

# *Predict the Winner in League of Legends using GANs*

Software Project Neural Networks

Supervisor:

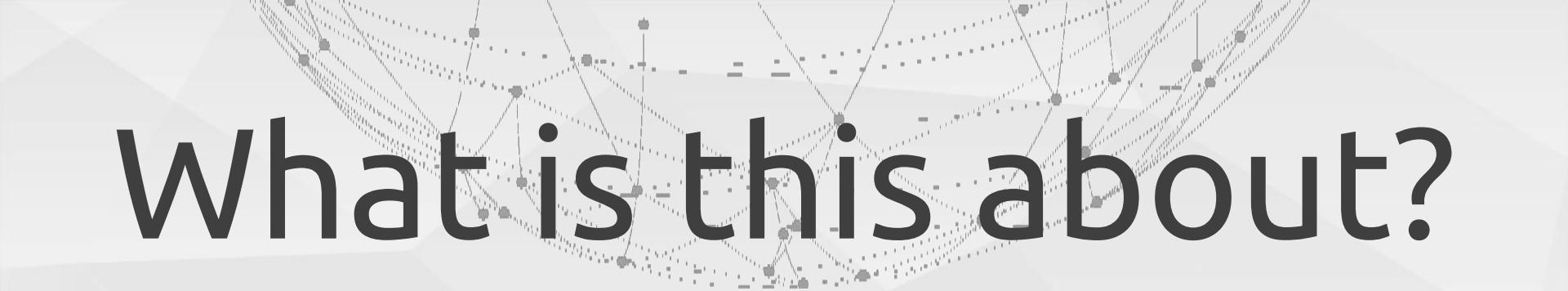
*Prof. Dr. Dietrich Klakow*

Participants:

*Koushik Chowdhury 2572865*

*Matteo Bonaker 2564542*

2019-03-28



# what is this about?

imagine this situation ...

# what is this about?

... 100 bored people sitting at home ...

# What is this about?



... they all want to go on an (online) adventure ...



# What is this about?

... so they start a game called



# What is this about?

... the game makes equal groups of 10 ...

# what is this about?



... and divides them into two teams.

# what is this about?

Middle



Bottom



Jungle



Support



Top



Everyone is assigned a **role**.



Support



Jungle



Middle



Top



Bottom

# what is this about?

Middle



Bottom



Jungle



Support



Top



Now the game tells everyone  
to select their champion:



Support



Jungle



Middle



Top

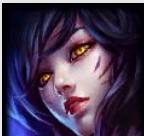


Bottom



# What is this about?

Middle



Bottom



Jungle



Support

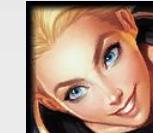
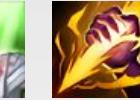


Top

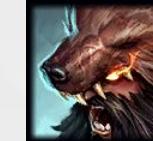


Also everyone selects

2 spells:



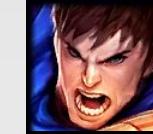
Support



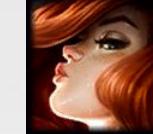
Jungle



Middle



Top



Bottom

# what is this about?

Middle							Support
Bottom							Jungle
Jungle							Middle
Support							Top
Top							Bottom

# what is this about?

... now the game loads the 3D environment ...



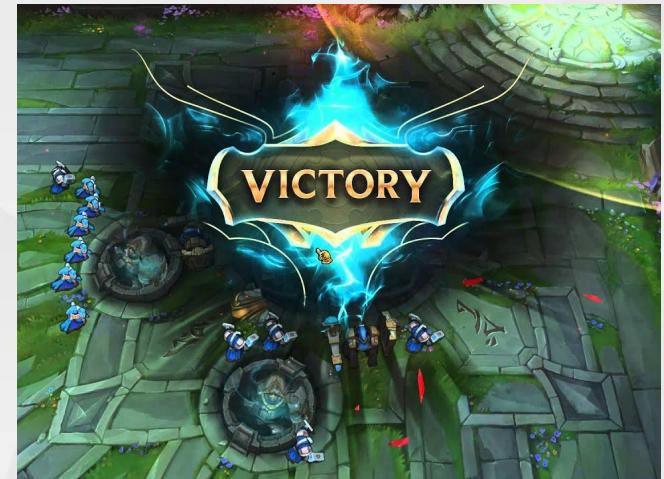
# what is this about?



# What is this about?

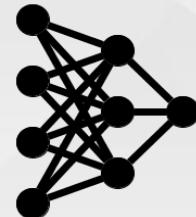
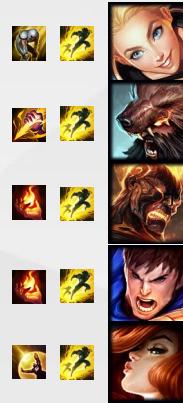
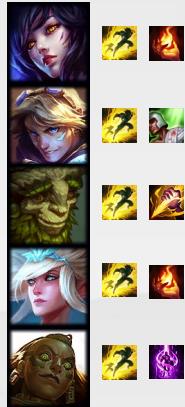
To win, participants will:

- **Need to** destroy the one enemy *Nexus*
- **Need to** destroy at least 5 *Turrets*
- **Need to** destroy at least 1 *Inhibitor*
- **Often** kill other *Champions*
- **Often** kill the *Dragon*
- **Often** kill the *Baron*

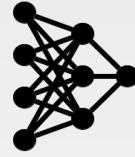
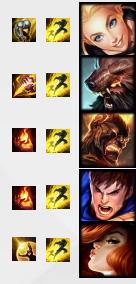
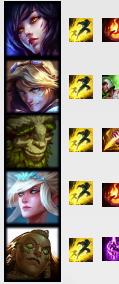


# what is this about?

Long story short:



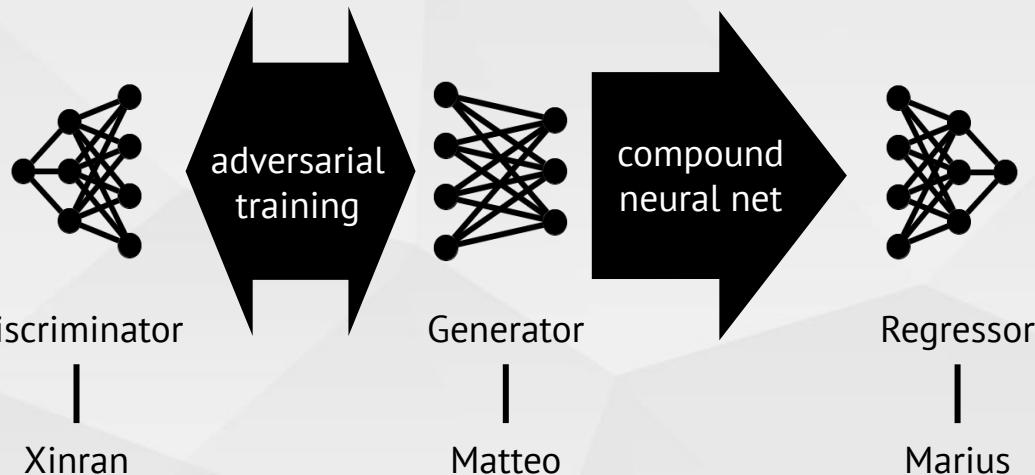
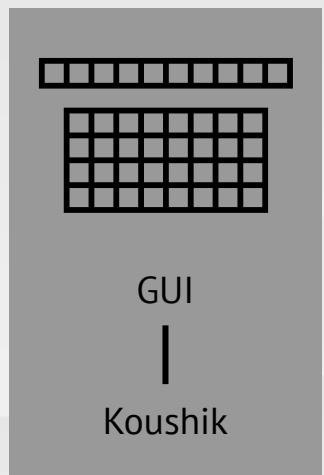
# what is this about?



