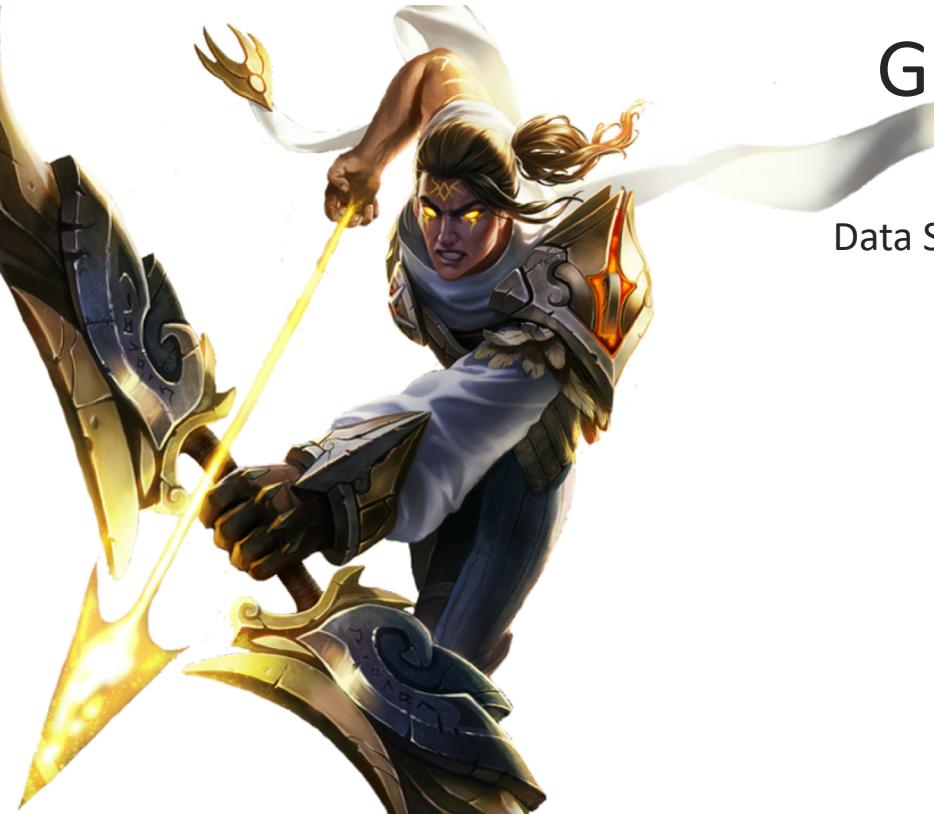




LoL Mid-season Invitational Game Result Prediction

Data Science | Machine Learning | Web Application

Zigeng Fu, Hao Cui, Yimu Jin, Ziyan Zhu



LoL Background Introduction

Annual Series in Worldwide **6** Areas:
LDL, LPL, LCK, LMS, LCS, LEC

2019 Spring season finished…
Incoming 2019 **Mid-season Invitational**

Opponents in a Game:
Red side vs Blue side

5 roles in each side:
Top, Middle, ADC, Support, Juggler

Player Performance Criteria:
Kill, Assist, Death, Engage Rate…



Project Overview

1 Goal: **Predict game results** in incoming mid-season invitational

2 Methods: **3 Machine Learning Models**



Neural Network (MLP)

