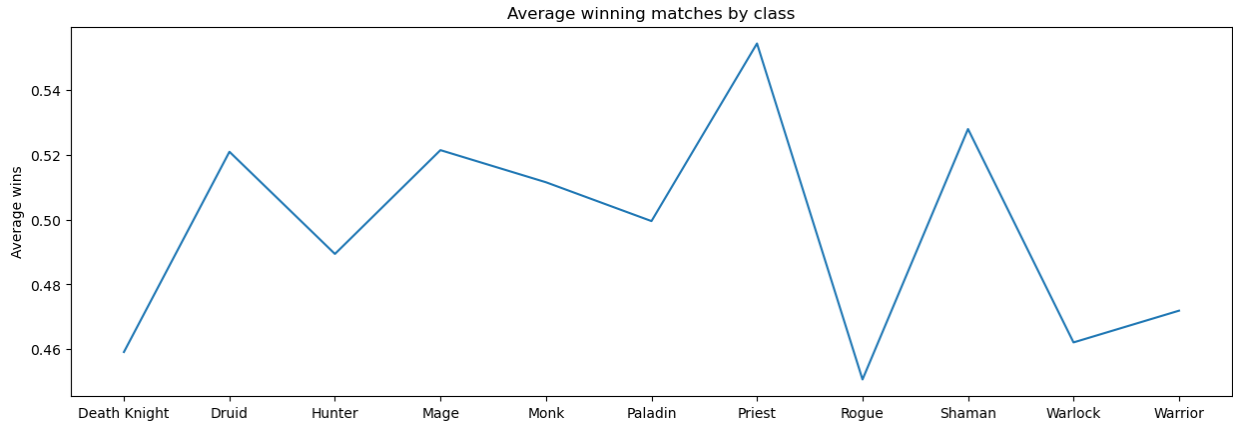
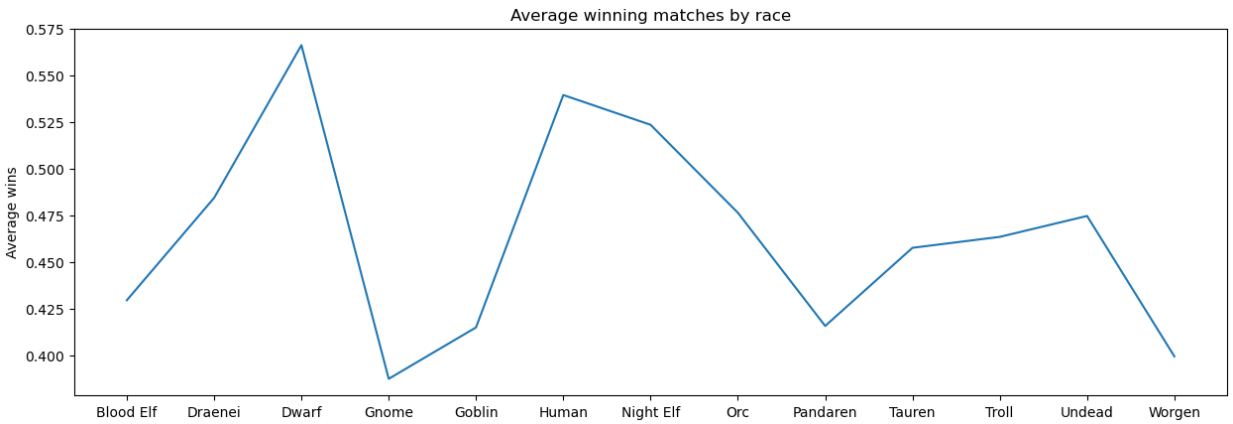
# WoW Stats Post-Implementation Report

MMOMoney is a betting website for online games and the company wants to begin accepting bets for World of Warcraft (WoW) two-player versus two-player (2v2) arena matches. The company needs a reliable way to predict the outcome of these matches for this business venture to be profitable. This application provides a more reliable way to predict match outcomes.