**Highest accuracy with this approach: ~0.58**

**Highest accuracy using LM: ~0.55**

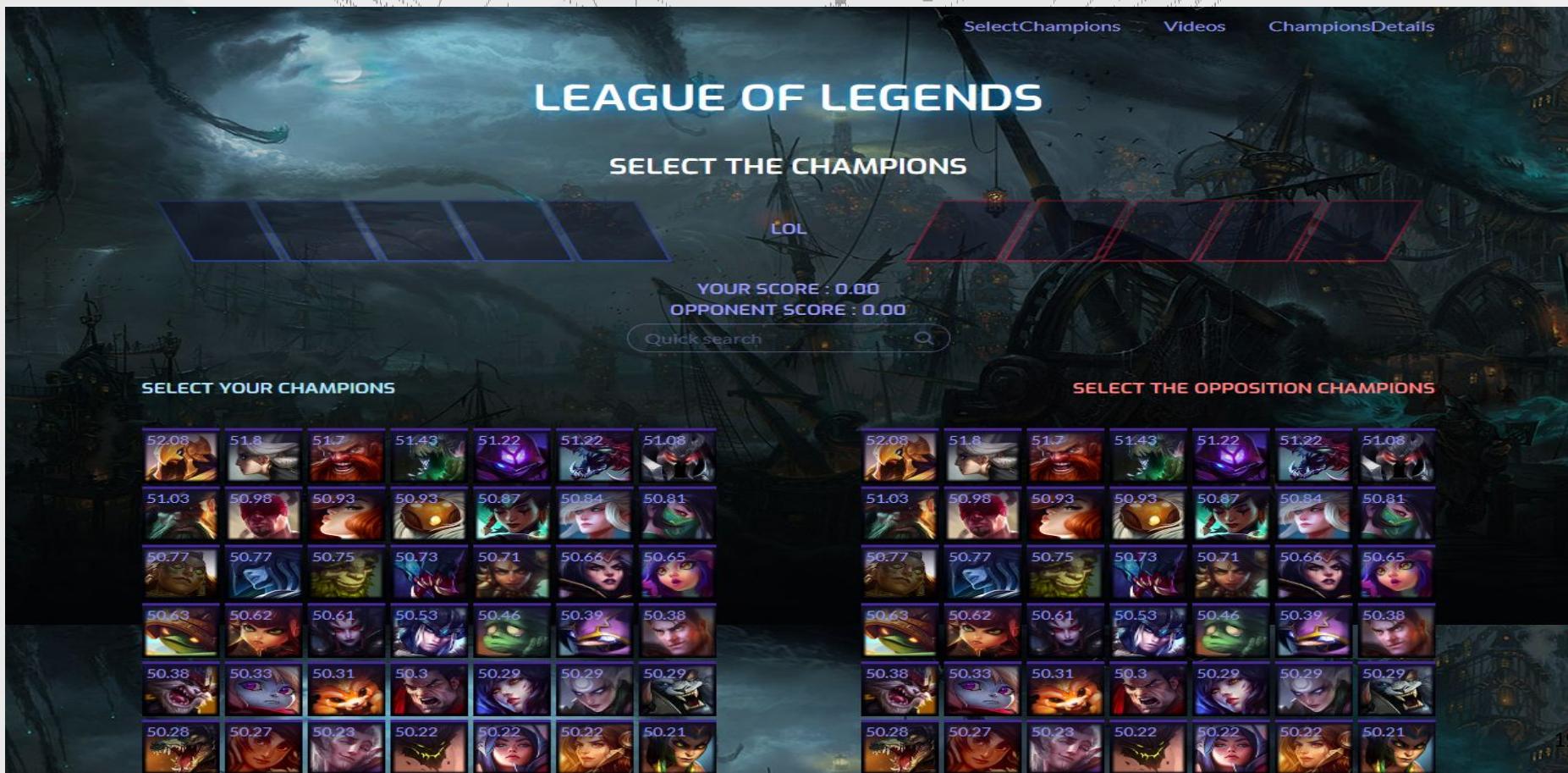
# So now we did this:-



# Json File.....

```
{"5":{"winrate":48.18},  
 "102":{"winrate":48.83},  
 "103":{"winrate":50.62},  
 "97":{"winrate":50.71}, "20":{"winrate":50.21},  
 "85":{"winrate":47.28}, "14":{"winrate":52.08},  
 "96":{"winrate":50.65}, "100":{"winrate":49.16},  
 "56":{"winrate":51.7}, "89":{"winrate":48.22},  
 "30":{"winrate":51.43}, "53":{"winrate":51.03},  
 "57":{"winrate":49.24}, "67":{"winrate":47.06},  
 "12":{"winrate":47.71}, "29":{"winrate":48.42},  
 "78":{"winrate":50.38}, "73":{"winrate":48.63},  
 "93":{"winrate":49.51}, "91":{"winrate":51.08},  
 "16":{"winrate":49.06}, "99":{"winrate":49.56},
```

# User Interface



The image shows the "Select Champions" screen from the League of Legends user interface. The background features a dark, atmospheric scene of a city at night with a full moon and a large mechanical structure.