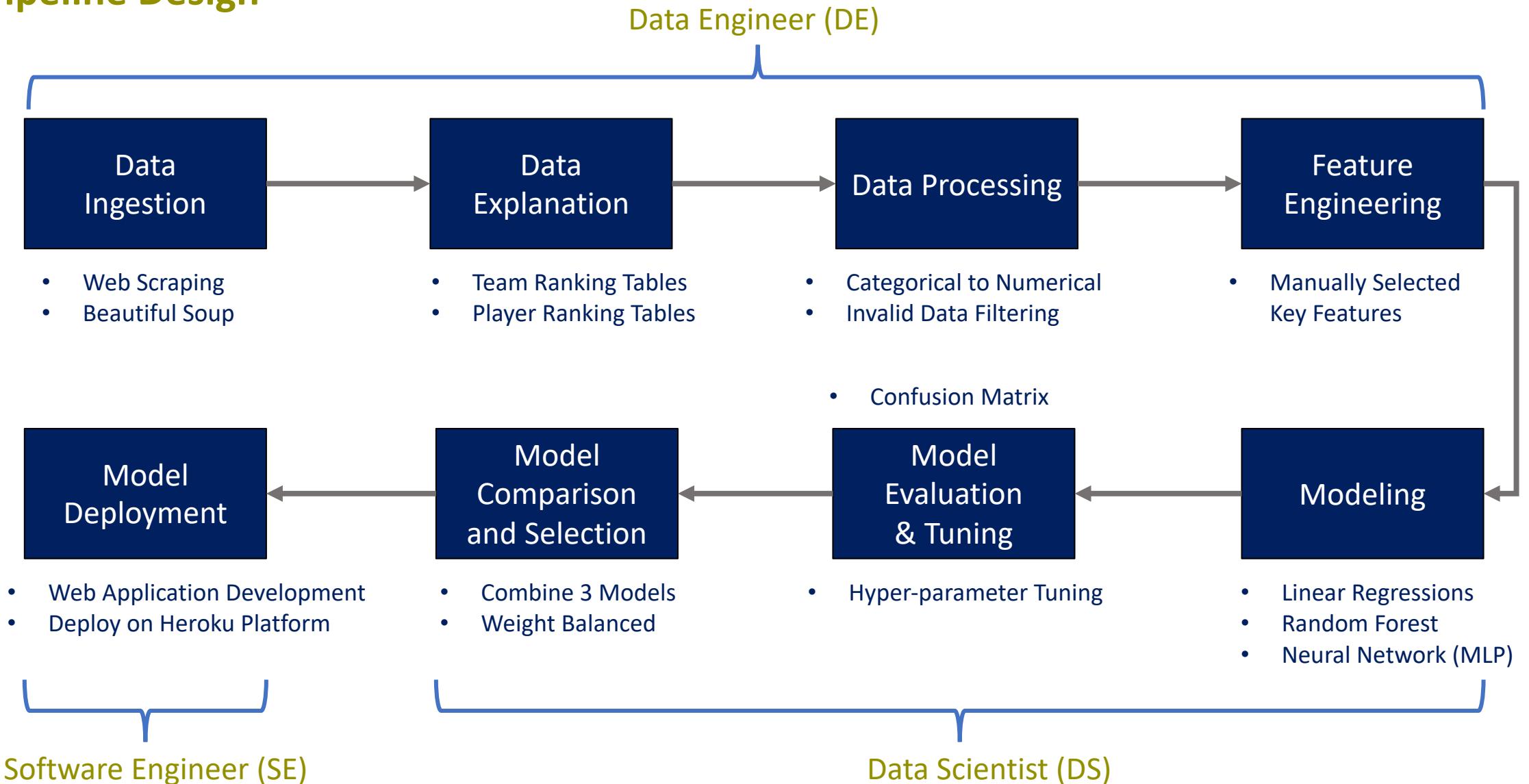
Logistic Regression

Random Forest

3 Outcome: **Predicted Game Result & Probability**



Pipeline Design



Roles in Our 'LoL' Team



Mr. Hao Cui

- Data Scientist



NEU INFO6105 2019 Spring Team 6 Members

Mr. Zigeng Fu

- Project Manager
- Frontend Developer



Mr. Yimu Jin

- Data Engineer
- Backend Developer



Miss. Ziyuan Zhu

- Data Engineer
- DevOps Engineer

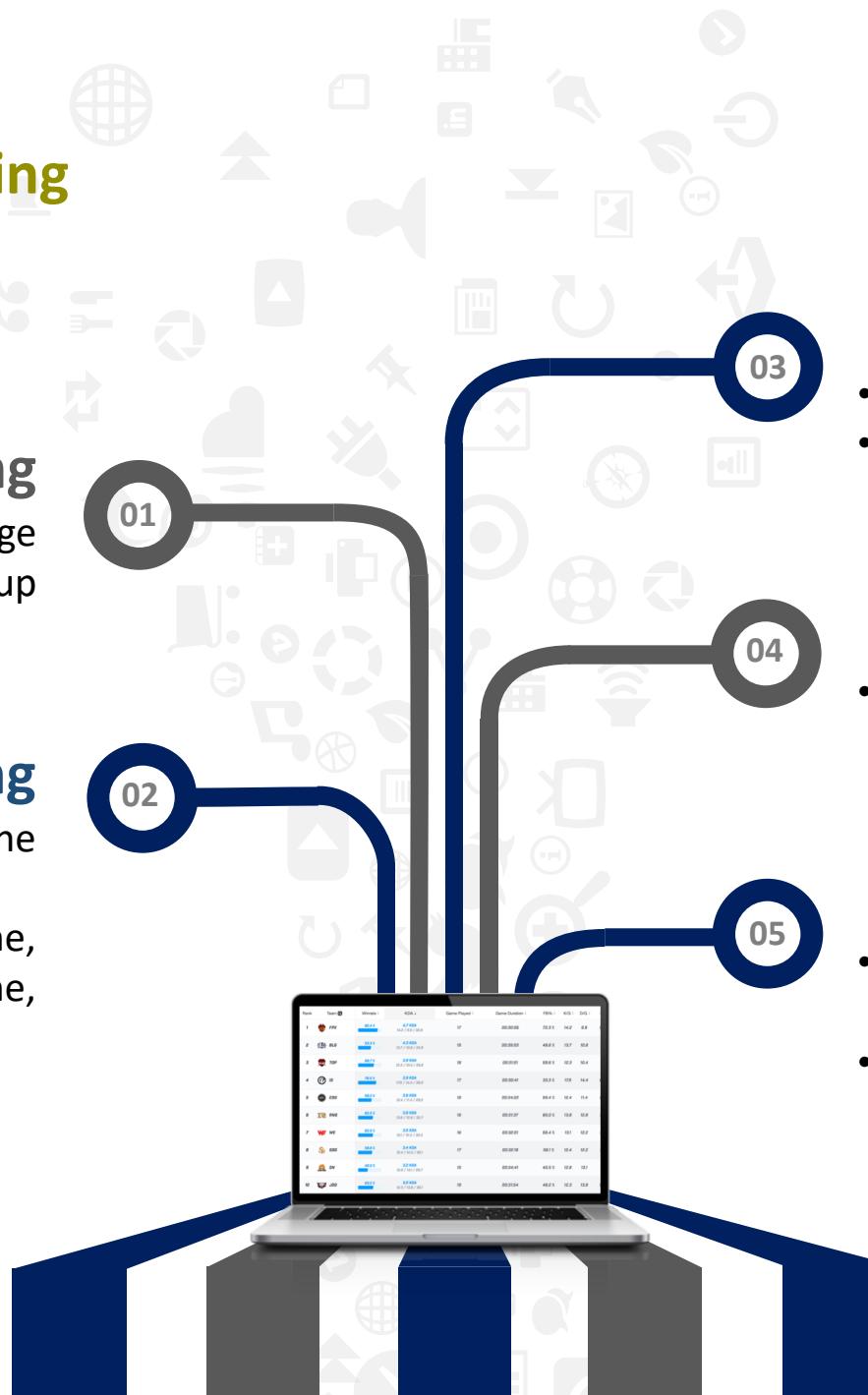
Data Injection and Processing

Players Data Scraping

- Team, KDA, Average Kill, Average Assistance, Average Death, Group Fight Engage Rate...

Game Data Scraping

- The game result dataset of in the format: **Team vs Team**
- Team 1 Name, Team 2 Name, Scores, Results, Date and Time, Season...



Invalid Data Filtering

- Filter the records with missing value
- Filter the games with even results

Categorical to Numerical

- Convert from Win/Lose to 1/0

Data Mapping

- Map the game results data with players data
- Come out the result dataset in the format: **Players vs Players**

Feature Engineering

51 Selected Features:

1 Result + 2 Teams × 5 Roles × 5 KPIs

Win/Lose

Red
vs
Blue

Top
Middle
Juggle
ADC
Support



Game Data



Team and Player Data

5 KPIs



Average Kill
Average Assistance
Average Death
KDA (Kill-Death-Assist)
Group Fight Engage Rate

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Models Evaluation (Confusion Matrixes)

