



# Insight System

Crispino, Zyrus  
CPE31

A Machine Learning System for Predicting Team Win Rates, Character Builds, and Individual Performance in Eternal Return

# Introduction



• • •

- Eternal Return is a MOBA–battle royale hybrid game with 80 characters, each with unique builds.
- Players must optimize team composition, character performance, and item builds to improve win rates.

# Introduction

- Decision-making in the game is complex due to the dynamic meta and diverse character interactions.
- This project aims to assist players by applying machine learning to analyze past matches and generate strategic recommendations.
- The system consists of three core models: Team Win Prediction, Character Performance Evaluation, and Character Build Recommendation.



# Objectives

- Predict Team Performance
- Predict Character Suggested Builds
- Predict Individual Character Performance



# Methodology

**Eternal Return Developer Portal**

Get Started      Our APIs

**EXPLORE AND BUILD**

"Get Started"에서 눌러 이터널 리턴 API를 시작하는 법을 알아보세요.

"Our APIs"에서 이터널 리턴 API가 제공하는 내용을 확인할 수 있습니다.

여기에서 이터널 리턴 API의 상세 정보를 확인할 수 있습니다.

문의 사항이 있다면 [이터널 리턴 개발자 디스코드](#)에 방문해주세요.

**bser-open-api-handler-release**

Endpoint: <https://open-api.bser.io>  
Description: Eternal Return Open API  
Unsubscribe

Schemes: **HTTPS**

**Eternal Return**

**Open API** Open API for Eternal Return

**data**

Method	Path	Description
GET	/v1/data/{metaType}	fetch metaType
GET	/v1/freeCharacters/{matchingMode}	fetch freeCharacters by matchingMode
GET	/v1/l10n/{language}	fetch l10n
GET	/v2/data/{metaType}	fetch metaType

Authorize

# Data Gathering

```
game_id = 46678263 ...

... Player: 尋風追憶 (UserNum: 5359969)
  Game ID: 46678263 | Season ID: 31 | Rank: 7
  Character Num: 29
  Team Kill: 12 | Kills: 3 | Deaths: 2 | Assists: 7
  Damage to Players: 10909 | Damage to Monsters: 31758
  Equipment: Weapon: 111501, Chest: 202404, Head: 201507, Arm: 705608, Leg: 204419
  Traits: Main1: 7010701, Main2: 7310301, Sub1: 7010501, Sub2: 7010601
  -----
  Player: 텁미니스트 (UserNum: 5294646)
    Game ID: 46678263 | Season ID: 31 | Rank: 5
    Character Num: 65
    Team Kill: 15 | Kills: 2 | Deaths: 1 | Assists: 11
    Damage to Players: 13044 | Damage to Monsters: 49417
    Equipment: Weapon: 102412, Chest: 202506, Head: 201522, Arm: 205401, Leg:
    Traits: Main1: 7110801, Main2: 7210801, Sub1: 7110101, Sub2: 7110601
  -----
  Player: 범지 (UserNum: 4911433)
    Game ID: 46678263 | Season ID: 31 | Rank: 4
    Character Num: 47
    Team Kill: 15 | Kills: 5 | Deaths: 4 | Assists: 6
    Damage to Players: 16738 | Damage to Monsters: 89521
    Equipment: Weapon: 109404, Chest: 202503, Head: 201409, Arm: 203413, Leg: 204412
    Traits: Main1: 7110701, Main2: 7110201, Sub1: 7211101, Sub2: 7210801
  -----
  Player: ふでペンボルペン (UserNum: 4888343)
  ...
    Damage to Players: 19210 | Damage to Monsters: 119376
    Equipment: Weapon: 121701, Chest: 202510, Head: 201521, Arm: 205501, Leg: 204509

  game_id = 46678263 ...

  ... Player: 尋風追憶 (UserNum: 5359969)
    Game ID: 46678263 | Season ID: 31 | Rank: 7
    Character: Leon | Weapon: Glove
    Equipment: Weapon: Obsidian_Jitte, Chest: Covert_Agent_Uniform, Head: Astronaut's_Helmet, Arm: Pharaoh's_Artifact, Leg: Delta_I
    Traits: Main1: Open_Wounds, Main2: Overwatch, Sub1: Dismantle_Goliath, Sub2: Anima_Reaper
    Team Kill: 12 | Kills: 3 | Deaths: 2 | Assists: 7
    Damage to Players: 10909 | Damage to Monsters: 31758
  -----
  텁미니스트 (UserNum: 5294646)
    Game ID: 46678263 | Season ID: 31 | Rank: 5
    Character: Debi_&_Marlene | Weapon: Two-handed_Sword
    Equipment: Weapon: Aurora_Longsword, Chest: Phantom_Jacket, Head: Commander_Headset, Arm: Moonlight_Pendant, Leg: Cowboy_Boots
    Traits: Main1: Camping_Guide, Main2: Penny_Pincher, Sub1: Embolden, Sub2: Dine_n'_Dash
    Team Kill: 12 | Kills: 2 | Deaths: 1 | Assists: 11
    Damage to Players: 13044 | Damage to Monsters: 49417
  -----
  범지 (UserNum: 4911433)
    Game ID: 46678263 | Season ID: 31 | Rank: 4
    Character: Laura | Weapon: Whip
    Equipment: Weapon: Plasma_Whip, Chest: Holy_Orders, Head: Imperial_Crown, Arm: Tellurian_Timepiece, Leg: Tachyon_Brace
    Traits: Main1: Unwavering_Mentality, Main2: Cavalcade, Sub1: Coin_Toss, Sub2: Penny_Pincher
    Team Kill: 15 | Kills: 5 | Deaths: 4 | Assists: 6
    Damage to Players: 16738 | Damage to Monsters: 89521
  -----
  Player: ふでペンボルペン (UserNum: 4888343)
  ...
    Traits: Main1: Dismantle_Goliath, Main2: Carnivore, Sub1: Stopping_Power, Sub2: Power_Crescendo
```