**Top Navigation:** SelectChampions, Videos, ChampionsDetails

**Title:** LEAGUE OF LEGENDS

**Section Header:** SELECT THE CHAMPIONS

**Scoreboard:** YOUR SCORE : 0.00, OPPONENT SCORE : 0.00

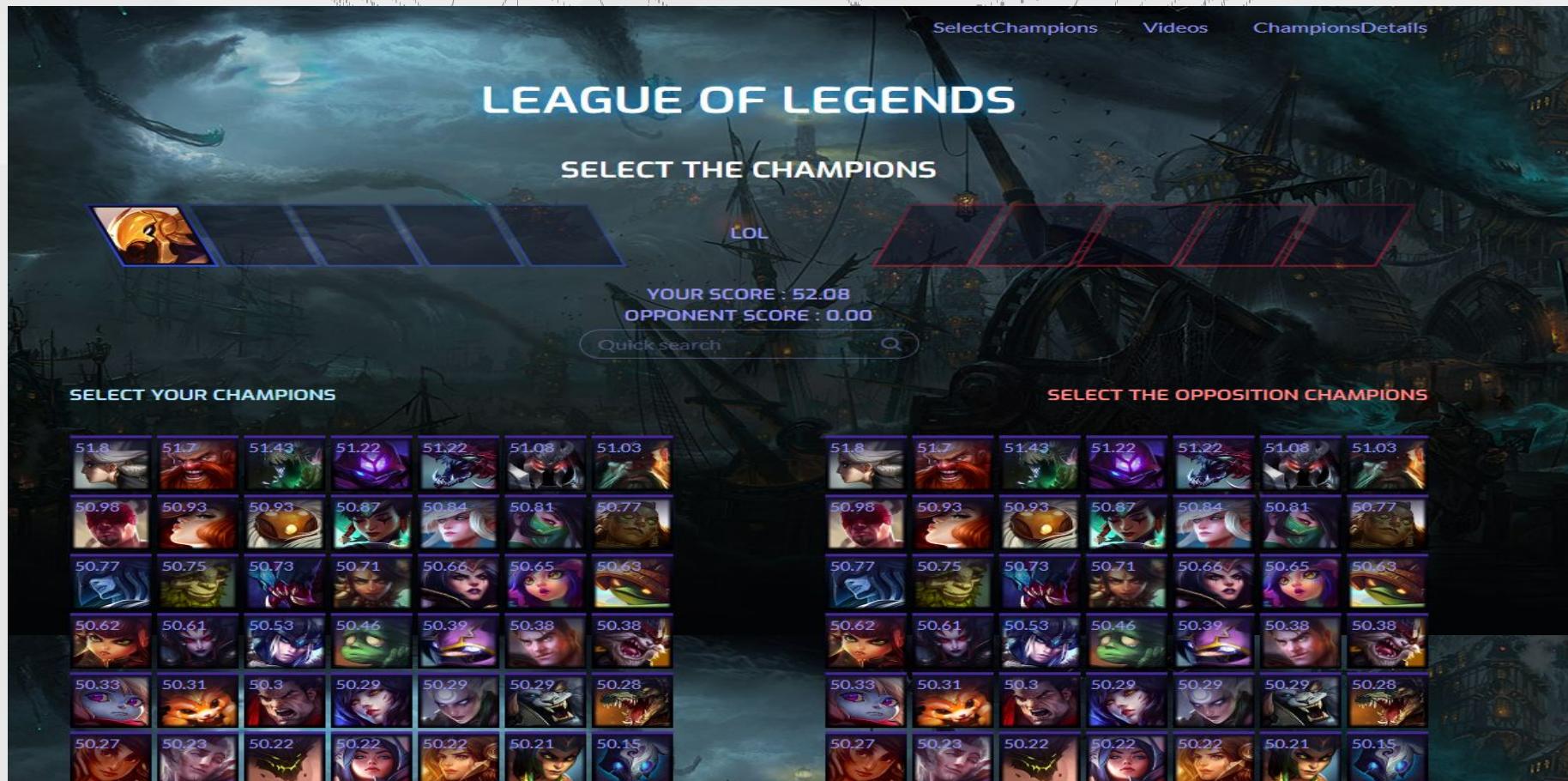
**Search Bar:** Quick search

**Champion Selection Grids:**

- SELECT YOUR CHAMPIONS:** A grid of 25 champion icons with numerical values below them. The values range from 50.21 to 52.08. The icons include: Teemo, Janna, Riven, Ahri, Kog'Maw, Ivern, Senna, Dr. Mundo, Leona, Rammus, Nautilus, Corki, Lux, Vayne, Kaisa, Ezreal, Jax, Akali, Nunu, Sivir, Thresh, Malphite, Zed, Lee Sin, and Soraka.
- SELECT THE OPPONITION CHAMPIONS:** A grid of 25 champion icons with numerical values below them. The values range from 50.21 to 52.08. The icons include: Teemo, Janna, Riven, Ahri, Kog'Maw, Ivern, Senna, Dr. Mundo, Leona, Rammus, Nautilus, Corki, Lux, Vayne, Kaisa, Ezreal, Jax, Akali, Nunu, Sivir, Thresh, Malphite, Zed, Lee Sin, and Soraka.

**Page Number:** 19

# User Interface (cont.)



The image shows the 'Select Champions' screen from League of Legends. The background features a dark, atmospheric scene of a city at night with a full moon. At the top center, the text 'LEAGUE OF LEGENDS' is displayed in large, white, block letters. Below it, the text 'SELECT THE CHAMPIONS' is centered. In the center of the screen, there is a blue rectangular area containing the text 'YOUR SCORE : 52.08' and 'OPPONENT SCORE : 0.00'. To the right of this score area is a search bar labeled 'Quick search' with a magnifying glass icon. On the left side, under the heading 'SELECT YOUR CHAMPIONS', there is a grid of 25 champion icons, each with a numerical value below it. The values range from 50.27 to 51.8. On the right side, under the heading 'SELECT THE OPPONITION CHAMPIONS', there is another grid of 25 champion icons with numerical values ranging from 50.15 to 51.03. The top navigation bar includes links for 'SelectChampions', 'Videos', and 'ChampionsDetails'. A small 'LOL' logo is located near the center of the screen.

YOUR SCORE : 52.08  
OPPONENT SCORE : 0.00

Quick search

SELECT YOUR CHAMPIONS

51.8	51.7	51.43	51.22	51.22	51.08	51.03
50.98	50.93	50.93	50.87	50.84	50.81	50.77
50.77	50.75	50.73	50.71	50.66	50.65	50.63
50.62	50.61	50.53	50.46	50.39	50.38	50.38
50.33	50.31	50.3	50.29	50.29	50.29	50.28
50.27	50.23	50.22	50.22	50.22	50.21	50.15

SELECT THE OPPONITION CHAMPIONS

51.8	51.7	51.43	51.22	51.22	51.08	51.03
50.98	50.93	50.93	50.87	50.84	50.81	50.77
50.77	50.75	50.73	50.71	50.66	50.65	50.63
50.62	50.61	50.53	50.46	50.39	50.38	50.38
50.33	50.31	50.3	50.29	50.29	50.29	50.28
50.27	50.23	50.22	50.22	50.22	50.22	50.15

# User Interface (cont.)



The image shows the 'Select Champions' screen from League of Legends. The background features a dark, atmospheric scene of a city at night with a full moon. At the top center, the text 'LEAGUE OF LEGENDS' is displayed in large white letters. Below it, 'SELECT THE CHAMPIONS' is centered. In the upper left, there is a row of five champion icons: Teemo, Riven, Leona, Janna, and Kog'Maw. To the right of this row, the text 'YOUR SCORE : 50.74' and 'OPPONENT SCORE : 0.00' is shown. A 'Quick search' input field with a magnifying glass icon is positioned below the score. The main area is divided into two sections: 'SELECT YOUR CHAMPIONS' on the left and 'SELECT THE OPPONITION CHAMPIONS' on the right. Each section contains a grid of 25 champion icons, each with a numerical ID above it. The IDs for the left section are: 51.8, 51.43, 51.22, 51.22, 51.08, 51.03, 50.98; 50.93, 50.87, 50.84, 50.81, 50.75, 50.66, 50.65; 50.63, 50.62, 50.61, 50.53, 50.46, 50.38, 50.38; 50.33, 50.31, 50.3, 50.29, 50.29, 50.29, 50.28; 50.27, 50.23, 50.22, 50.22, 50.22, 50.21, 50.15. The IDs for the right section are: 51.8, 51.43, 51.22, 51.22, 51.08, 51.03, 50.98; 50.93, 50.87, 50.84, 50.81, 50.75, 50.66, 50.65; 50.63, 50.62, 50.61, 50.53, 50.46, 50.38, 50.38; 50.33, 50.31, 50.3, 50.29, 50.29, 50.29, 50.28; 50.27, 50.23, 50.22, 50.22, 50.22, 50.22, 50.21; 50.14, 50.13, 50.13, 50.13, 50.12, 50.11, 50.1.