1 Random Forest

- Train F-score: 0.90
- Test F-score: 0.58

Confusion Matrix (Training Set)		True Condition	
		Actual Win	Actual Lose
Predict Condition	Predicted Win	TP (338)	FP (48)
	Predicted Lose	FN (38)	TN (331)

Training Set

Testing Set

Confusion Matrix (Testing Set)		True Condition	
		Actual Win	Actual Lose
Predict Condition	Predicted Win	TP (55)	FP (40)
	Predicted Lose	FN (41)	TN (53)

2 Logistic Regression

- Train F-score: 0.72
- Test F-score: 0.66

Confusion Matrix (Training Set)		True Condition	
		Actual Win	Actual Lose
Predict Condition	Predicted Win	TP (270)	FP (99)
	Predicted Lose	FN (106)	TN (280)

Confusion Matrix (Testing Set)		True Condition	
		Actual Win	Actual Lose
Predict Condition	Predicted Win	TP (61)	FP (29)
	Predicted Lose	FN (35)	TN (64)

3 Neural Network (MLP)

- Train F-score: 0.90
- Test F-score: 0.71

Confusion Matrix (Training Set)		True Condition	
		Actual Win	Actual Lose
Predict Condition	Predicted Win	TP (268)	FP (94)
	Predicted Lose	FN (107)	TN (286)

Confusion Matrix (Testing Set)		True Condition	
		Actual Win	Actual Lose
Predict Condition	Predicted Win	TP (57)	FP (28)
	Predicted Lose	FN (40)	TN (64)

Hyper-parameter Tuning

Regression Model

Parameter adjusted: **L1, L2, L-BFGS**

Test Set Evaluation	Recall	Precision	F-score
Original	0.64	0.68	0.66
Tuned	0.68	0.68	0.68

