



Machine Learning Systems Design

-

LolFFate

Milestone 3

Context

- League of Legends (LoL) is one of the most popular video games in the world.
- It is:
 - a 5v5 multiplayer game;
 - which mixes **strategy** and **skills**;
 - and is known for causing lots of **frustration**.
- Players can **ForFeit (FF)** after **15 min** of gameplay
- **Problem:**
 - Players waste time in unwinnable games, with no early-game performance insight. **They wonder if their match is worth continuing, or how to improve it.**



source: <https://wiki.leagueoflegends.com/en-us/>

Idea

- A web dashboard which could predict the probability of winning after 15 minutes.
- Improves decision-making for the players:
 - They can have a **clearer view** by seeing which factors matter most (like gold, or objectives);
 - They can quickly spot their **strengths and weaknesses**.
- Potentially **reduces frustration**.



Our solution: LolFFate

- FFate is a platform allowing LoL players to **predict the probability of winning or losing** their current game.
- The predictions are made thanks to an **ML model** and are served through **Flask** and **Streamlit** applications .
- As of now, our app has **2 versions** :
 - The **online** version on **Google Cloud**;
 - The **local** version for the **player's machine**.

The screenshot shows the LolFFate web application interface. At the top, the 'LOLFFATE' logo is displayed in a stylized gold font. Below the logo, there is a text input field containing 'EUW1_6880919285' and a yellow 'Predict' button. The prediction result is shown as 'Blue team wins with probability of 78.73%'. Below this, the 'Prediction History' section features a yellow 'Update Prediction History' button and a list of recent matches with their respective win probabilities. At the bottom, an 'Advanced: Update History' section is visible, containing two input fields for 'Prediction ID' and 'Prediction value', and two yellow buttons: 'Update Last Entry' and 'Update Specific Entry'.

LOLFFATE

EUW1_6880919285 Predict

Prediction:

Blue team wins with probability of 78.73%

Prediction History:

Update Prediction History

Match EUW1_6879899279: Blue team wins with probability of 86.01%
Match EUW1_6879616200: Blue team wins with probability of 85.01%
Match EUW1_6878702910: Red team wins with probability of 76.19%
Match EUW1_6879416454: Red team wins with probability of 80.13%
Match EUW1_6880919285: Blue team wins with probability of 78.73%

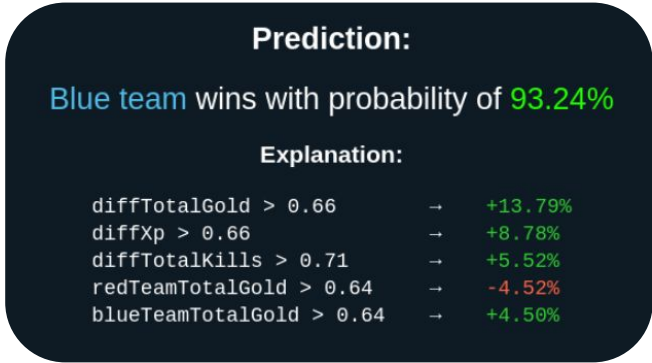
▼ Advanced: Update History

Blue team probability of winning [0, 1]

Prediction value Update Last Entry

Prediction ID Prediction value Update Specific Entry

- Regarding explanation of the results:

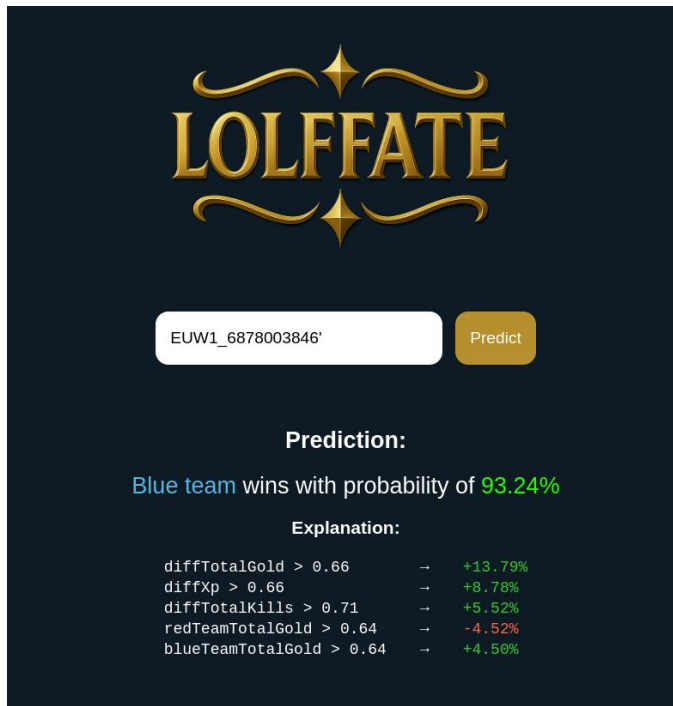


LIME explainer

Architectures for Model Serving

- A Flask application serves our model.
- The Flask web platform provides different functionalities to the users:
 - Get the **prediction probability** of a game by inputting a **game ID** ;
 - See the **explanations** on the **features relevance** in the prediction for a better understanding of them.

Scan to see it!



Architectures for Model Serving

Scan to see it!



Select visualization type

Predictions

Predictions

Dataset

- A Streamlit app also serves our model !
- Same functionality as the flask app for the prediction part.
- ... but it also has data visualization for the dataset.

