

Predictive Ranking in PUBg

Joe P.



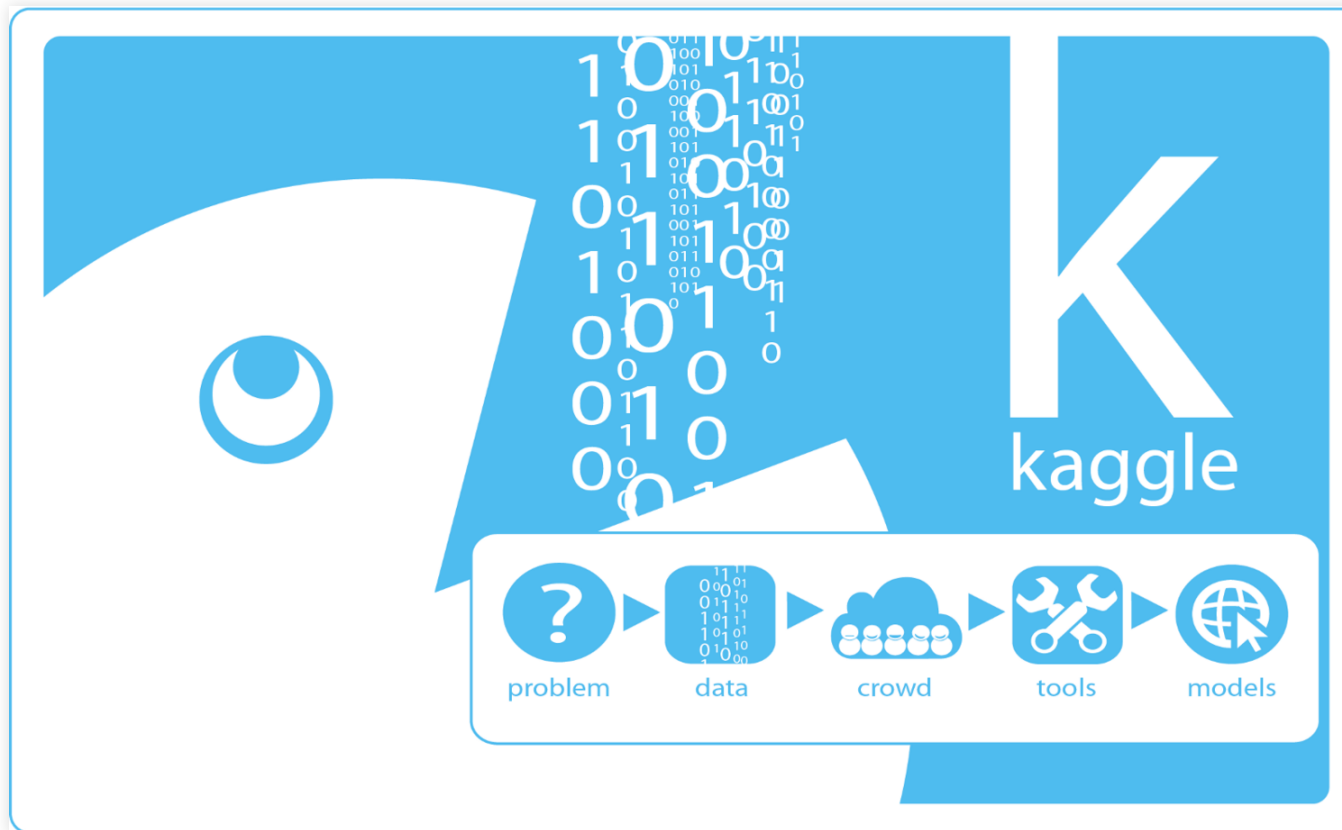
The Problem

Kaggle Competition: <https://www.kaggle.com/c/pubg-finish-placement-prediction>

- 100 people are dropped on an Island (littered with weapons)
- How to predict who lives?
- What's the best survival strategy?

