
A128 Team 9

# League of Legends Data Science Mini-Project

Member:
Qian Jianheng Oscar (U2220109K)
Dimas Rizky (U2221229F)
Jain Mokshit (U2222438K)

# Table of Contents

# Introduction

## 2009 - Current

Game released

**153 Million**

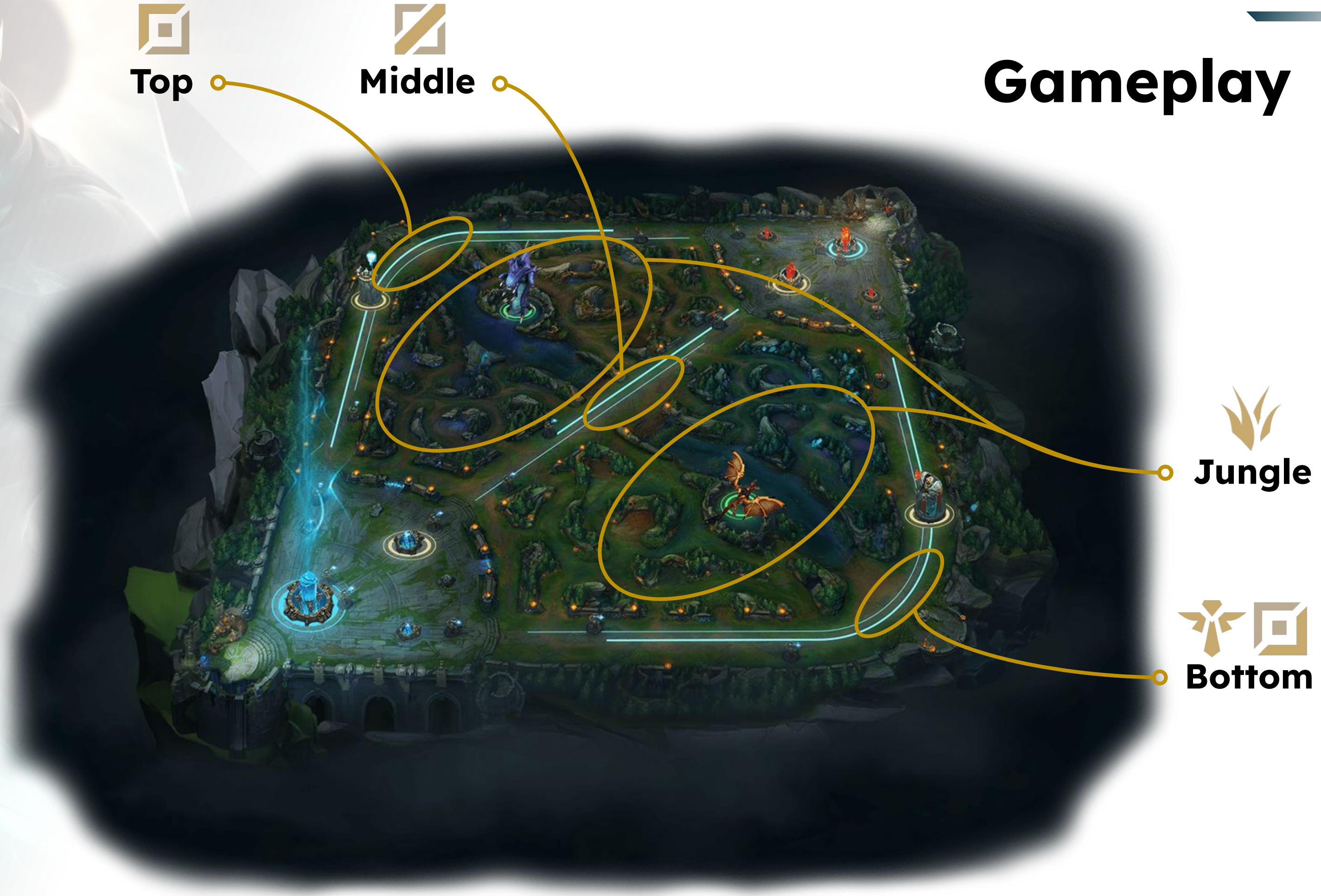
Monthly Active Players

**$2.23 Million**

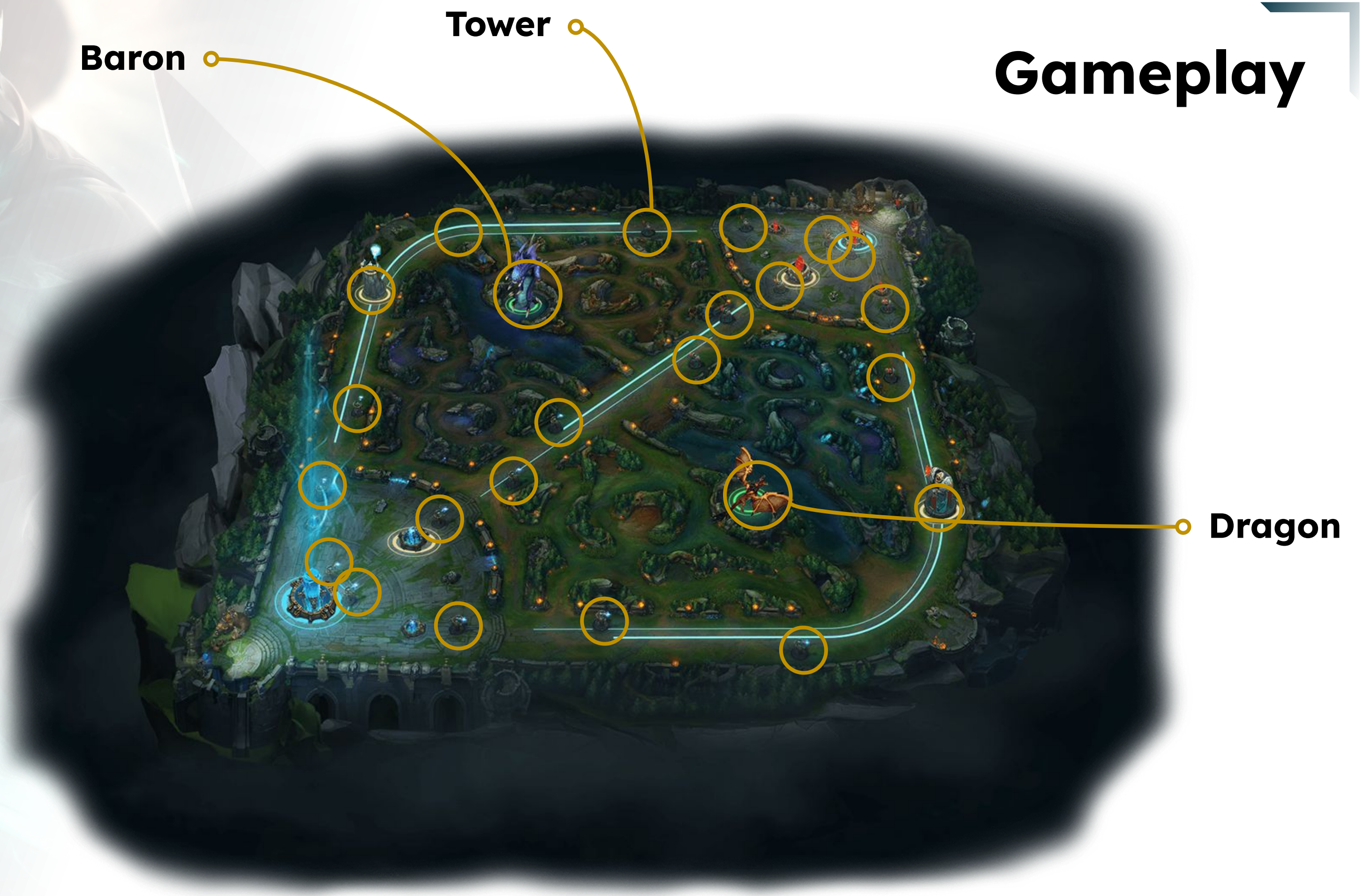
World Championship
Prize Pool

# Gameplay

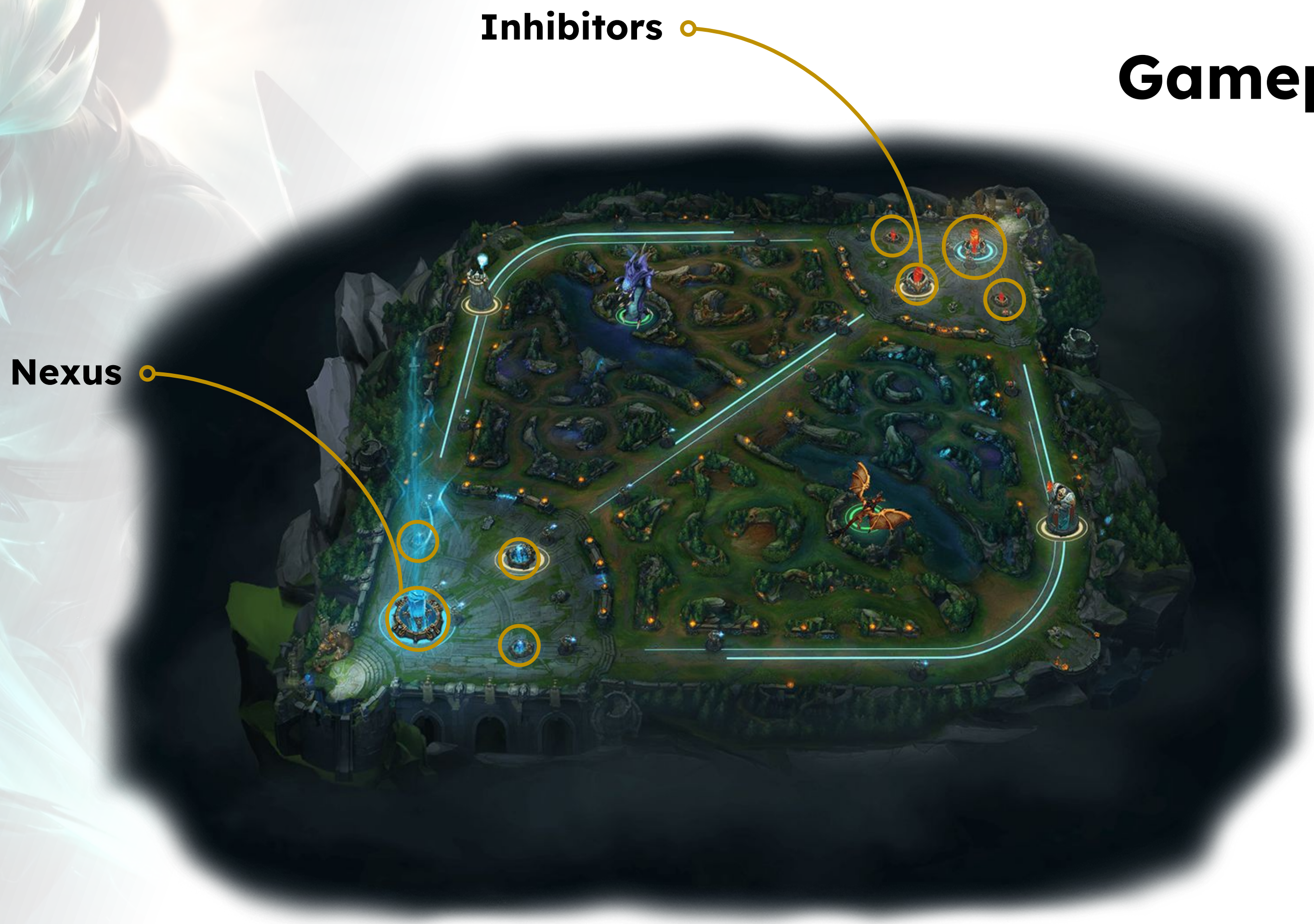
# Gameplay



# Gameplay



# Gameplay



# Problem Formulation

## Can we derive strategies for the game?

By observing trends in dataset

## Can we predict the game's outcome?

Using variables in the dataset

# Project Objective

## Derive Strategies

By exploratory analysis of the data set