Predictions

Enter Game ID ↗

Game ID

EUW1_6882489515'

Predict

Prediction:

Probability of blue team winning: 0.19

Prediction explanation:

The following features contributed to the prediction:

diffTotalGold <= -2821.00	→	-13.53%
diffXp <= -2092.00	→	-8.77%
diffTotalKills <= -5.00	→	-6.73%
blueTeamTotalGold <= 25911.25	→	-3.81%
diffTotalDamageToChamps <= -5053.75	→	-3.56%

Processed Data Statistics

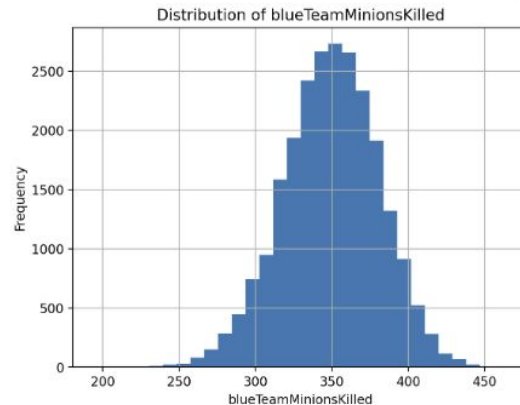
	blueTeamControlWardsPlaced	blueTeamWardsPlaced	blueTeamTotalKills	blueTeamDragonKills	blueT
count	24218	24218	24218	24218	
mean	3.6104	41.3643	12.7909	0.7379	
std	2.0183	43.4773	4.9092	0.7221	
min	0	9	0	0	
25%	2	25	9	0	
50%	3	29	12	1	
75%	5	35	16	1	
max	37	603	38	2	

Feature Distribution

Select feature to visualize

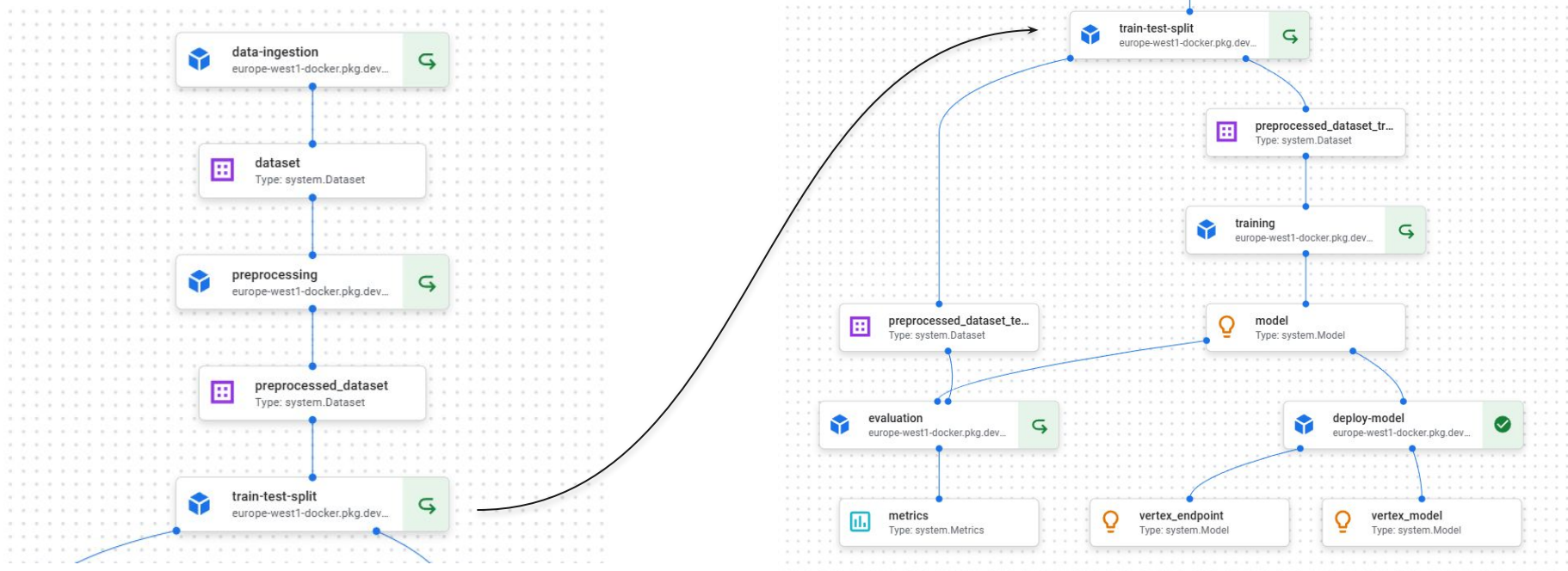
blueTeamMinionsKilled

Distribution of blueTeamMinionsKilled:



Model Pipeline

- Vertex AI pipeline to build our model :



Model Deployment



- Both the **Flask** and **Streamlit** applications are available on **Google Cloud Run**
 - They can therefore be **accessed remotely**
Flask App → <https://flask-app-30182159501.europe-west1.run.app/>
Streamlit App → <https://streamlit-app-30182159501.europe-west1.run.app/>
- The **online** version of our app makes predictions on **existing matches** from the training dataset.
 - For example, try entering the match ID `EUW1_6880890229`
- The **latest Flask application** is **automatically re-deployed on Google Cloud** when a pull request on the main branch is merged thanks to the **GitHub CICD**.

Riot Web API

- Riot Games provides **two APIs** for LoL:
 - Web API;
 - Client API (local).
- The **Web API** features :
 - Player info;
 - Match history;
 - Game statistics;
 - Timeline...
- The **API key**:
 - Only lasts 24h;
 - is Rate-limited.
- **Permanent access** to the API requires
 - Production key;
 - Manual validation from Riot Games through a formal application process.



Client API & Local Version

- The **Client API** features:
 - Information about local live game.
- We built a **local version** of our application which:
 - Uses Client API;
 - Makes outcome predictions on the ongoing match launched on the player's machine.
- Demo





Machine Learning Systems Design

-

LolFFate

Thank you for listening!