SelectChampions Videos ChampionsDetails

# LEAGUE OF LEGENDS

## SELECT THE CHAMPIONS

LOL

YOUR SCORE : 50.74  
OPPONENT SCORE : 0.00

Quick search

SELECT YOUR CHAMPIONS

51.8	51.43	51.22	51.22	51.08	51.03	50.98
50.93	50.87	50.84	50.81	50.75	50.66	50.65
50.63	50.62	50.61	50.53	50.46	50.38	50.38
50.33	50.31	50.3	50.29	50.29	50.29	50.28
50.27	50.23	50.22	50.22	50.22	50.21	50.15
50.14	50.13	50.13	50.13	50.12	50.11	50.1

SELECT THE OPPONITION CHAMPIONS

51.8	51.43	51.22	51.22	51.08	51.03	50.98
50.93	50.87	50.84	50.81	50.75	50.66	50.65
50.63	50.62	50.61	50.53	50.46	50.38	50.38
50.33	50.31	50.3	50.29	50.29	50.29	50.28
50.27	50.23	50.22	50.22	50.22	50.22	50.21
50.14	50.13	50.13	50.13	50.12	50.11	50.1

# User Interface (cont.)

The screenshot shows the 'Select Champions' screen from the League of Legends game. At the top, there are three tabs: 'SelectChampions' (highlighted in blue), 'Videos', and 'ChampionsDetails'. Below the tabs, the title 'LEAGUE OF LEGENDS' is displayed in large white letters, followed by 'SELECT THE CHAMPIONS'.

On the left, a row of five champion icons is shown, each with a small 'LOL' logo below it. On the right, another row of five champion icons is shown, also with a small 'LOL' logo below them. In the center, the text 'YOUR SCORE : 50.74' and 'OPPONENT SCORE : 50.36' is displayed, along with a 'Quick search' input field and a magnifying glass icon.

The main area is divided into two sections: 'SELECT YOUR CHAMPIONS' on the left and 'SELECT THE OPPONITION CHAMPIONS' on the right. Each section contains a grid of 25 champion icons, each with a numerical score below it. The scores range from 50.11 to 51.8. The background of the interface features a dark, atmospheric scene of a city at night with flying kites.

Champion	Score																																																														
51.8	51.43	51.22	51.22	51.08	51.03	50.98	51.8	51.43	51.22	51.22	51.08	51.03	50.98	50.93	50.87	50.84	50.81	50.75	50.66	50.65	50.63	50.62	50.38	50.38	50.33	50.31	50.3	50.29	50.28	50.27	50.23	50.22	50.22	50.22	50.21	50.15	50.14	50.13	50.13	50.13	50.12	50.21	50.15	50.14	50.13	50.13	50.13	50.12	50.11	50.1	50.09	50.07	50.06	50.03	50.03	50.03	50.03	50.03	50.03	50.03	50.03	50.03	50.03

# Limitation

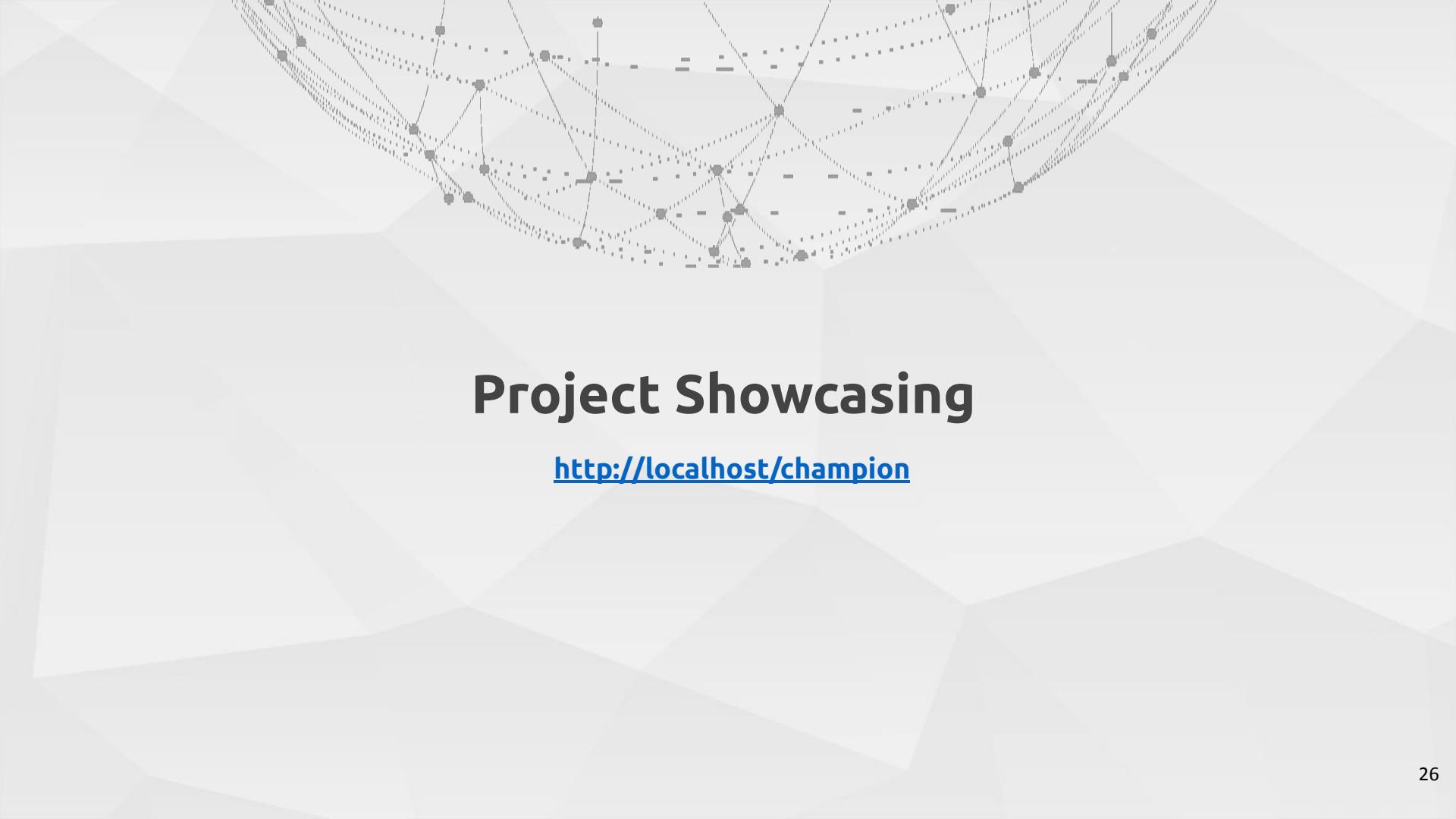
- Had no previous experience on League of Legends
- Application is running on Localhost.
- System cannot generate new winning accuracy for each champion after one champion is being selected.
- Not properly tested system, no automatic test script is being used to find the bugs.
- System is not well suited for marketing but with modification and time, it can be....