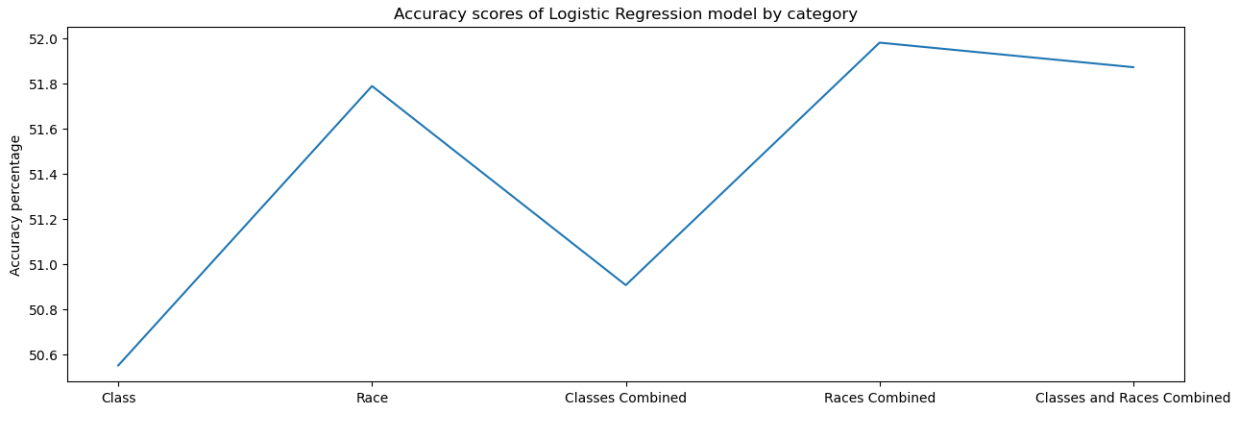
Historical match data from a dataset on Kaggle.com was used in the development of this project. Several subsets of data were examined over the course of the project to determine the most reliable predictor of match outcomes. Because class was assumed to be the most important factor, that data was examined first. Afterwards, predictions were tested using race. Surprisingly, race was much more reliable. Combinations of race/class, class/class, and race/race were also considered, with race/race being the most reliable. Trends were identified using averages and sklearn was used to train machine learning models and develop accuracy scores for each prediction. Average winning matches by race and class can be observed in line graphs within the application. A line graph depicting the accuracy scores of each model follows the averages graphs. The following graphs show average wins by class and race:





After the most accurate metric had been identified, a logistic regression model was used to predict the win for a combination of two races selected by a user. Since the goal of the application is to predict the outcome of a 2v2 match, the application makes predictions for each pair of two races and compares them. If the model predicts that both pairs will either win or lose, a tie-breaking function decides the predicted winner based on the sum of average winning matches for the given races in each pair. This metric was decided as the tiebreaker because it was more accurate than all the other measurements in the application (~58%). This is shown in the following graphs:

The validation method used to check the machine learning model was the “accuracy\_score” method from the sklearn metrics module. The score varies slightly from one run to another, but they were approximately 50.5% for class, 51.8% for race, 52% for a combination of race/race, 51% for a combination of class/class, and 51.9% for a combination of race/class/race/class. The following graph depicts these results:



The following page contains a step-by-step user guide for navigating this application.

# WoW Stats Application User Guide

1. Navigate to <https://colab.research.google.com/drive/1Wmow23SV98vdvjcFApc--PGkTlxHRBzt?usp=sharing> in a web browser. This is a Jupyter Notebook application with cells containing Python code. Each cell must be run in the order they are presented for the application to work properly by clicking the triangular play button in the lefthand margin of each cell.
2. The first cell imports all necessary libraries for machine learning models, displaying graphs, and widgets for user input. Click run and wait for the asterisk in the left margin to disappear (approximately 30 seconds).
3. The next cell calculates average wins by class, trains a machine learning model on class data, calculates an accuracy score for that model, and displays a graph for averages. Click run to observe this data.
4. The next cell performs the same operations as the previous cell, except they are based on race. Click run to observe this data.
5. The next three cells train models and calculate accuracy scores for combinations of race/race, class/class, and race/race/class/class. Click run on each cell to observe that model’s accuracy score.
6. The next cell displays a graph showing the accuracy score of each machine learning model previously tested. Click run to display the graph. Here we can see that a combination of races is the most accurate model.
7. The final cell takes user input and uses our most accurate model of race/race to predict the outcome of a match. Select the appropriate race from each dropdown box and click “Predict Winner”. Below this button the program will either print “Team 1 will win” or “Team 2 will win”. Happy gaming!