# StarCraft 2: Predicting Rank through Player Game Data

Submission for Evil Geniuses Data Scientist assessment.

### Table of Contents

### Directory Structure

```
├── data
│   ├── external        <- Data from third party sources.
│   ├── interim         <- Intermediate data that has been transformed.
│   ├── processed       <- The final, canonical data sets for modeling.
│
├── models              <- Trained and serialized models.
├── notebooks           <- Jupyter notebooks containing analysis and documenation.
├── src                 <- Source code and scripts used in this project.
│   ├── data            <- Script to extract, transform, and load data.
│   │   └── make_data.py
│   │
│   ├── models          <- Script to train models and serialize.
│   │   ├── build_model.py
├── README.md              <- Report for users/stakeholders.
```

## EDA

Please see notebooks/1-EDA.ipynb for the exploratory data analysis and documentation of decision making.

## ETL

Please see notebooks/2-ETL.ipynb for data preprocessing and transformation documentation and walkthrough.

## Model

Please see notebooks/3-Model.ipynb for feature selection, model training, and model comparisions.

## Results

In this project, our goal was to predict a StarCraft 2 player's rank using performance data within games and compared 3 classifier models to do so:

- Logistic Regression
- Naive Bayes Classifier
- Support Vector Machines

Each model was evaluated using the accuracy score, which represents the fraction of predictions our model predicted correctly.

Using cross validation (a robust method to validate the performance of models) to expose our models to 5 different subsets of the data, we arrived at these mean accuracy scores for each model:

- Logistic Regression: 0.408
- Naive Bayes Classifier: 0.473
- Support Vector Machines: 0.410

Based off the accuracy score of 0.473, the best model for our problem statement would be the Naive Bayes (NB) Classifier .

The better performance of the NB classifier could be due to the fact that NBs require much less training data with a small number of features than other models. The data provided to use was a relatively small dataset with only 3,395 rows and 19 features (and 1 target variable).

However, the low scoring on each of the 3 models reflects a lack of quality data to fully explain the differences between low performing players and high performing players.

### Guidelines for Future Data Collection

As mentioned previously, not only is more data needed, but higher quality data is needed as well. While conducting our EDA, we noticed that many of the features provided demonstrated little to no variance when analyzed across the LeagueIndex, meaning that the values and differences were too small for a model to fit its complexity. Additionaly, the class imbalance of the data should be addressed by gathering data as evenly as possible across the various ranks, although it may be harder to obtain data from professionals as evident by the number of missing data.

In conclusion, the features within this dataset mainly focused on the mechanics of one game (e.g. numberofunitscreated, minimap clicks, etc.) rather than data regarding the players statistics over time. Every player has a variability of performance and every map also requires different forms of mechanics. Having aggregated player data over time can help us get closer to a true mean of what a higher performing player looks like versus a lower performing player.