# Future Work

- Generate accuracy for each champion after one champion is being selected and so on.....
- System will suggest the best spells to the champion to win a game.

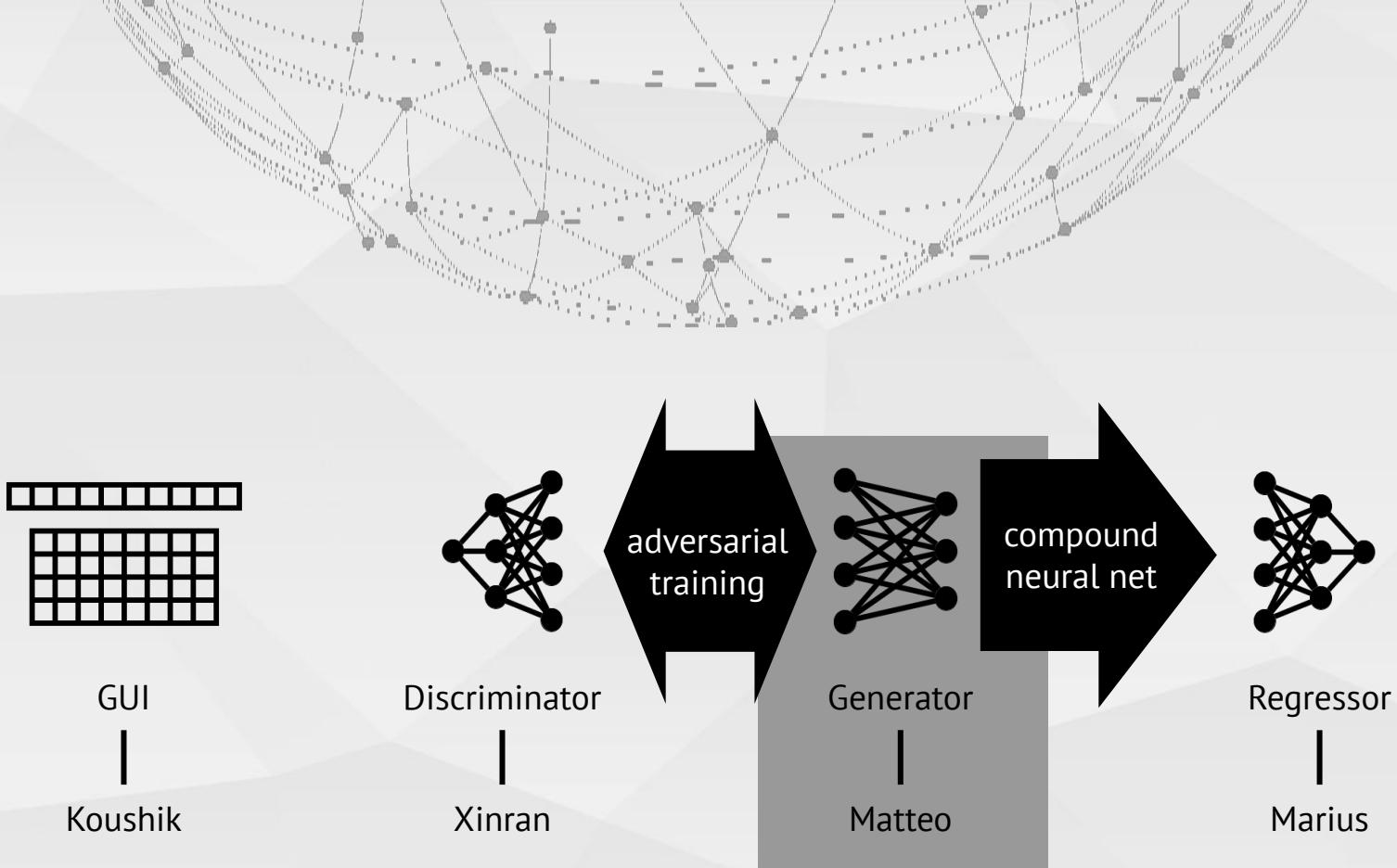
# Resources

- <https://euw.leagueoflegends.com/de/game-info/champions/>
- [https://fonts.googleapis.com/css?family=Lato:400,700|Exo:600"](https://fonts.googleapis.com/css?family=Lato:400,700|Exo:600)
- <https://cdnjs.cloudflare.com/ajax/libs/normalize/3.0.2/normalize.min.css>
- <https://www.heroescounters.com/>
- <https://cdnjs.cloudflare.com/ajax/libs/jquery/1.11.2/jquery.min.js>
- <https://cdnjs.cloudflare.com/ajax/libs/selectize.js/0.12.0/js/standalone/selectize.min.js>



# Project Showcasing

<http://localhost/champion>



# The Generator

## corpus - methodology - results - conclusion

### Acquisition:

- No publicly available corpora → need to crawl stuff
- Use the publicly available official League of Legends API
- Use one players game history to find more players
- Set filters:
  - Only in the current version of the game
  - Only on this specific type of game (ranked)
- Convert to usable format: CSV

