STAT 486 Report

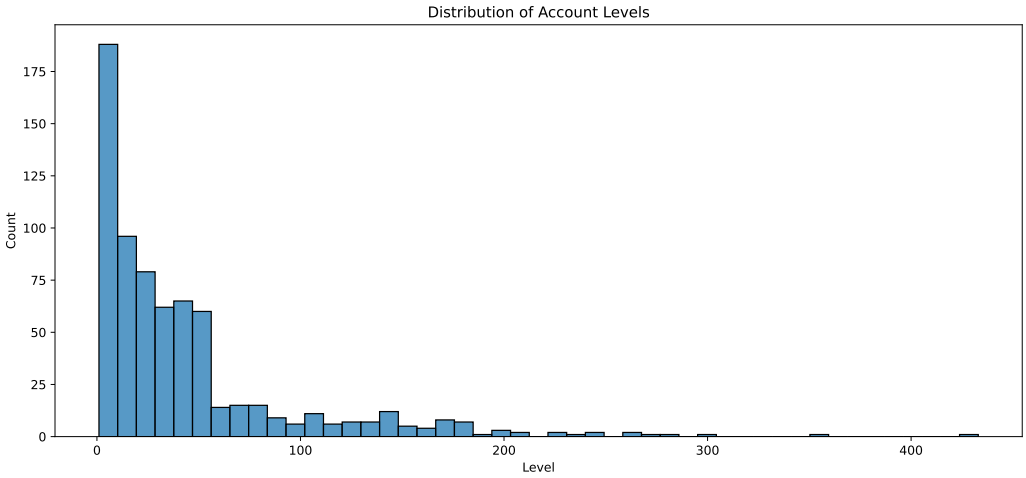
Justin Ross

2024-04-15

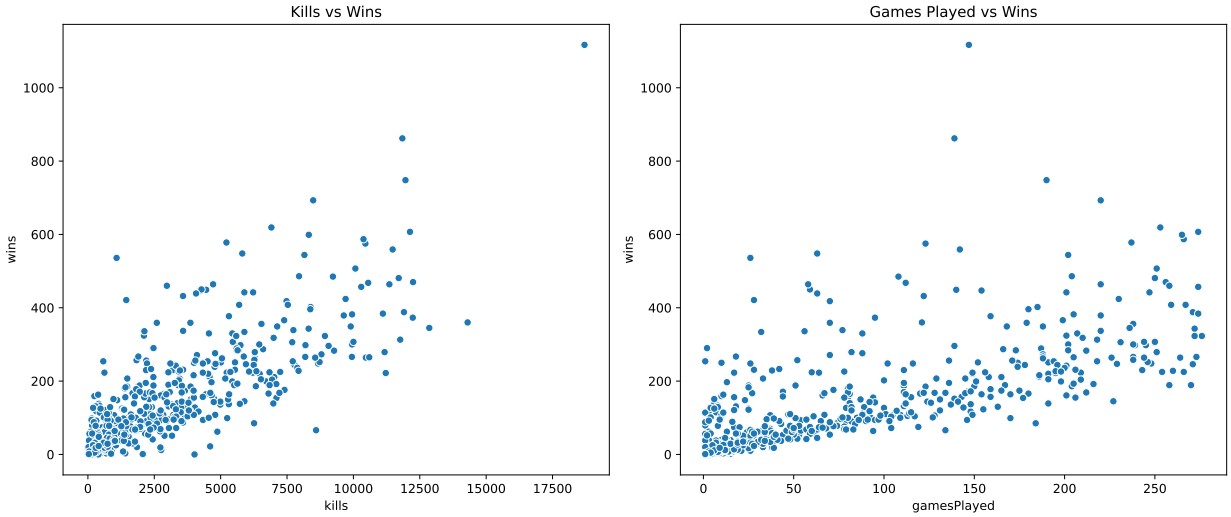
Introduction One of life's most desperately desired achievements is a win in Call of Duty. As such, our goal in this analysis is to determine the factors that contribute to success in online multiplayer matches. The data for our analysis was collected from Kaggle, and contains a number of variables relating to player performance.