Data Source

- Kaggle
- PUBg made data public
- Train & Test datasets



Data Source

The Data Available

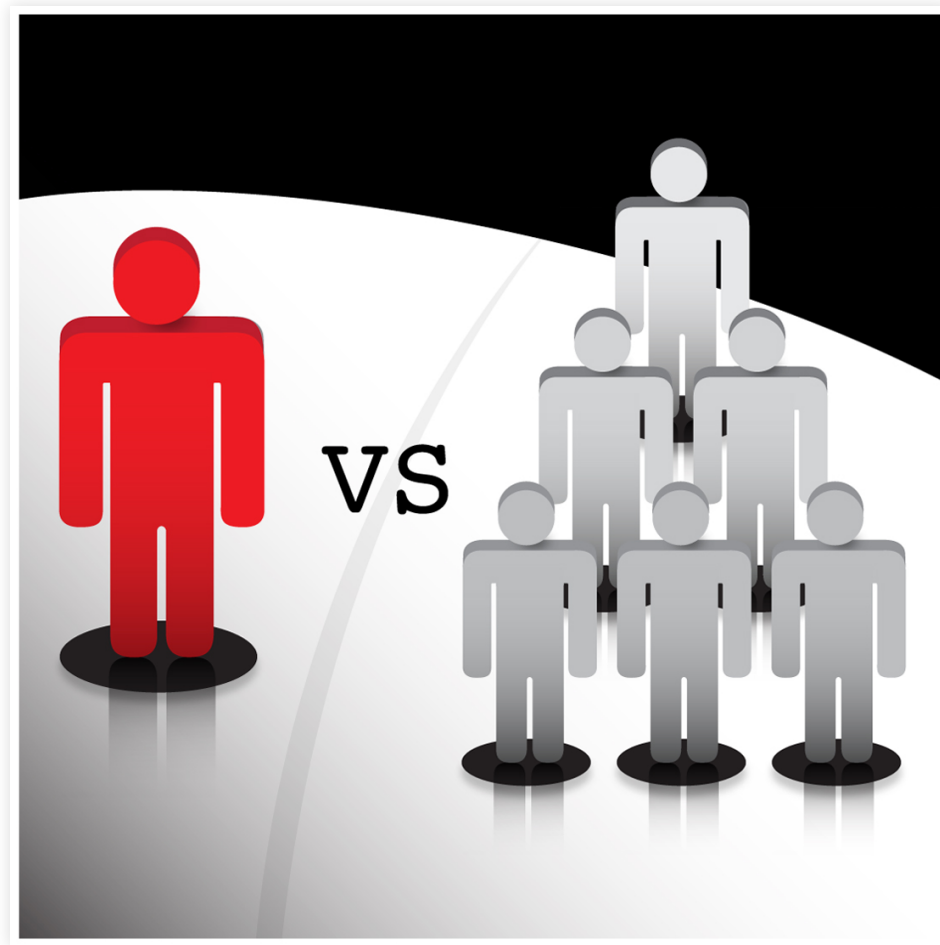
- Training Set: from ~4.5 million games
- Test Set: from ~2 million games
- Variables: Match ID, Type, Duration, Individual Stats

Methods

- Linear Models
- Regression Trees
- Neural Nets

Methods

- Split by game type
 - Solo, Duo, Squad, “Other”
 - Solo -> Neural Net
 - Rest -> Decision trees



Methods

My Linear Models:

- $WPP \sim kP + wD + rD + dD + k + wA + mP + b + IK$
- $WPP \sim kP + k + kS + mD + mP + rD + wD + wA$
- $WPP \sim kP + rD + wD + wA + rP + DBNOs$
- $WPP \sim kP + \text{factor}(rD) + \text{factor}(wD) + \text{factor}(wA)$

*WPP = Win Place Percent (percentile winning placement)//
kP = ranking in match based on enemies killed// k = number of kills//
kS = max number of players killed in a short time period// mD = match
duration// rD = distance traveled in vehicle// wD = distance traveled on
foot// wA = number of weapons acquired// DBNOs = number of players
knocked down, but not killed// mP = Max Place possible that round//
b = Boosts used// IK = Longest Kill*

