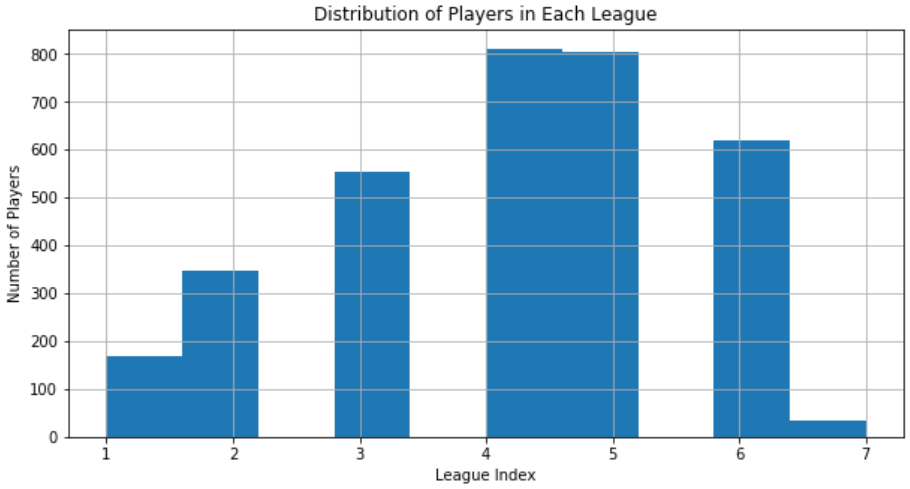
leagues, but across all league ranks for a fair and accurate representation of player performance.

Exploratory data analysis

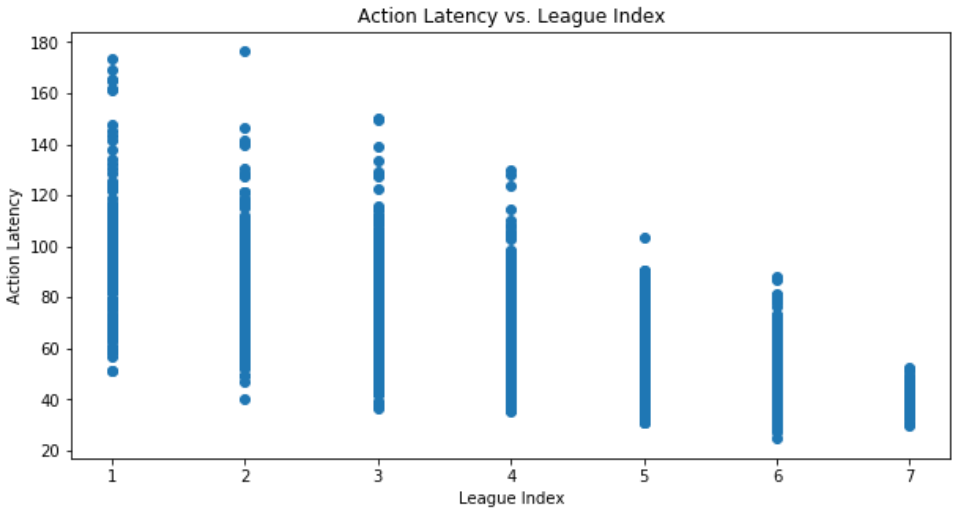
```
4    811
5    806
6    621
3    553
2    347
1    167
8     55
7     35
```

The values above represent the distribution of LeagueIndex in the dataset. Therefore we are facing the dominant class issue addressed in the metrics section.