## Prediction of Outcome

Using various machine learning models

# Our Dataset

## KAGGLE

League of Legends -By Chuck Ephron

https://www.kaggle.com/datasets/chuckephron/leagueoflegends

# Our Dataset

- LeagueofLegends.csv — Main file containing data of 7620 competitive matches
- _columns.csv
- bans.csv
- gold.csv
- kills.csv — Contains data of every kill in the 7620 matches
- matchinfo.csv
- monsters.csv
- structures.csv — Contains data of every structures in the 7620 matches

# Our Dataset

Competitive matches from

**2014 - 2018**

# EXPLORATORY ANALYSIS

## 01 Game Length

Time taken for a match to end.

## 02 Objectives

Analysis of side objectives.

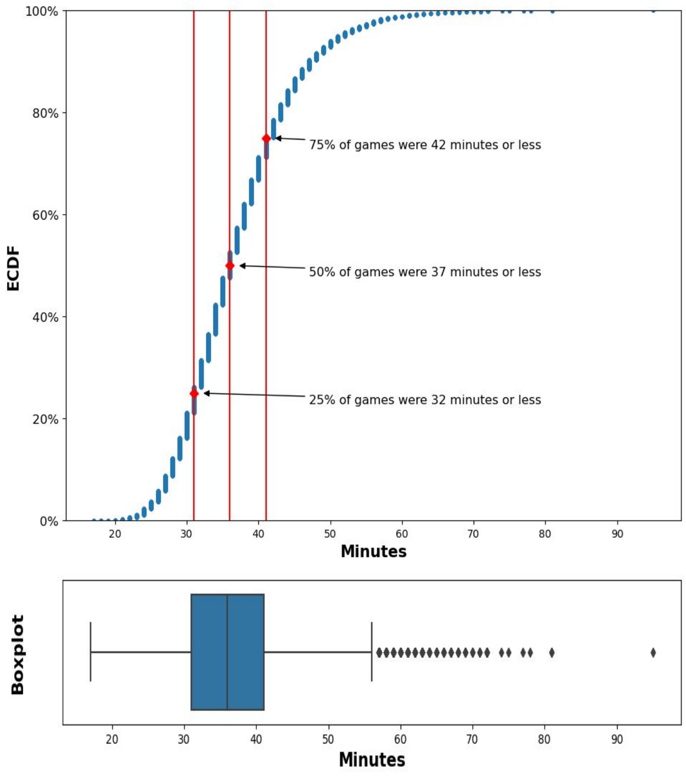
## 03 Kill Analysis

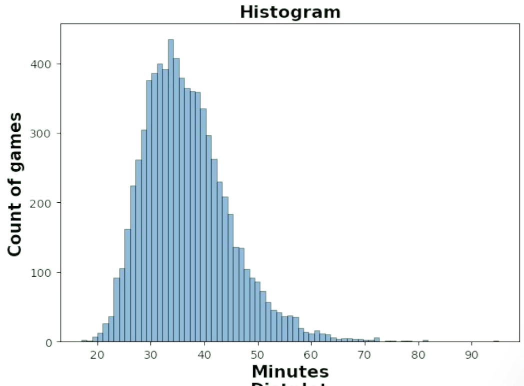
Analysing the defeats of the players.