Exploratory Data Analysis Our dataset originally included data on 1558 unique Call of Duty players. However, many of these players had not played any games and as such they were removed. We also removed significantly large outliers. As a result, our dataset now consists of 694 unique players.

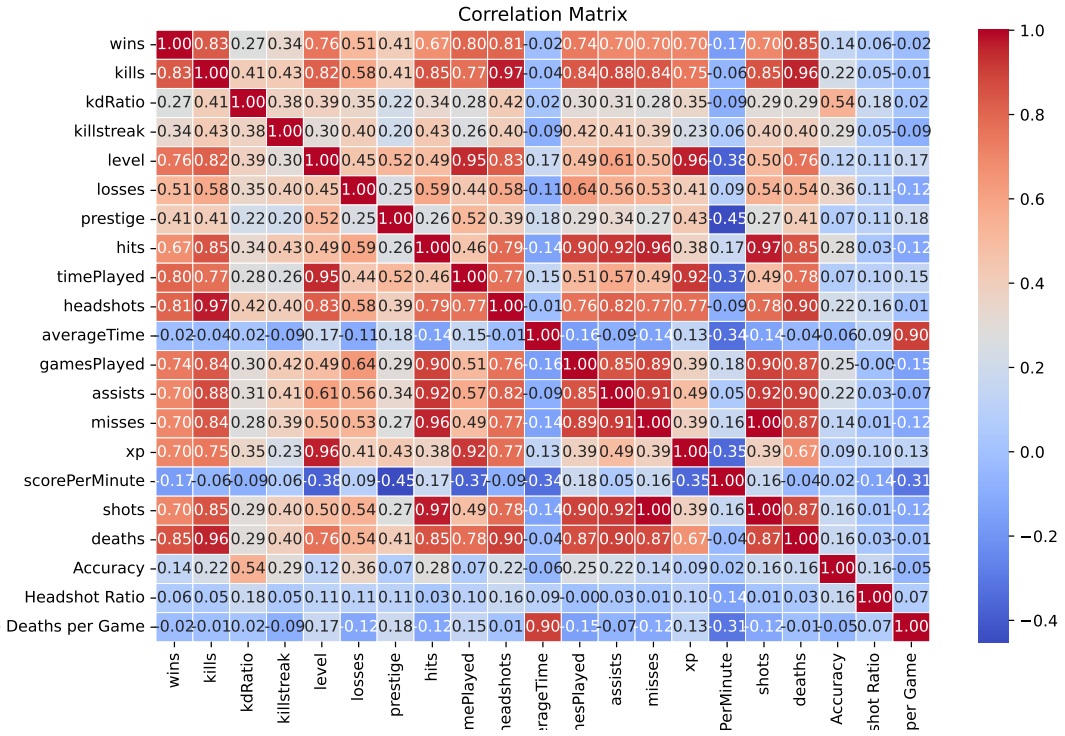
For the players in our analysis, the average number of online matches is 68, and the average number of total online wins is 116. The mean kill-death ratio is 0.838, suggesting that on average, players are killed more often than they kill their opponents. As the plot below demonstrates, most players have relatively young accounts with levels less than 50.



To better understand the relationships between the explanatory variables and the number of wins, we plotted some of the variables we thought would be most related to wins.



These plots demonstrate the high correlations between these variables and a player's number of wins. Kills has a correlation of 0.833 and games played has a correlation of 0.742. However, an issue we face is that there is correlation between some of our explanatory variables.



Due to the multicollinearity in the data, we have to be decisive in the model fitting process. We will have to be careful with linear models, as multicollinearity can cause a model to produce inaccurate model coefficients. To combat this, we will try a number of linear and non-linear models, as well as regularization and feature selection to create our best possible model.