# The Generator

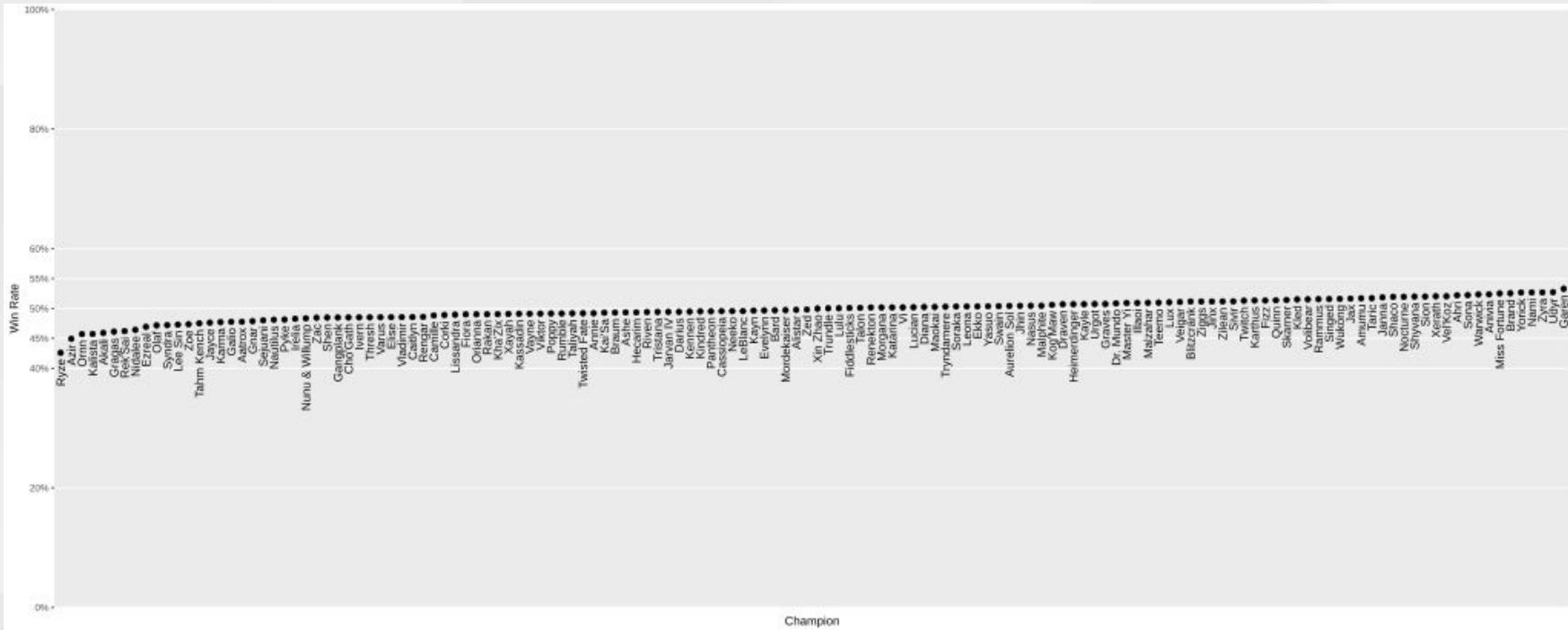
## corpus - methodology - results - conclusion

### Inspection:

- Mostly *results* of matches, not the flow of the game
  - Exception for every 10-minutes interval per participant:
    - CreepScore, DamageTaken, ExperiencePointsGained
  - Exception for some "firsts"
- Per Participant: ChampionID, Spell1ID, Spell2ID, Kills, Deaths, Gold, TurretKills, DamageHealed/-Taken/-Dealt/-Mitigated, etc.
- Per Team: DragonKills, BaronKills, TurretKills, InhibitorKills, Win
- Per Match: Time, Duration

# The Generator

## corpus - methodology - results - conclusion

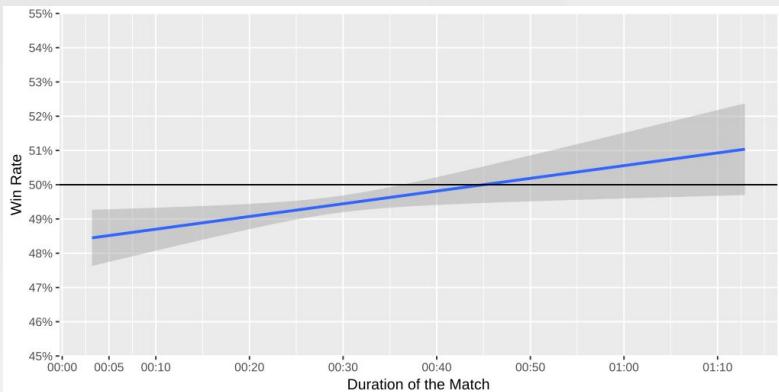


# The Generator

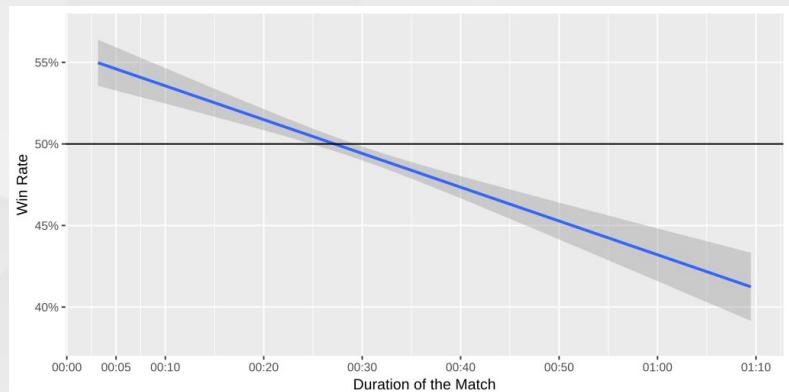
## corpus - methodology - results - conclusion

LM for the win rate by match duration

“Tristana”



“Pantheon”



# The Generator

## corpus - methodology - results - conclusion

### Data to feature conversion:

- Remove unnecessary data
  - gameId, gameMode, queueId, etc.
- Normalization
  - Use R to calculate all means, standard deviations etc.
  - Caveat: Reversing Normalization may lead to over- and underflows



# The Generator

## corpus - methodology - results - conclusion

- Use the “known”  
data as features to produce “unknown” data, except “win”
- Because of the big amount of features, use a csv file to describe them
- Use the data structure from the API
  - no “multiple hot”
  - no permutation resistance



# The Generator

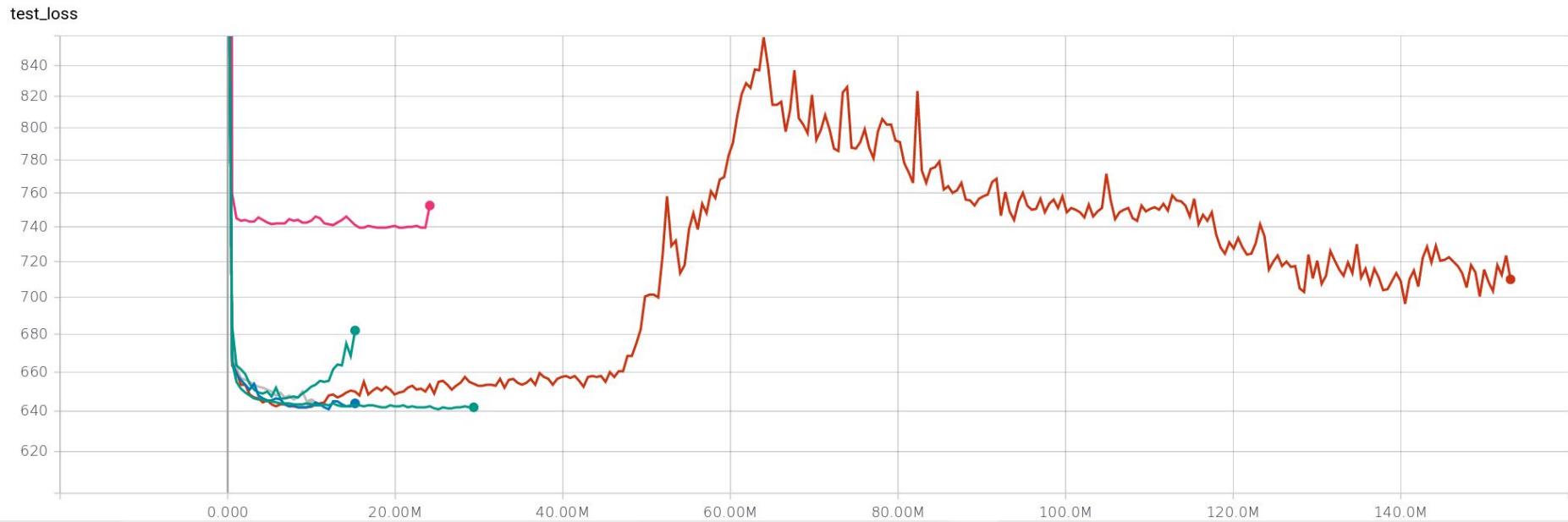
## corpus - methodology - results - conclusion