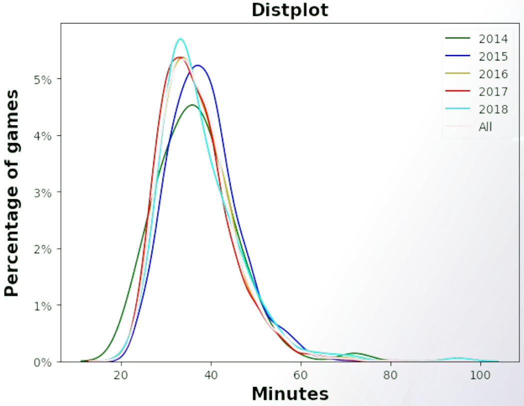
## 04 Champions

Usage of characters that one can play as.

# Game Length



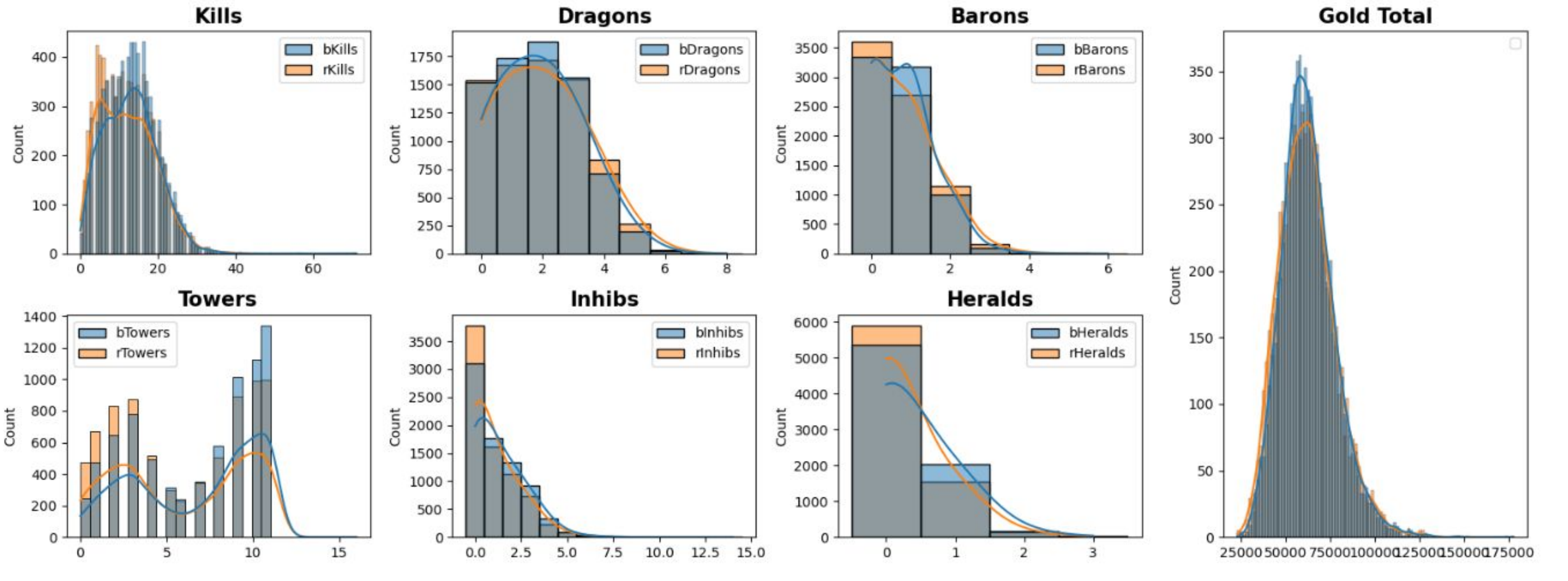




We are able to conclude that most games are typically around 30-40 minutes.

1st Quartile = 32 min
Median = 37 min
3rd Quartile = 42 min

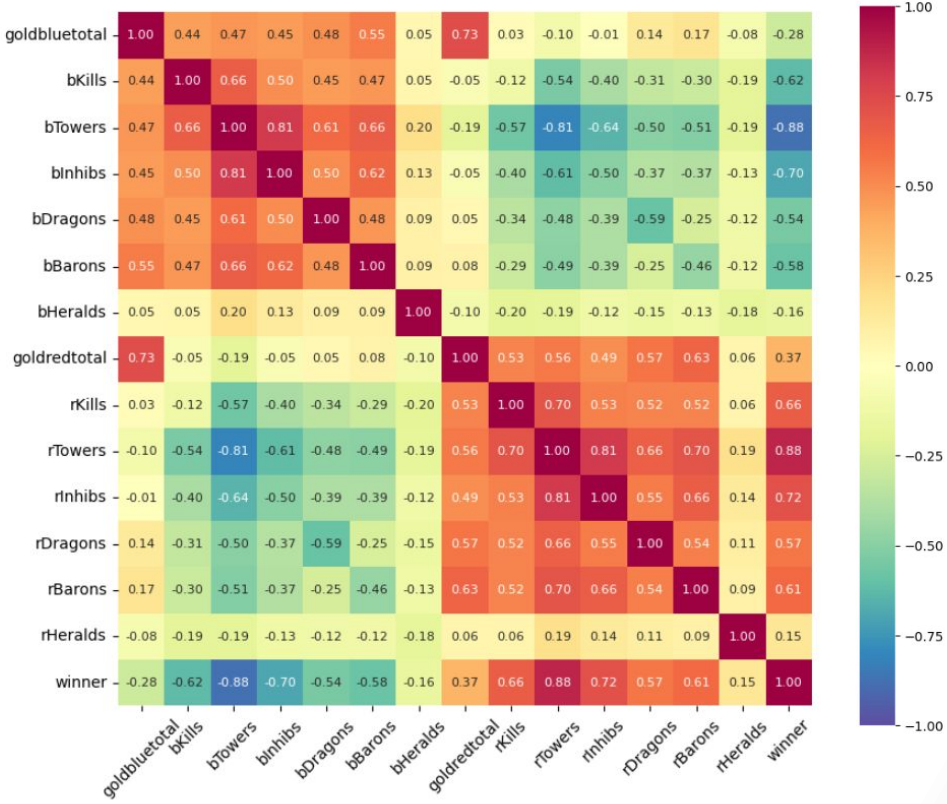
Longer than 57 min would already be considered an outlier.

# Objectives




A collection of histograms for the objectives.

# Objectives - Correlation Matrix




After a correlation matrix is created, we can conclude that the objectives that correlate the most with winning would be:

Tower Takedowns: 0.88
Inhibitor Takedowns: 0.71
Enemies Killed: 0.64
Baron Kills: 0.60
Dragon Kills: 0.56