Random Forest (No. of Trees, Tress Depth)

Parameter adjusted: **n_estimators, random state**

Test Set Evaluation	Recall	Precision	F-score
Original	0.57	0.58	0.58
Tuned	0.57	0.58	0.58

Neural Network (dropout, loss, activation, optimizer, epochs)

Parameter adjusted: **dropout, loss, activation, optimizer, epochs**

Test Set Evaluation	Recall	Precision	F-score
Original	0.59	0.67	0.63
Tuned	0.78	0.64	0.70

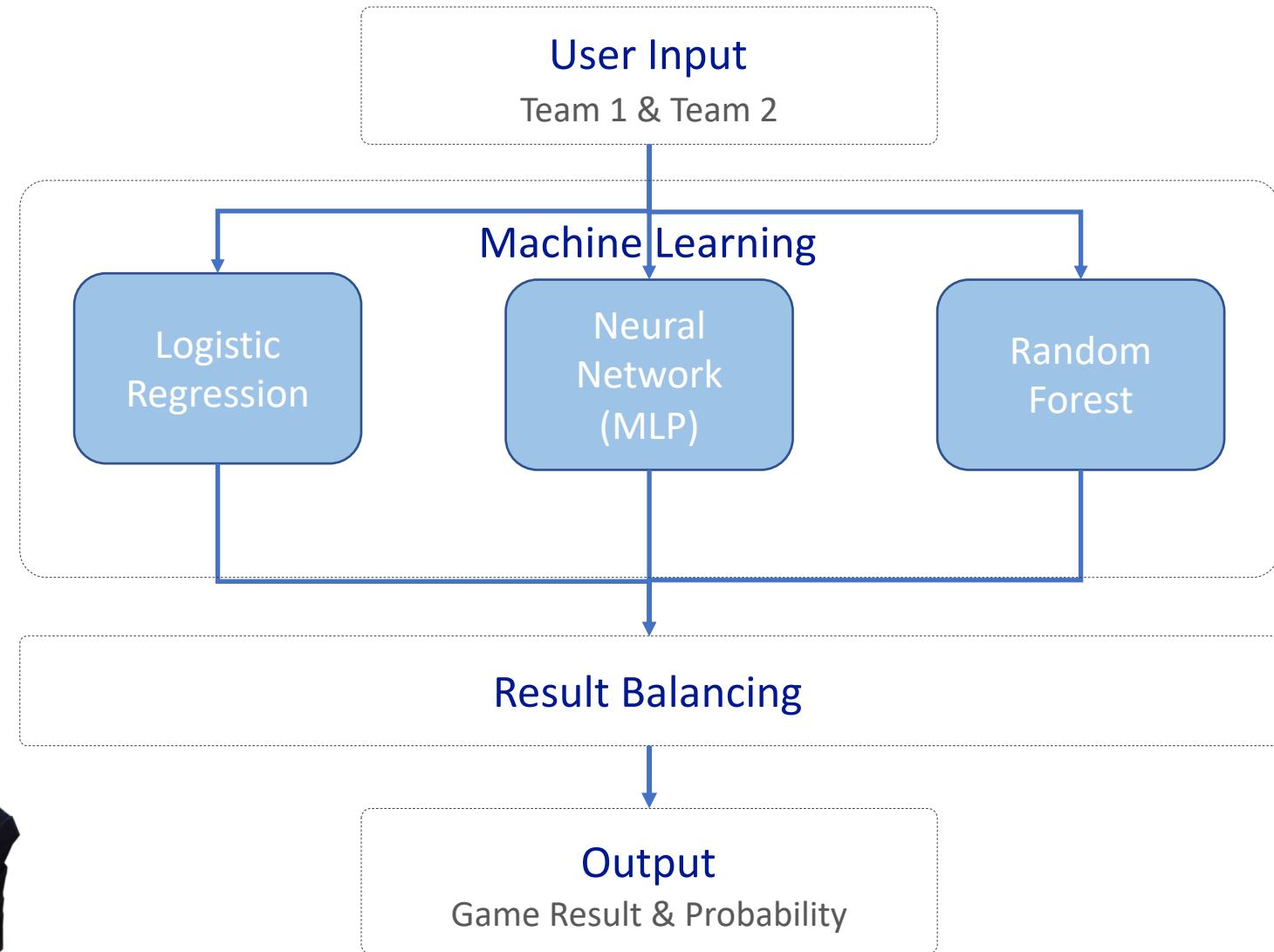
Performance:

1. Neural Network
2. Logistic Regression
3. Random Forest

But not so large differences...

Model Selection

- Combined-model solution
- Improve the accuracy of prediction



Web Application DevOps

Web Development
Framework

Frontend Development



App Deployment



Demo Presentation

Thanks for professor, TAs and contributions from lovely team members