# Machine Learning Systems Design

-

# LolFFate

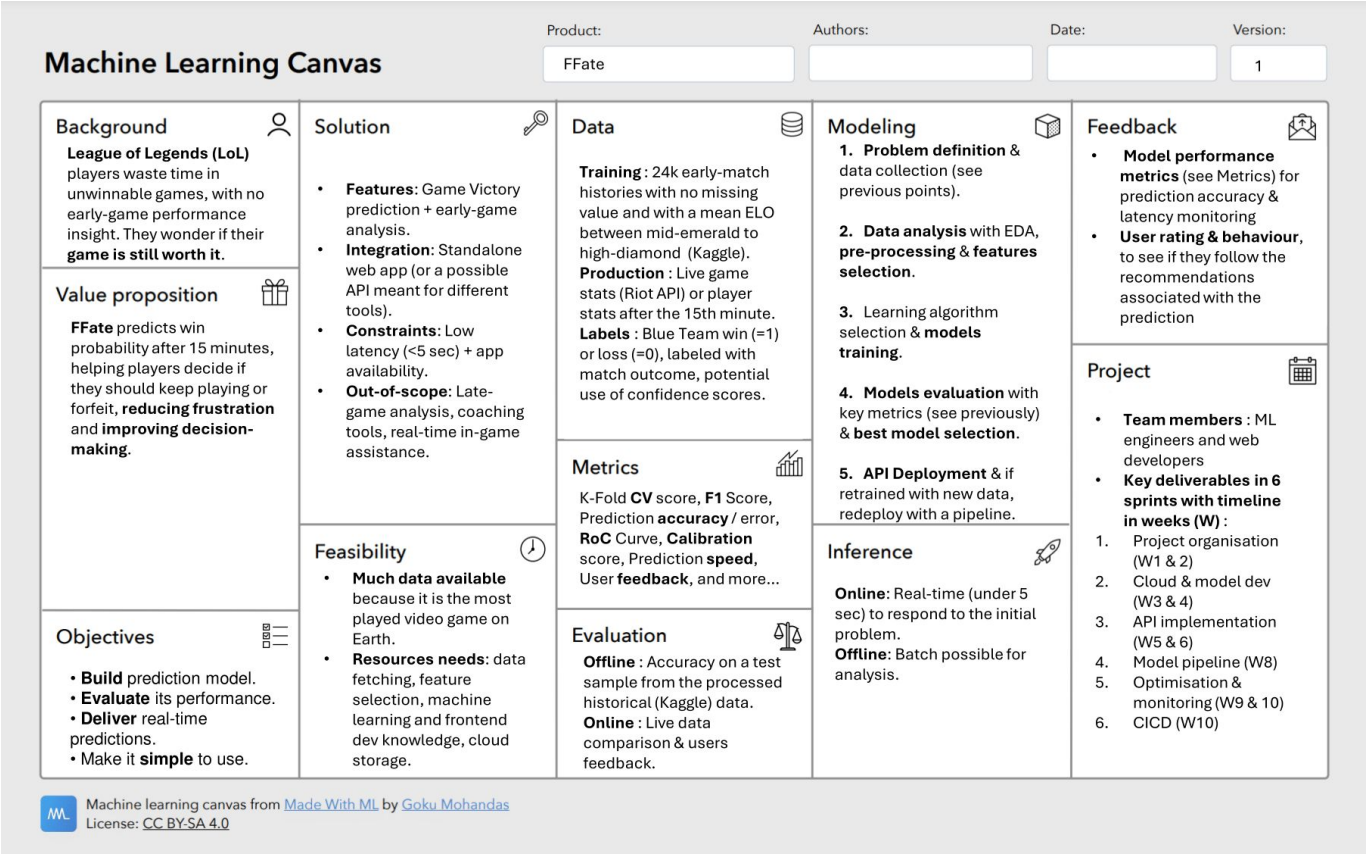
## Milestone 1

# Context

- 5v5 multiplayer game
- Mix of strategy and skills
- Known for causing lots of frustration
- Players can Forfeit after 15 min of gameplay
- But should they… ?


# Use case

Our solution : **FFate**



## Machine Learning Canvas

Product: FFate | Authors: | Date: | Version: 1

### Background
**League of Legends (LoL)** players waste time in unwinnable games, with no early-game performance insight. They wonder if their **game is still worth it**.

### Value proposition
**FFate** predicts win probability after 15 minutes, helping players decide if they should keep playing or forfeit, **reducing frustration** and **improving decision-making**.

### Objectives
- **Build** prediction model.
- **Evaluate** its performance.
- **Deliver** real-time predictions.
- Make it **simple** to use.

### Solution
- **Features**: Game Victory prediction + early-game analysis.
- **Integration**: Standalone web app (or a possible API meant for different tools).
- **Constraints**: Low latency (<5 sec) + app availability.
- **Out-of-scope**: Late-game analysis, coaching tools, real-time in-game assistance.