# Objectives - Hypothesis Testing

## T-test

**Null hypothesis:** Independent sample means are equal, i.e. variable does not affect outcome

**Alternate hypothesis:** Independent sample means are not equal, i.e. variable affects outcome

**Significance Level:** 0.05

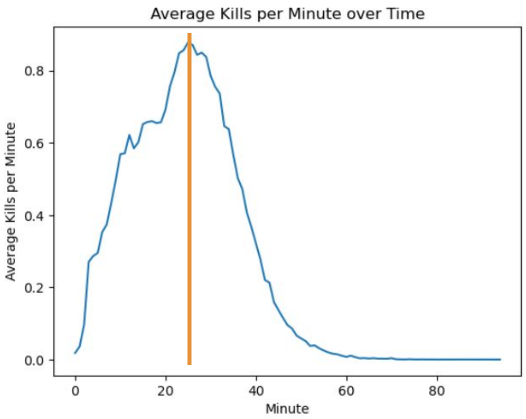
# Objectives - Hypothesis Testing

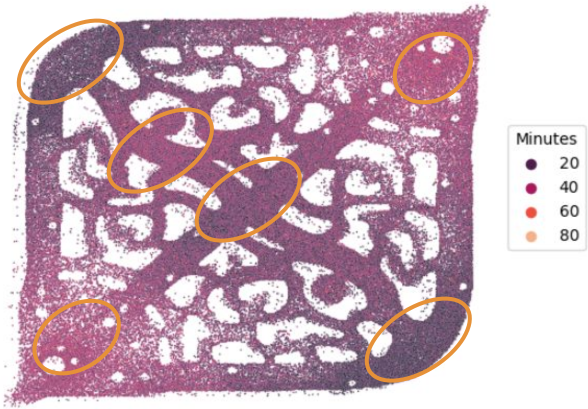
| | variance_ratio | t_statistics | p_value |
|---|---|---|---|
| goldbluetotal | 0.750019 | 4.173006 | 4.059590e-03 |
| bKills | 0.905055 | 11.253983 | 7.644119e-15 |
| bTowers | 0.714765 | 25.843247 | 2.011317e-50 |
| bInhibs | 1.175400 | 14.322101 | 5.557657e-18 |
| bDragons | 1.159410 | 9.237539 | 2.507444e-10 |
| bBarons | 1.202765 | 10.439995 | 2.462910e-11 |
| bHeralds | 1.250170 | 2.271230 | 1.063136e-01 |
| goldredtotal | 1.270327 | -5.813249 | 9.248785e-05 |
| rKills | 1.230237 | -12.545939 | 3.544069e-14 |
| rTowers | 1.393368 | -27.342079 | 1.204281e-54 |
| rInhibs | 1.292470 | -14.677361 | 2.610721e-15 |
| rDragons | 1.267668 | -9.874751 | 2.195461e-11 |
| rBarons | 1.225391 | -11.060842 | 1.053633e-10 |
| rHeralds | 1.181457 | -2.183298 | 1.158978e-01 |

> 0.05

> 0.05

# Kill Timing/Density Map

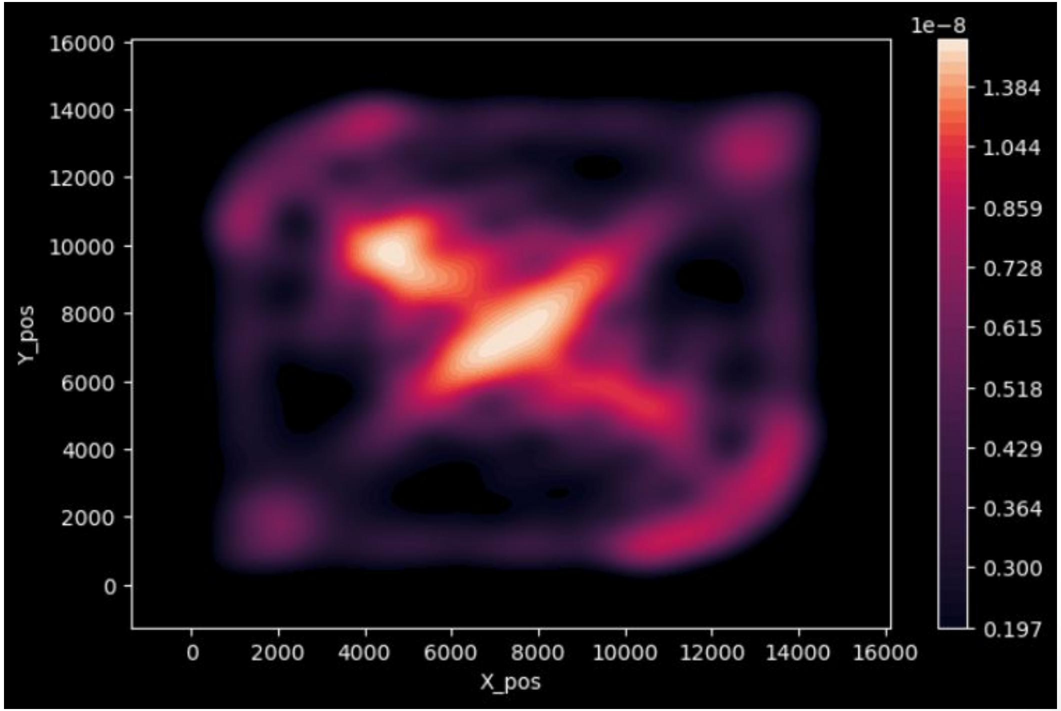



We can conclude that most kills are done around the between the 20 to 30 minute mark.

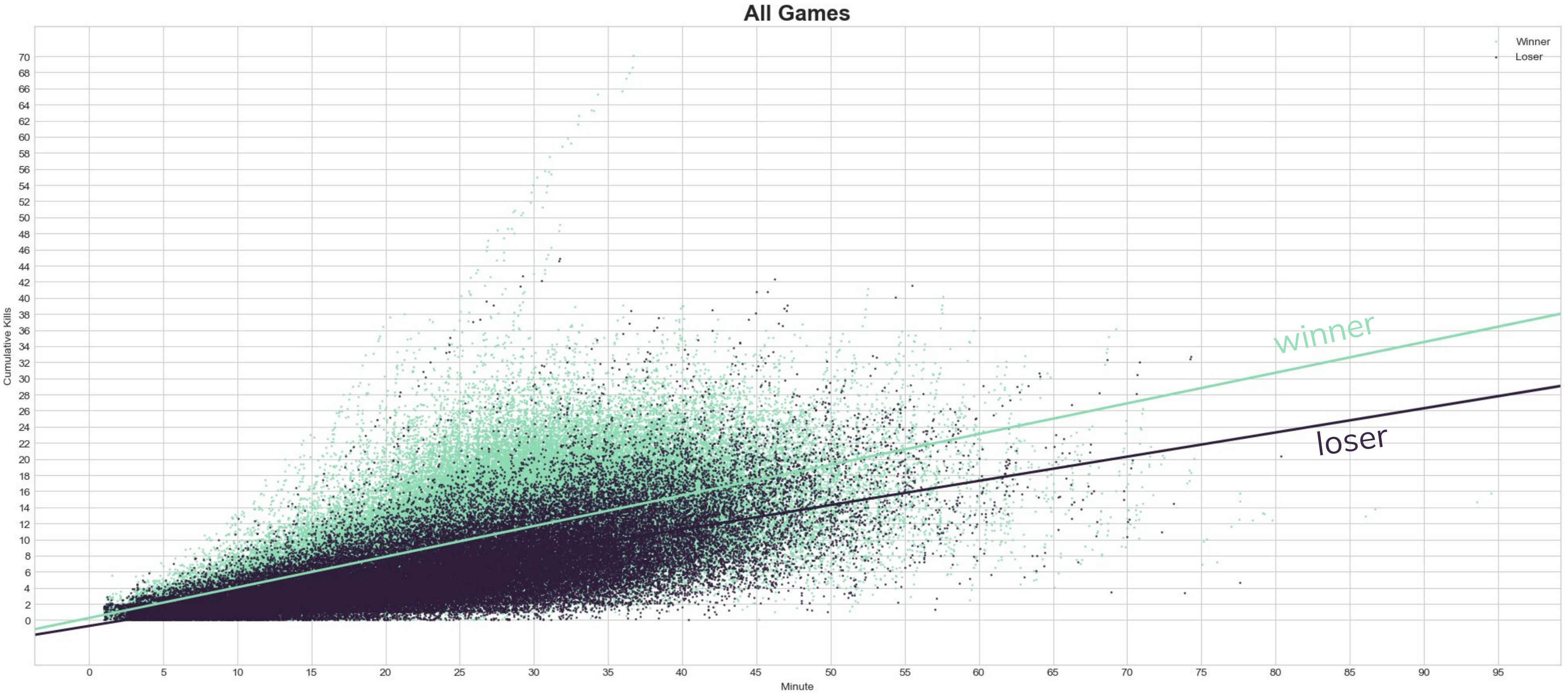
Later kills are done closer to the top right and bottom left corners of the map, which are where the nexus are located.