Methods

My Tree Models:

- $WPP \sim kP + kS + wD + rD + wA$
- $WPP \sim kP + wD + rD + dD + k + wA + mP + b + IK$

Methods

My Neural Network Models:

- $WPP \sim kP + wD + rD + dD + k + wA + mP + b + IK$
- Up to 10 hidden layers

Results

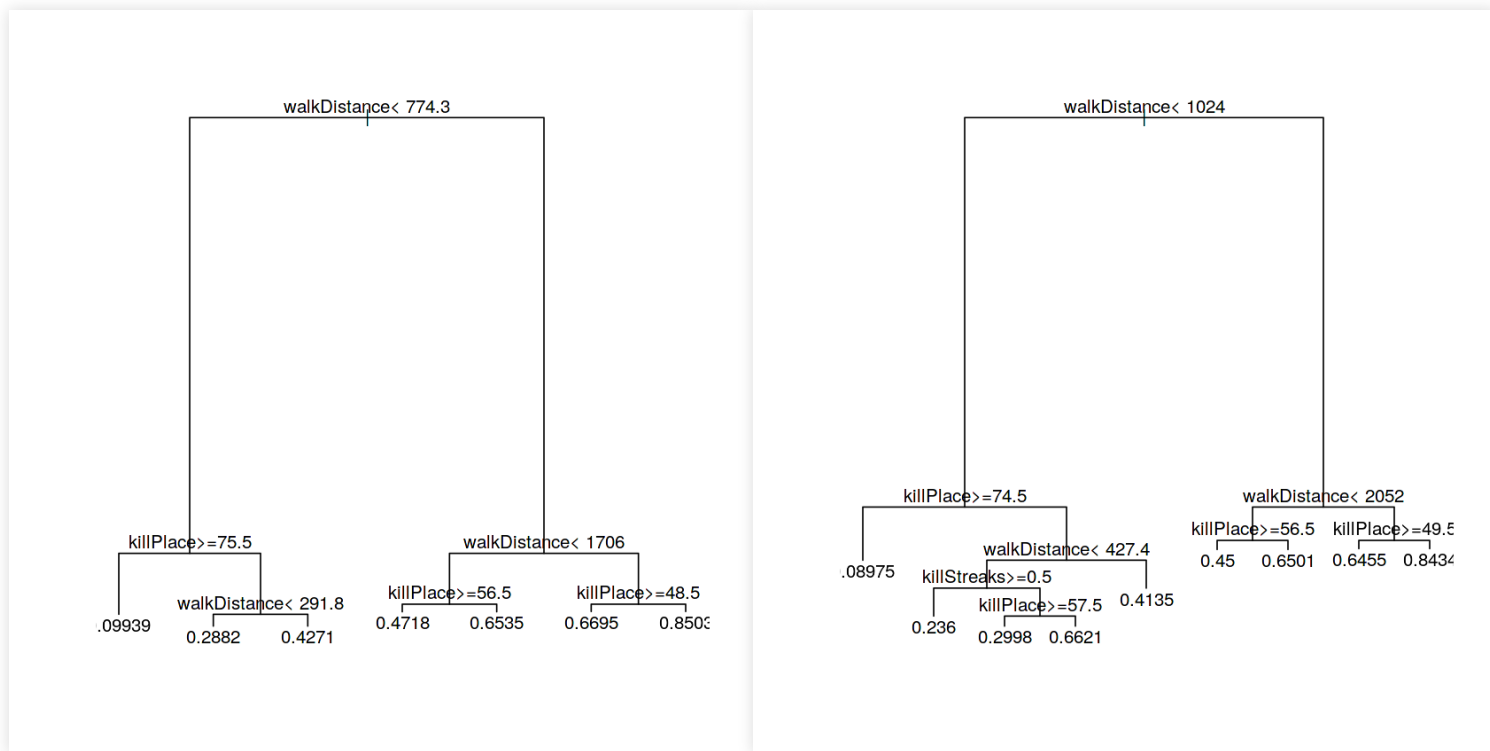
- Me:0.098 error rate
- Contest winner:0.013 error rate