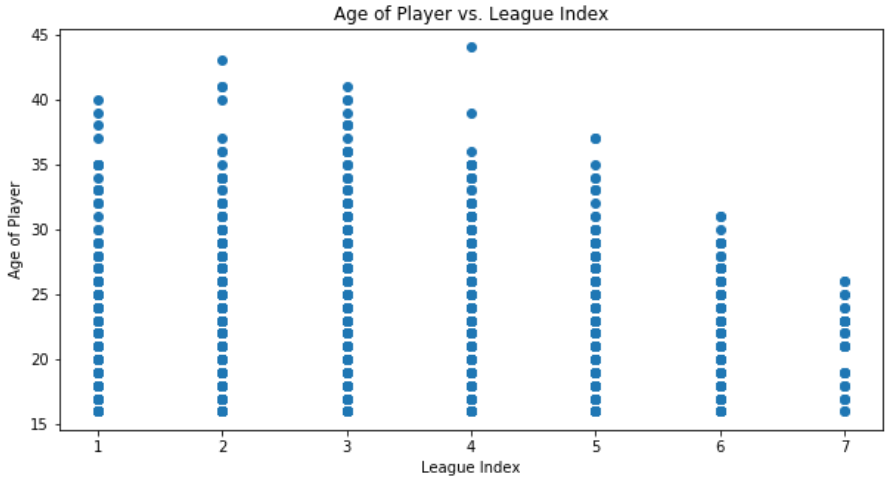
From the figure below we can infer the following trends across the dataset. League 4 and League 5 have maximum concentration of players. This can be used as a baseline for evaluating players as “High” and “Low”.

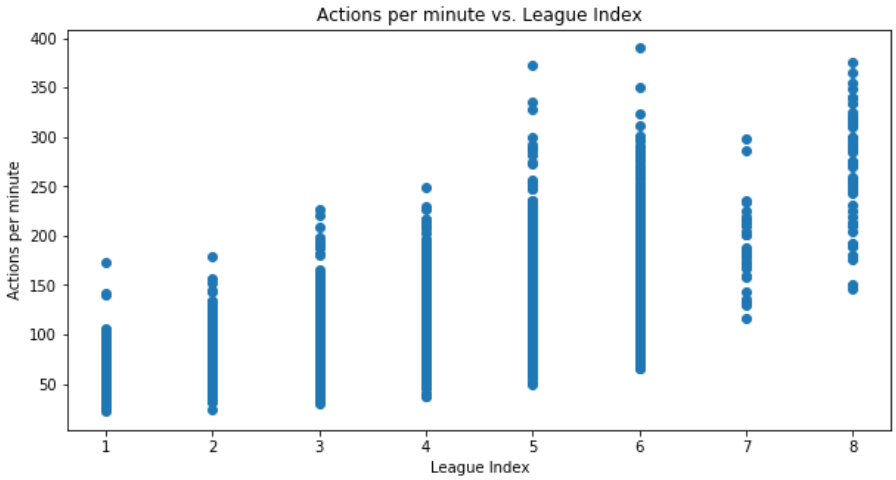


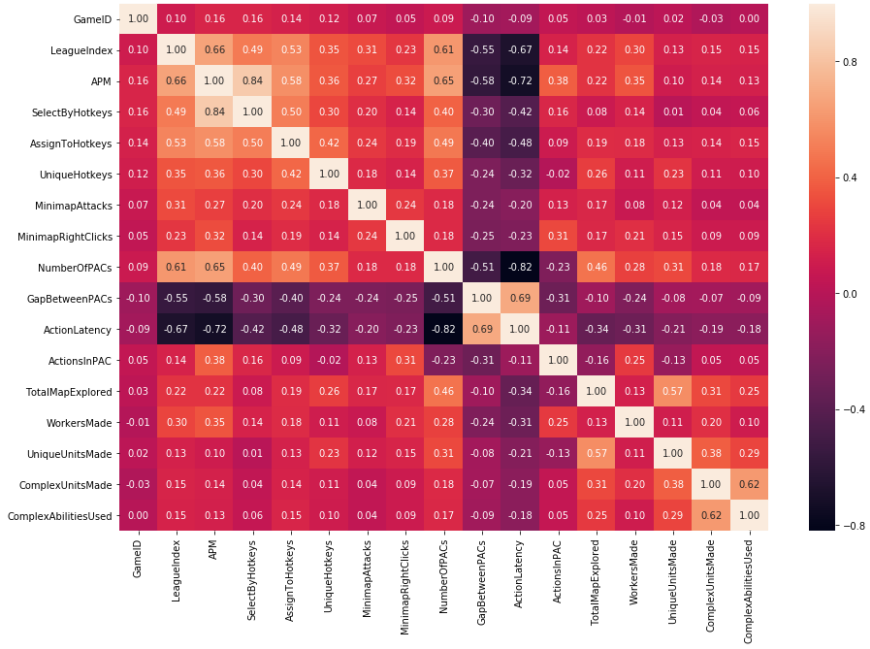
Additionally, There is a negative relation between ActionLatency and League rank. This makes sense because a higher latency slows players in game reflexes and therefore impedes their skill level.