# Kill Density Map

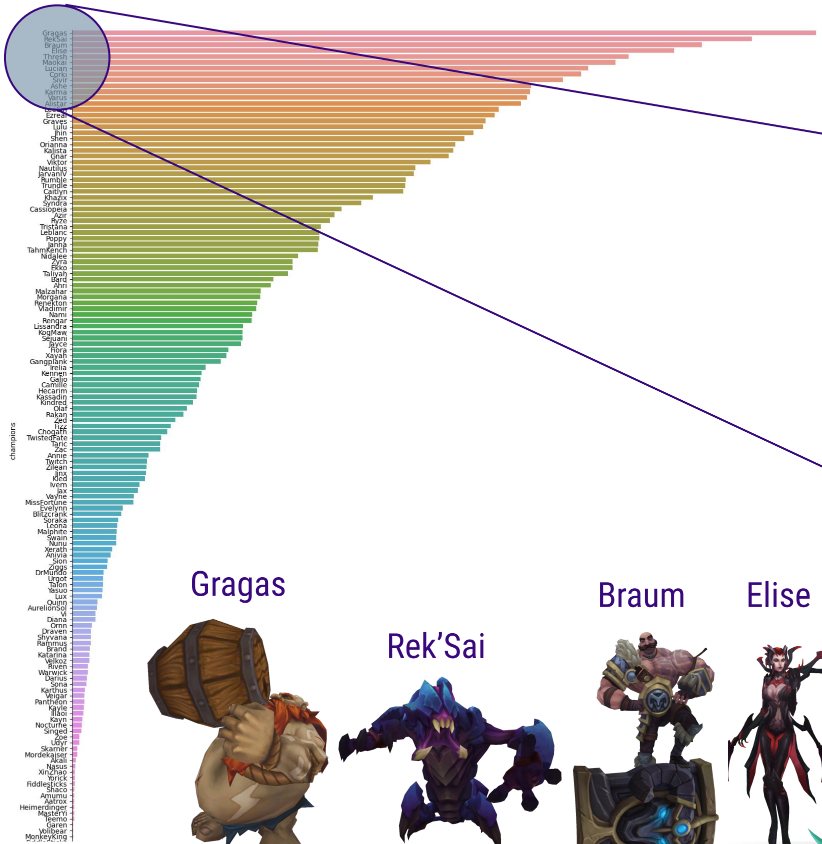



A kill density map shows that most kills are concentrated near the center, as well as in each corner of the map.

# Cumulative Kills



# Most Popular Champions




The top 10 most chosen champions, along with the exact number of uses were the following:

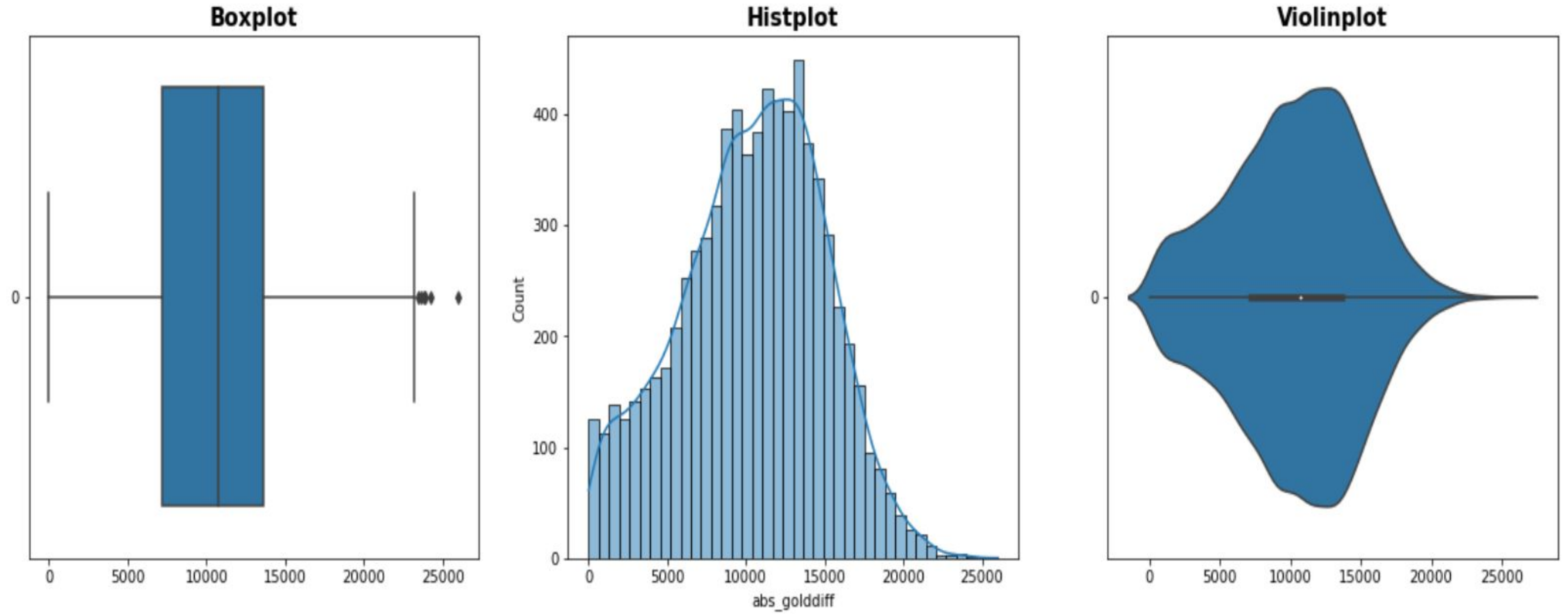

2551
2330
2158
2063
1908
1863
1768
1744
1682
1573

# Most Popular Champions Chosen for Each Role

| Top Lane | Jungler | Mid Lane | ADC | Support |
| --- | --- | --- | --- | --- |
| #1. Maokai | #1. RekSai | #1. Orianna | #1. Sivir | #1. Braum |
| #2. Gnar | #2. Gragas | #2. Victor | #2. Lucian | #2. Thresh |
| #3. Shen | #3. Elise | #3. Syndra | #3. Ashe | #3. Alistar |


Top Lane - Maokai
Mid Lane - Orianna
Jungler - Rek'sai
ADC - Sivir
Support - Braum