### Feasibility
- **Much data available** because it is the most played video game on Earth.
- **Resources needs**: data fetching, feature selection, machine learning and frontend dev knowledge, cloud storage.

### Data
**Training** : 24k early-match histories with no missing value and with a mean ELO between mid-emerald to high-diamond (Kaggle).
**Production** : Live game stats (Riot API) or player stats after the 15th minute.
**Labels** : Blue Team win (=1) or loss (=0), labeled with match outcome, potential use of confidence scores.

### Metrics
K-Fold **CV** score, **F1** Score, Prediction **accuracy** / error, **RoC** Curve, **Calibration** score, Prediction **speed**, User **feedback**, and more...

### Evaluation
**Offline** : Accuracy on a test sample from the processed historical (Kaggle) data.
**Online** : Live data comparison & users feedback.

### Modeling
1. **Problem definition** & data collection (see previous points).
2. **Data analysis** with EDA, **pre-processing** & **features selection**.
3. Learning algorithm selection & **models training**.
4. **Models evaluation** with key metrics (see previously) & **best model selection**.
5. **API Deployment** & if retrained with new data, redeploy with a pipeline.

### Inference
**Online**: Real-time (under 5 sec) to respond to the initial problem.
**Offline**: Batch possible for analysis.

### Feedback
- **Model performance metrics** (see Metrics) for prediction accuracy & latency monitoring
- **User rating & behaviour**, to see if they follow the recommendations associated with the prediction

### Project
- **Team members** : ML engineers and web developers
- **Key deliverables in 6 sprints with timeline in weeks (W)** :

1. Project organisation (W1 & 2)
2. Cloud & model dev (W3 & 4)
3. API implementation (W5 & 6)
4. Model pipeline (W8)
5. Optimisation & monitoring (W9 & 10)
6. CICD (W10)

# Data


CSV format

24 225 observations

Features :

- match ID
- + 14 features for the red team
- + 14 features for the blue team
- + target feature about the win

# EDA Report

- 1 additional column only made of 0 -> removed
- 7 duplicates observations -> removed
- no missing values

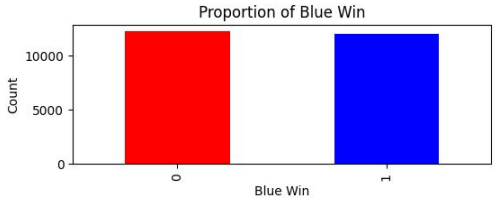
Other observations :

- 20 numerical & 8 categorical features
- Many highly correlated features because of the game mechanics.
  - e.g. `blueTeamMinionsKilled` and `blueTeamXp`
  - e.g. `blueTeamDragonKills` and `redTeamDragonKills`

New features : difference between the teams mirror numerical values

Dataset statistics

| | |
|---|---|
| **Number of variables** | 28 |
| **Number of observations** | 24218 |
| **Missing cells** | 0 |
| **Missing cells (%)** | 0.0% |
| **Duplicate rows** | 0 |
| **Duplicate rows (%)** | 0.0% |
| **Total size in memory** | 6.7 MiB |
| **Average record size in memory** | 289.0 B |



# Model experimentation

Model :  Random Forest

80% / 20%  for training and test set

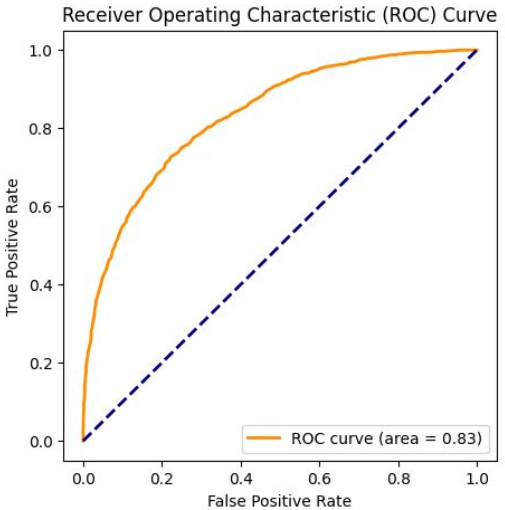
Standard Scaler

Feature selection

Hyperparameter -> 5-split cross validation

Evaluation -> Accuracy, F1 score, ROC curve

```
Model Evaluation:
Accuracy:  0.7487613542526838
F1 Score:  0.7487481257038276
AUC(ROC):  0.8332002422070925
```




Best K number: 15

Selected features:  ['blueTeamTotalKills', 'blueTeamTotalGold', 'blueTeamXp', 'redTeamTotalKills', 'redTeamTurretPlatesDestroyed', 'redTeamTotalGold', 'redTeamXp', 'diffMinionsKilled', 'diffTotalGold', 'diffTotalKills', 'diffXp', 'diffTotalDamageToChamps', 'diffDragonKills', 'diffTowersDestroyed', 'diffTurretPlatesDestroyed']

Best Parameters:  {'n_estimators': 200, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': None, 'criterion': 'gini', 'bootstrap': True}

# Machine Learning Systems Design

-

# LolFFate

Thanks for listening !