An interesting observation is that the mean age for the best league players is quite low with respect to other leagues which means that young players have a higher probability of being in league 1. We also notice that better players have a higher APM value than other players.







As seen above I created a correlation matrix for the dataset to study the relationship between the features to ensure there is no multicollinearity in my model. There is a strong correlation between ActionsinPAC and Action Latency so I dropped ActionsinPAC since it also ranks lower on explanatory importance as presented in the results section.

Models

In this analysis, I used three distinct predictive models to understand and predict player ranks in the Starcraft game: Logistic Regression, Random Forest Classifier, and Neural Networks.

Logistic Regression was my first choice due to its simplicity and the binary nature of its classification, as it could effectively classify players into various ranks. Although a basic model,

its interpretability and ability to provide probability scores for outcomes were advantageous for the initial analysis.

Next, I incorporated a Random Forest Classifier, an ensemble machine learning algorithm that constructs multiple decision trees and merges them together. This model is known for handling higher dimensional data effectively and reducing the risk of overfitting by averaging the result. The Random Forest model's ability to detect non-linear interactions between features was a key advantage for this analysis, considering the complexity of the gaming data.

Finally, I utilized Neural Networks, which are especially effective for large datasets and complex problems. Neural Networks are a set of algorithms modeled loosely after the human brain, designed to recognize patterns. Given the large dataset and the high dimensionality of the Starcraft game data, Neural Networks were a natural choice. They allowed the model to learn and improve over time, identifying complex patterns and relationships in the data.

Each model brought its unique strengths to the ensemble, providing a robust and comprehensive approach to predicting player ranks in Starcraft. The use of multiple models helped to ensure that the final predictions were not overly reliant on a single method and mitigated the individual weaknesses of each model. Through their combined use, I aimed to achieve the most accurate predictions possible.

Results

I made confusion matrices for all the models to compare which model has the best performance based on the metrics discussed earlier. The results are as follows:\

	Random Forest	Logistic Regression	Ensemble	Neural Networks
Accuracy Rate (%)	42	37	39	37
F1 Value	0.42	0.36	0.38	0.36

The table above shows the performance of the four models used in this project: Random Forest, Logistic Regression, Ensemble, and Neural Networks.

The Random Forest model performed the best in terms of accuracy, with an accuracy rate of 42%. It also had the highest F1 score of 0.42. The Random Forest model's higher performance

can be attributed to its ability to handle complex interactions between variables and its robustness to outliers in the data.

The Logistic Regression and Neural Networks models had a similar performance, both yielding an accuracy rate of 37% and an F1 score of 0.36. Despite their lower performance compared to the Random Forest model, these models are still valuable due to their interpretability (Logistic Regression) and their ability to capture non-linear patterns (Neural Networks).

The Ensemble model, which combined all three models, yielded an accuracy rate of 39% and an F1 score of 0.38. The Ensemble model's performance highlights the power of model stacking - leveraging the strengths of multiple models to yield better predictive performance.

In conclusion, while the Random Forest model yielded the best performance in this analysis, the value of utilizing multiple models, each with their unique strengths, is demonstrated by the performance of the Ensemble model. Future work may look into optimizing these models further or exploring other machine learning models to improve predictive accuracy.