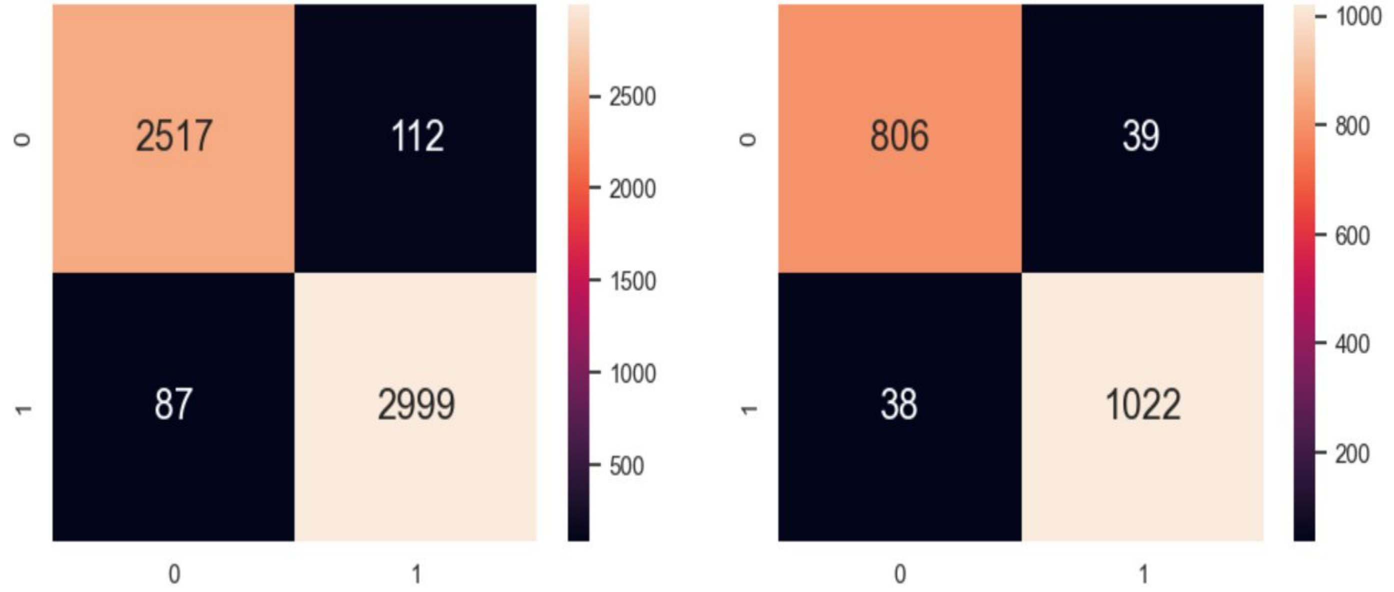
# Gold Analysis



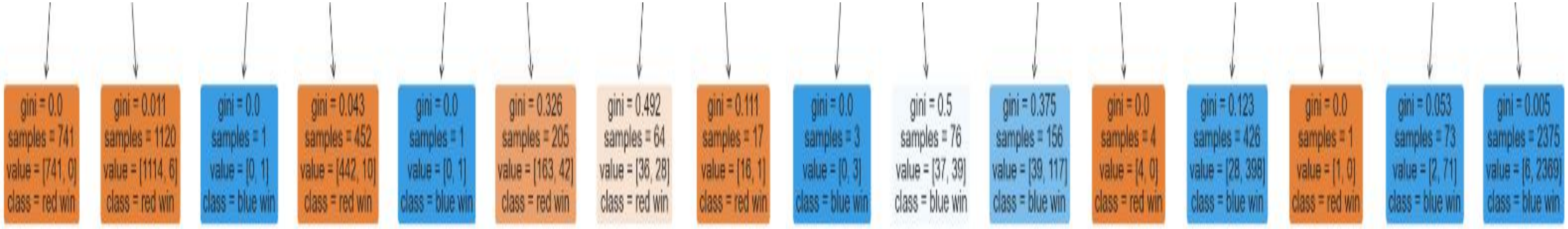

Now, we can see the actual distribution of gold diff. It seems to have a right-skewed distribution.

```
Goodness of Fit of Model        Train Dataset
Classification Accuracy         : 0.9651793525809274

Goodness of Fit of Model        Test Dataset
Classification Accuracy         : 0.9595800524934384
```

Out[ ]: <AxesSubplot: >



```python
In [ ]:
# Plot the trained Decision Tree
from sklearn.tree import plot_tree

f = plt.figure(figsize=(40,40))
plot_tree(dectree, filled=True, rounded=True,
          feature_names=["golddiff"],
          class_names=["red win","blue win"])
```



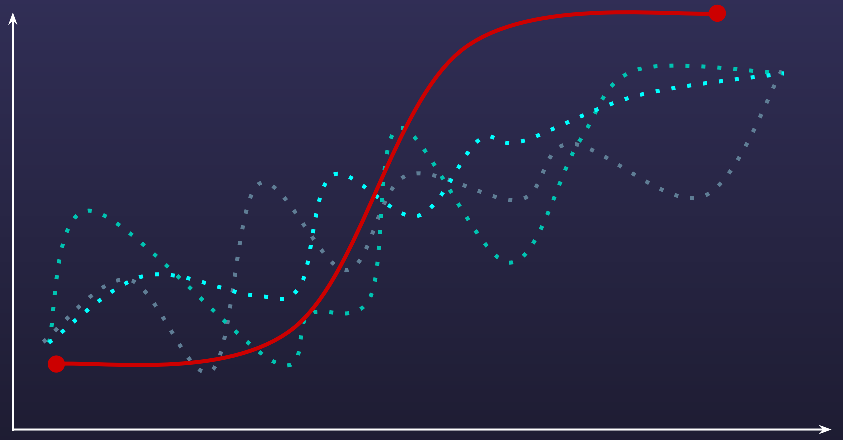
Out[ ]:

| | bResult | finalgold |
|---|---|---|
| **bResult** | 1.000000 | 0.894409 |
| **finalgold** | 0.894409 | 1.000000 |

Prediction by

# Logistic Regression

A model to predict binary outcome using a independent variable.



# Data Cleaning

rKills, bKills:

Time of each event

[0, 3.44, 5.45, 7.93, 8.38.....]

Whether the team won

Cumulative kill

| | Address | type | winner | counts | value | won | bin |
|---|---|---|---|---|---|---|---|
| 0 | http://matchhistory.na.leagueoflegends.com/en/... | bKills_time | 1 | 1 | 10.820 | True | (10, 15] |
| 1 | http://matchhistory.na.leagueoflegends.com/en/... | bKills_time | 2 | 1 | 11.104 | False | (10, 15] |
| 2 | http://matchhistory.na.leagueoflegends.com/en/... | bKills_time | 1 | 1 | 5.255 | True | (5, 10] |
| 3 | http://matchhistory.na.leagueoflegends.com/en/... | bKills_time | 2 | 1 | 8.274 | False | (5, 10] |
| 4 | http://matchhistory.na.leagueoflegends.com/en/... | bKills_time | 1 | 1 | 11.438 | True | (10, 15] |

# Model Accuracy

Kills

Accuracy: **64%**

Towers

Accuracy: **71%**

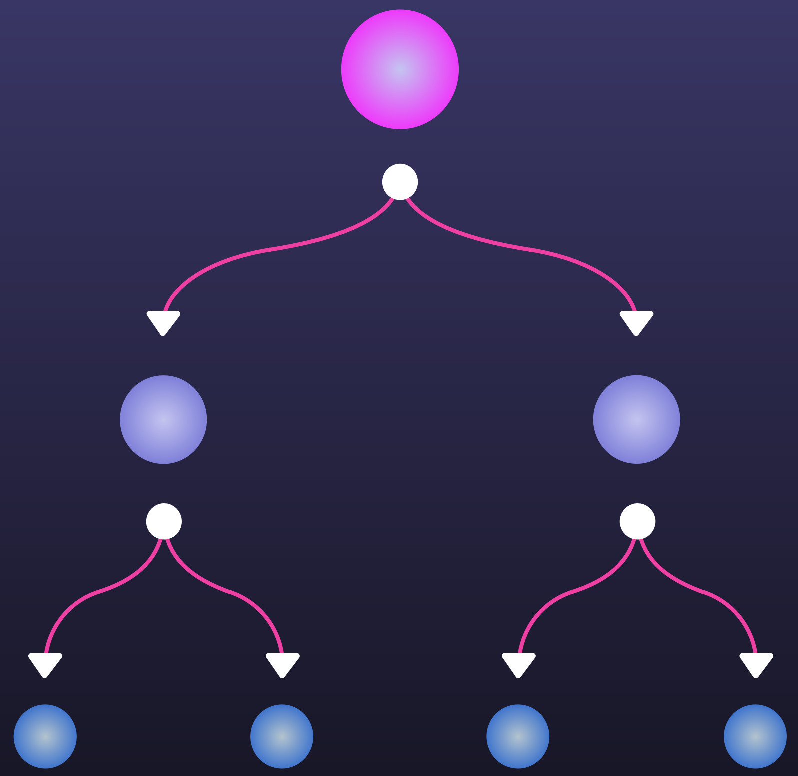
## Prediction by

# Classification Tree

We will create the classification trees the timings of the first time an objective is completed.