- **Simple FFN**
  - Optimized hyper parameters: Learning Rate, Batch Size, Hidden Layer Structure
  - Loss functions:
    - Cross entropy (Softmax for One-Hot and Sigmoid for Boolean)
    - MAE for continuous variables
  - Use multiple hot encoding for “bans”, still, for loss calculations
- Offer empty inputs (no information from the user)
  - Problematic for the discriminator as it does not have that requirement
    - we simply extended the corpus of real data for the discr.

# The Generator

## corpus - methodology - results - conclusion

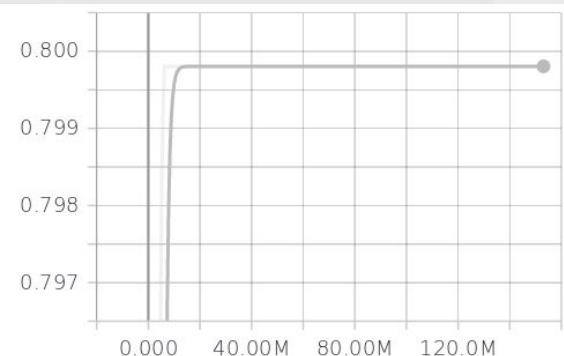
Accumulated loss (divide by 855 columns to get loss per column)



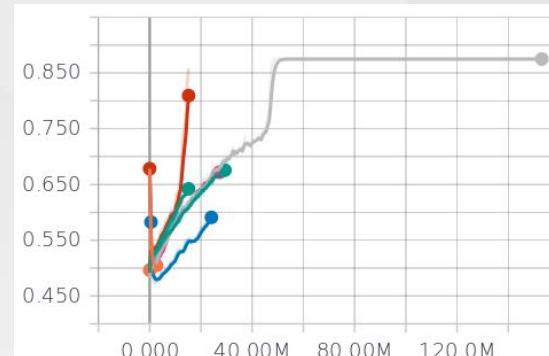
# The Generator

## corpus - methodology - results - conclusion

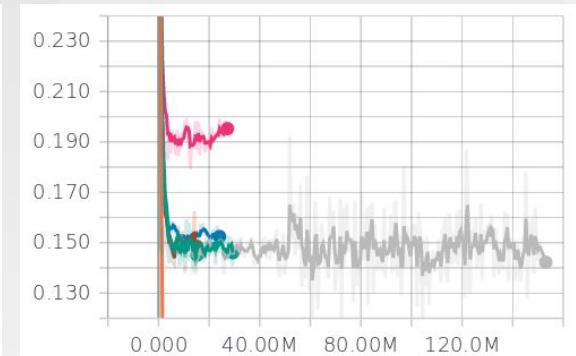
'lane' and 'role' overfit early:



'highesAchievedSeasonTier'  
also overfits:



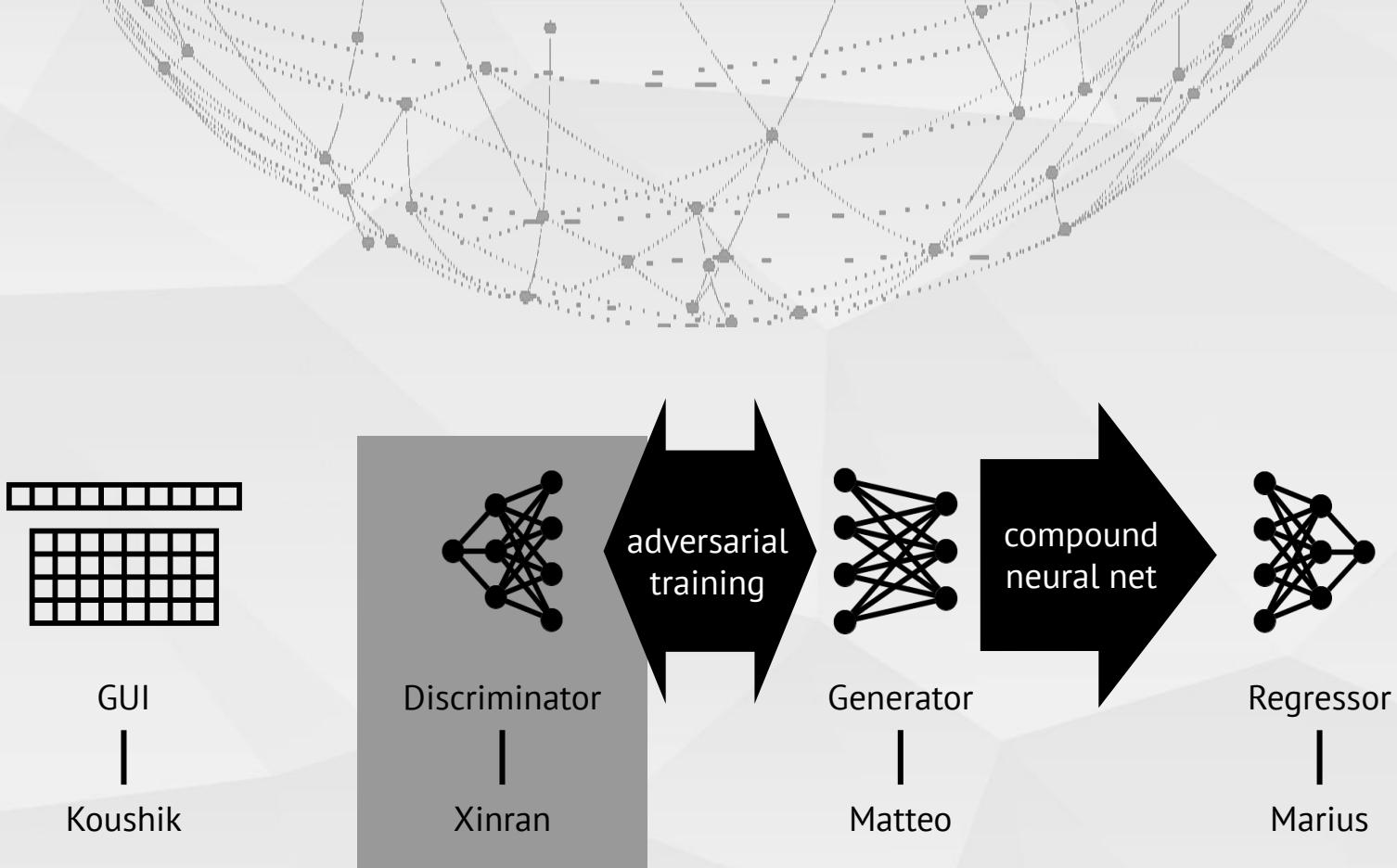
'firstTowerKill' does *not* overfit  
at the same time:

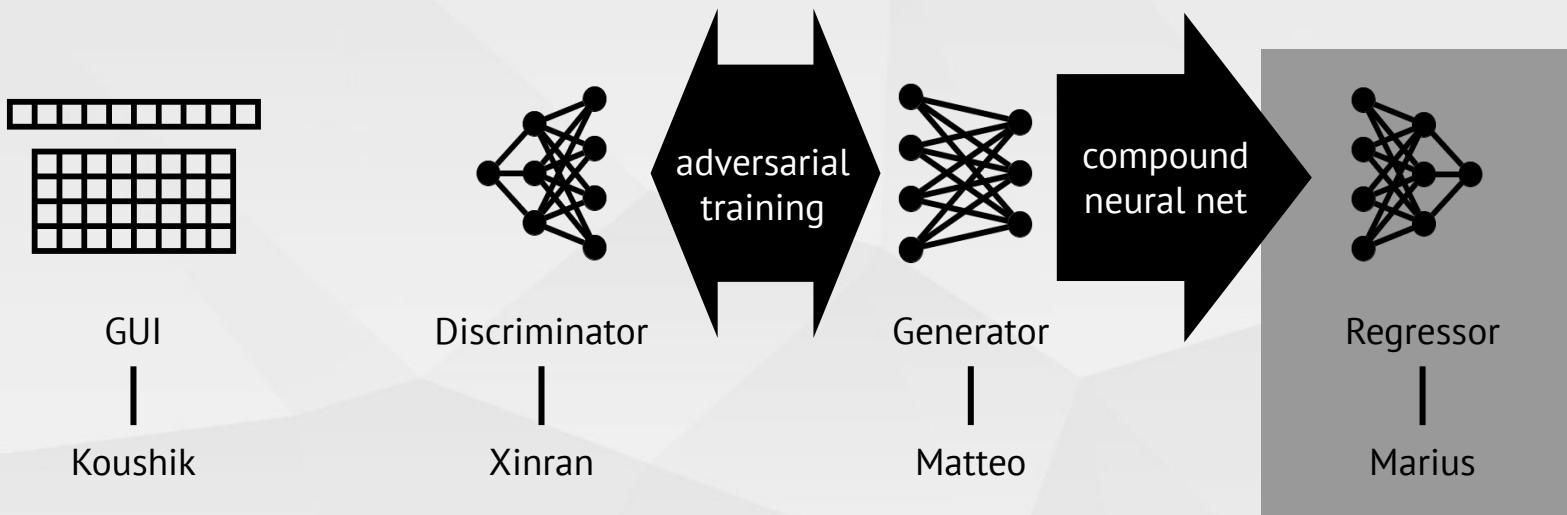


# The Generator

## corpus - methodology - results - conclusion

- To improve
  - Let the network train longer
  - Change the structure of inputs and outputs as described in earliest sessions
  - Run hyperopt for hyper parameter optimization
  - Invent some metric for 'certainty'
- Results
  - No baseline





# **Regressor**

## **Introduction - Methodology - Results - Conclusion**

### **Basic Principle:**

- Generator generates statistics important for final outcome
- Regressor uses these statistics to estimate win chance
- Regressor output not necessarily accurate win chance:
  - Too many combinations to actually calculate chance
  - Important is the order relative to each other
  - Gets used in interface to suggest the best champions

# Regressor

## Introduction - Methodology - Results - Conclusion

### Neural Network:

- All data is one-hot encoded based on champion ID
  - Every combination has one possible encoding
- Feed-forward neural network with one hidden layer
  - 16354 neurons -> 100 neurons -> 1 output neuron
  - Output gets passed through sigmoid operator
    - Naturally leads to output between 0 and 1

# Regressor

## Introduction - Methodology - Results - Conclusion

### Results:

- When win =  $>50\%$  and loss =  $<50\%$ : ~96% accuracy
  - Actual accuracy can't be calculated
- Accuracy should be taken with a grain of salt:
  - Matches often have clear indicators who won:
    - Teams need to destroy 5 at least 5 towers to win
    - Teams with less than 5 towers can't have won

# Regressor

## Introduction - Methodology - Results - Conclusion

### Conclusion:

- While regressor accuracy on test data can be very high, actual generator data may lead to less accurate results
- Generator can't perfectly predict stats -> imperfect results
- This is okay, as champion combinations don't inherently lead to wins or losses
- Goal is achievable: Suggest good champions to select

# Android Interface

**PART A**  
Information

**PART B**  
Design

**PART C**  
Win Chance  
Generation

**PART D**  
Conclusion

# Android Interface

## Information - Design - Win Chance Generation - Conclusion

### What information can the user provide?

- Which side of the map they play on
  - Map is point-symmetric, but player roles aren't
    - Leads to advantages for roles on certain sides
- Which team selects first (mainly used for interface)
- Which champions were selected
- Which summoner spells were selected in user's team
- Indirectly: Time of game creation



Map position:

Top Right

Bottom Left

First to pick:

Enemy

You

Confirm

# Android Interface

## Information - Design - Win Chance Generation - Conclusion

### Mirrors in-game champion selection process:

- Champion selection order same as in-game
  - Alternates between teams, start team is random
- Win rate gets calculated per champion
  - Champions are ordered by win rate, best at top
  - Win rate is indicated by green / red stripe on icon
    - Green =  $>50\%$ , red =  $<50\%$  win rate
- In-game bans are ignored, would be unnecessary
  - User can just ignore banned champions



# Android Interface

## Information - Design - Win Chance Generation - Conclusion

### After champions are selected:

- Summoner spells are displayed for user's team
  - Enemy team's summoner spells aren't visible to user
- Every player selects two summoner spells
  - Flash is selected by default, it's the most popular
- Win rate gets calculated for every spell combination
  - Each spell displays its highest win rate on its icon
- Win rate displayed for each player
  - May advise teammates which spells to select



# Android Interface

## Information - Design - Win Chance Generation - Conclusion

### How does the interface retrieve the win rates?

- Option 1: Run python within app
  - Problem: Dependencies would have to be included
  - Python bindings are for outdated versions
- Option 2: Run neural network using math library
  - Problem: Requires additional code
  - Libraries may not be optimized for mobile devices
- Solution: Set up server, send requests to server
  - Only server needs libraries and dependencies

# Android Interface

## Information - Design - Win Chance Generation - Conclusion

### Conclusion:

- Suggests best champions and summoner spells to select
  - Imperfect accuracies may be disregarded
  - Roughly correct order is most important
- Due to Python binding difficulties, app needs to be online
  - League of Legends requires internet connection anyway
- Mirrors in-game champion selection process:
  - Viable real-time companion software