Methods (depending on the structure of your project, only some of the following are applicable):

- Overview of feature engineering and how it improved your model (including any features from dimension reduction)
- Present a list or table of all models used. For each model, provide a very brief description (slightly longer if the model is not commonly known)

Model	Description	Hyperparameters
Linear Regression	This does Linear Regression and some cool other info goes here	hyperparameters
K-Nearest Neighbors	New predictions are the averaged target values of the k nearest neighbors in the training data	Number of neighbors, Weights
Lasso/Ridge	Linear models that penalize the size of the model coefficients (this helps combat multicollinearity)	α
SVM	Creates a margin around predicted values, trying to find an optimal balance between fitting the data and avoiding overfitting	C, Kernel, Gamma

Discussion on Model Selection: Without going into deep details, mention any patterns observed across models, such as:

- Were tree-based models consistently outperforming linear models?
- Did ensemble models show significant promise over single models?
- Talk about major pitfalls or challenges faced with certain models, e.g., overfitting with a deep neural network, or convergence issues with a certain algorithm. Briefly touch upon why certain models didn't make the cut. This doesn't have to be detailed but can include reasons like:
- Poor performance on validation data.
- Overfitting issues.
- Computationally too intensive for the marginal gain in accuracy.
- Difficulty in hyperparameter tuning.

To our surprise, tree based models did not consistently outperform linear models. We used a random forest to predict wins, but the best test RMSE that we obtained was higher than the RMSE that we obtained with linear models like linear regression and penalized linear regression.

Ensemble methods also did not show significant promise over single models. The test RMSE from the random forests and gradient boosting models performed about the same as single models like k-nearest neighbors and performed worse than linear regression and penalized linear regression.

The biggest challenge that we all faced with more complex models like random forests and gradient boosting was overfitting. We consistently obtained significantly higher training RMSE's than testing RMSE's which was a surprise. Even with extensive hyperparameter tuning on the number of trees, depth of trees, and number of features considered at each split, the testing RMSE never got super close to being less than or equal to the training RMSE. Hyperparameter tuning on the learning rate and number of estimators in the gradient boosting model yielded similar results to the random forest.

Gradient boosting and random forests are both robust machine learning methods and we are sure that with an even more thorough hyper parameter tuning process, we could have found a model that we were satisfied with that doesn't overfit the data. However, we determined that the computation would be too intensive for

the marginal gain in accuracy. Even though k-nearest neighbors and penalized linear regression are simple methods, we are satisfied with the RMSE that it yielded and the short amount of time it took to run.

Detailed Discussion on Best Model:

- Go in-depth into the model that performed the best
- Hyperparameter tuning
- Performance metrics
- SHAP and/or feature importance results including a few individual false positives/false negatives and true positives/true negatives (adapted to regression if applicable)
- Insights from cluster analysis or anomaly detection

CLUSTERING
A K-means clustering analysis offered interesting insights into player performance differences. It split players into four groups based off their stats: new players, elite players, experienced and competent players, and intermediate active players.

- New Players (Cluster 0)
  - These players that have low wins (30.25) and kills (478.57). These players also have a low kill-death ratio (0.71) and engage minimally with the game, as indicated by lower levels (11.73) and moderate prestige (14.5).
- Elite Players (Cluster 1)
  - Players who are highly skilled veterans with high wins (395.79) and an extensive number of kills (9112.71). High levels (173.21) and maximum prestige (100.24) indicate significant time investment and expertise.
- Experienced and Competent Players (Cluster 2)
  - Players with moderate-to-high wins (211.98) and kills (4828.83), displaying a good kill-death ratio (0.96). With substantial prestige (62.08), these players are very good but might not play as intensively as the elite players.
- Intermediate Active Players (Cluster 3)
  - Characteristics: Players who engage deeply during play sessions with moderate wins (103.96) and kills (1654.91), high prestige (99.40), and significant game time per session (47.34 minutes).

Conclusion and Next Steps:

- Conclude by reinforcing the choice of the best model and its implications.
- Report or reinforce the answer/conclusion about your original problem statement
- Discuss any potential future steps or improvements that could be made, perhaps leveraging models or techniques not yet explored.