Results

My Trees:

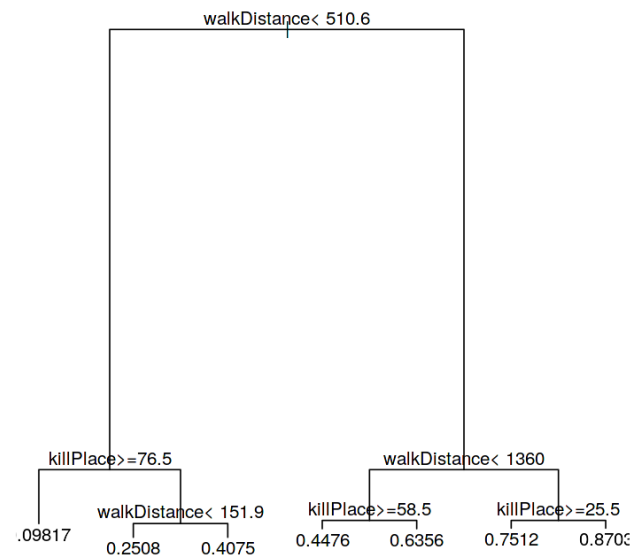
Duo Model-----Squad Model



Results

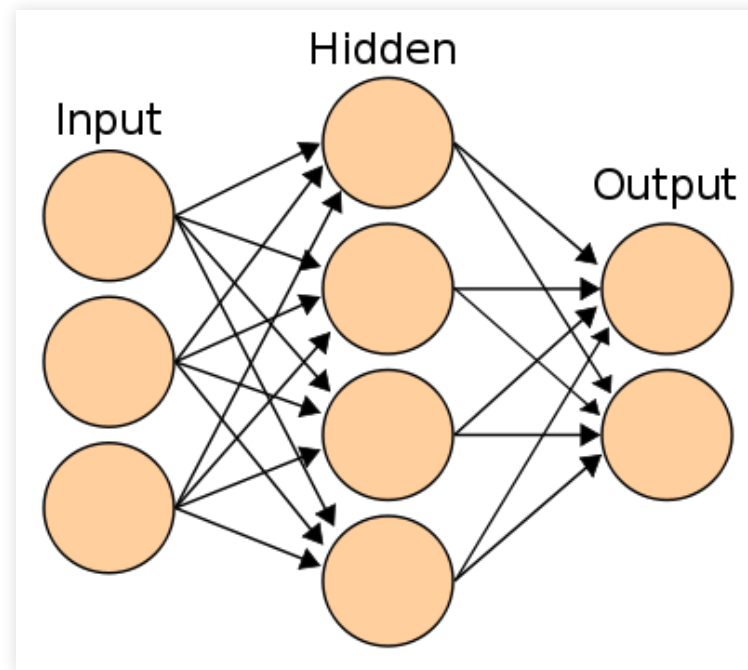
My Trees (Cont.):

“Other” Model



Results

My Neural Net:



Insights

- Keep your distance
 - Running is the best way to stay alive longer
 - Kill them before they get close (*Long distance kills correlated*)
- “Skill” did not matter



Other Attempted Strategies

- Random forest (required more computing power)
- Created other complex variable
 - Example: total distance = swim + ride + walk
 - Example: skill = headshots / kills
- Nearest neighbors regression (was not successful)

Next Steps

- Determine situations the model fails
 - Example: If it consistently mixes up players at the top of the ranking
- Ranked ordering by Match
- SVM
- Consider who are teammates/enemies

All the **talent** in the world
won't take you anywhere
without your teammates.

Anonymous

@AthleteAssess

Limitations

- My computer is slow (costly to test on large datasets)
- Variables not provided (e.g. type of weapon, proximity to X, etc...)
- There are always more prediction techniques to learn!

The End

- Questions?