| | win | firstTower | firstDragon | firstBaron | firstKill | firstInhib |
|---|---|---|---|---|---|---|
| 0 | True | 27.542 | 37.267 | None | 10.82 | 36.686 |
| 1 | False | 39.23 | 17.14 | 29.954 | 16.529 | None |
| 2 | True | 19.257 | 12.264 | None | 12.387 | 36.813 |
| 3 | False | 23.239 | 32.545 | 29.255 | 11.104 | None |
| 4 | True | 15.045 | 24.577 | 35.144 | 5.255 | 37.511 |
| ... | ... | ... | ... | ... | ... | ... |
| 15235 | False | None | None | None | 13.534 | None |
| 15236 | True | 22.043 | 12.083 | 25.777 | 3.055 | 33.943 |
| 15237 | False | 32.793 | None | None | 13.399 | None |
| 15238 | True | 32.298 | 35.963 | 26.427 | 10.327 | 37.152 |
| 15239 | False | 18.042 | 22.787 | None | 7.68 | None |

15240 rows × 6 columns

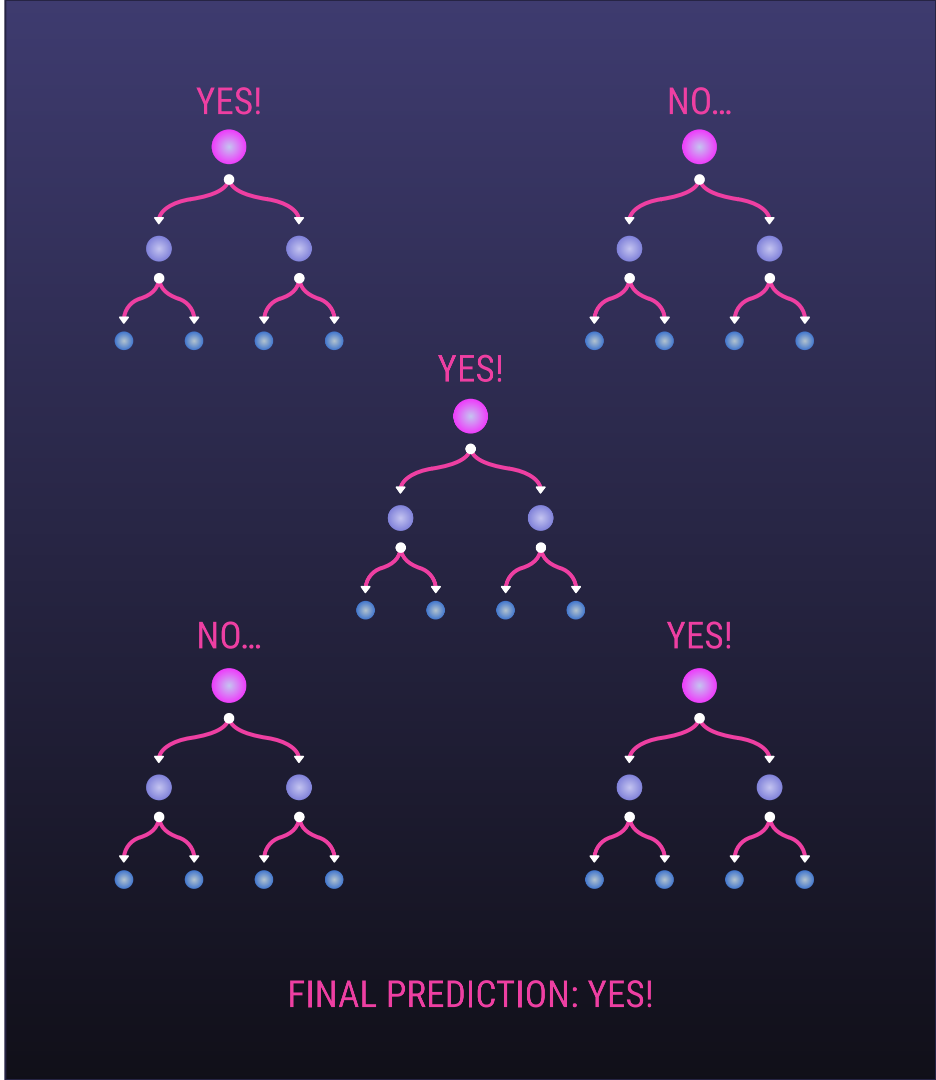


### Going Further....

# Random Forest

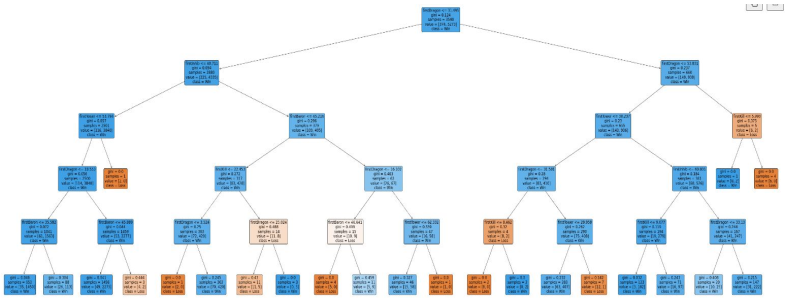
A process where we gather a number of randomly selected classification trees from a given dataset,

And a prediction will be made by a majority vote by all of these trees.

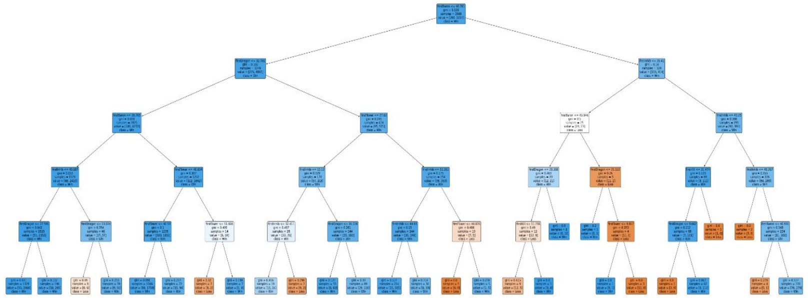


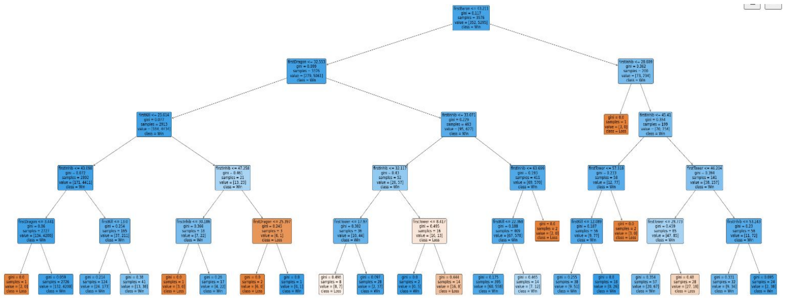
# Created Forests

5 forests for each objective + 1 forest combining all.
Each forests has 400 trees of depth 5.



3 sample trees from the same forest.





# Created Forests

Here are the classification accuracies.

Let’s try predicting a match with these.

| Forest | Classification Accuracy |
|---|---|
| firstTower | ~0.66 |
| firstInhib | ~0.94 |
| firstBaron | ~0.84 |
| firstDragon | ~0.68 |
| firstKill | ~0.57 |
| combined | ~0.95 |

Predicting

# 2016 World Championship Finals

## Actual

| SKT | SSG |
| --- | --- |
| 3 | 2 |

## Random Forests

First dragon takedown

| SKT | SSG |
| --- | --- |
| 0 | 5 |

First tower takedown / First baron takedown / First kill

| SKT | SSG |
| --- | --- |
| 2 | 3 |

First inhibitor takedown / All data combined

| SKT | SSG |
| --- | --- |
| 3 | 2 |

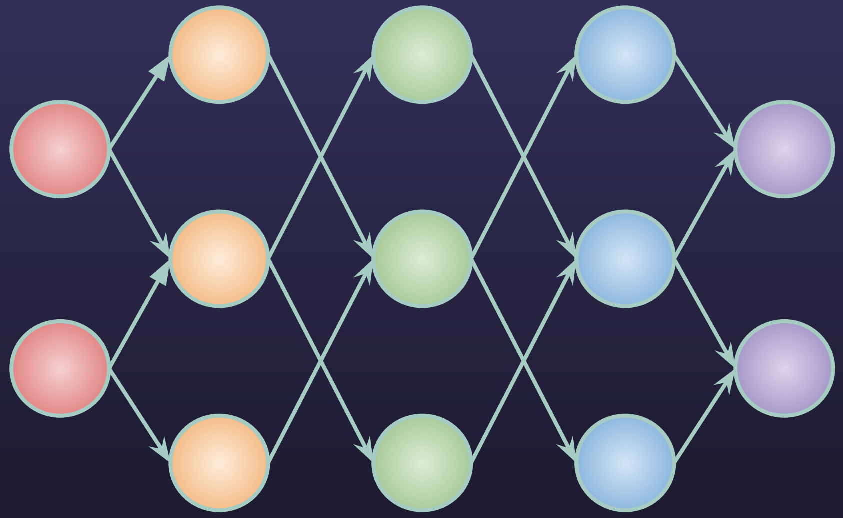
## Prediction by

# Recurrent Neural Network

RNN is a class of neural networks that takes in a time-series dataset, and executes a feedback loop which considers the output from a previous iteration.

In our case, we will be using the sequential data of the 6 predictors we had earlier, across 30 minutes to feed into the RNN model using the Cross-entropy loss function and Stochastic gradient descent optimiser.

We will be using Pytorch RNN model for this.



# Data Cleaning

Blue: Gold Towers Inhibs Kills Dragons Barons

Red: Gold Towers Inhibs Kills Dragons Barons

## Data Cleaning

Time of each event

[0, 3.44, 5.45, 7.93, 8.38.....]

Counts of events at each minute

$$\underbrace{[0, 0, 0, 1, 1, 2, 2, 2, 3, 4, 5, 5, ........]}_{\text{30 minutes}}$$

# Data Cleaning

Counts of events at each minute of Blue Team

[0, 0, 0, 1, 1, 2, 2, 2, 3, 4, 5, 5, ........]

Counts of events at each minute of Red Team

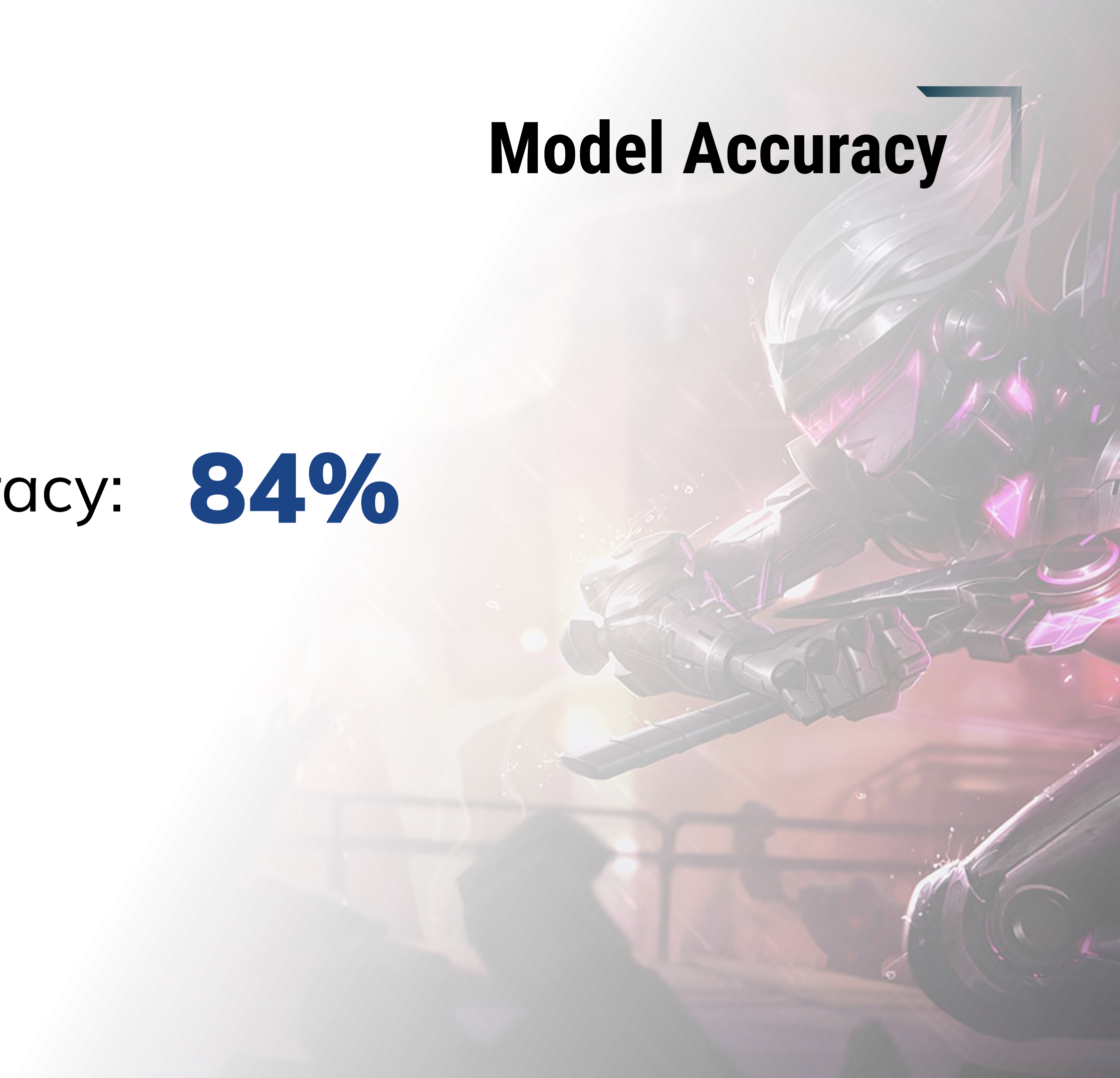
[0, 0, 1, 2, 2, 2, 3, 3, 3, 4, 7, 8, ........]

Difference of counts of events at each minute

[0, 0,-1,-1,-1, 0,-1,-1,0,0,-2,-3, ........]

# Model Accuracy

Accuracy: **84%**



Predicting

# 2016 World Championship Finals

## Neural Network

| SKT | SSG |
| --- | --- |
| 4 | 1 |

## Actual

| SKT | SSG |
| --- | --- |
| 3 | 2 |

# Possible Strategies

## Early Kills

Getting early 1-2 early kills before 3 minutes translates to a significant increase in chance of winning. However, getting more kills after the first 2 only yields marginal increase in chance of winning.

## Early Dominance

Early kills and advantage seems to snowball into later parts of the game. Hence it might be wiser to choose a team composition which can secure early advantages

# Conclusion

## ACCURACY

As we've seen, prediction models are never 100% accurate, and should be taken with a grain of salt, as there is only so much data a model can utilize.

## UNPREDICTABILITY

That being said, we must remember that a League Of Legends match is often unpredictable. As a skill-based game, player's skills are hard to express in a dataset.

# THANK YOU!

Qian Jianheng Oscar (U2220109K)
Dimas Rizky (U2221229F)
Jain Mokshit (U2222438K)