# Data Gathering

AutoSave Off players\_data.csv • Saved to this PC

File Home Insert Page Layout Formulas Data Review View Automate Help

Cut Copy Format Painter

Aptos Narrow 11 A A Wrap Text General Conditional Formatting

Font Alignment Number Styles

Clipboard Font Alignment Number Styles

Normal Bad Good Neutral Calculation Check Cell

Insert Delete Format AutoSum Fill Sort & Find & Select Add-ins Analyze Data

Comments Share

A1 userNum nickname gameId seasonId gameRank character weapon itemWeap itemChest itemHead itemArm itemLeg Trait\_Mair Trait\_Mair Trait\_Sub1 Trait\_Sub2 TeamKill Kill Death Assist Dmg\_Play Dmg\_Monster

1 5426009 7 46678118 31 2 Cathy Dagger Asura The\_Rever Imperial\_(Tindalos Straitjack Unwaverir Cavalcade Coin\_Toss Penny\_Pir 5 0 3 5 12065 53392

2 5296453 7 46678118 31 1 Hisui Two-hand Aurora\_Lo\_ř\_o\_D Blaster\_H Mythril\_Q Blade\_Boc Contempt Anima\_Re Unwaverir Cavalcade 17 5 4 7 15397 79563

3 5162955 7 46678118 31 5 Adina Arcana The\_Moor Blood\_Clo Persona Nightmare Rose\_Step Open\_Woi Overwatch Dismantle Anima\_Re 19 12 2 6 36449 67664

4 5140049 7 46678118 31 7 Ly Anh Dagger Mahara Dragon\_D Cowboy\_H Helix Steel\_Kne Frenzy Carnivore Painkiller Dine\_n\_D 10 0 3 6 6315 23338

5 5051669 7 46678118 31 7 Tsubame Shuriken Petal\_Torr Changpac Crystal\_Ti Dice\_of\_D Gladiator Dismantle Anima\_Re Unwaverir Cavalcade 10 9 2 0 17556 50226

6 4507510 Lisha 46678118 31 5 Abigail Axe Scarlet\_Sc The\_Rever Astronaut Emerald\_1 Rose\_Step Stopping\_ Overwatch Embolden Cavalcade 19 5 3 9 26103 31188

7 4388419 7 46678118 31 2 Debi\_&\_M Two-hand Monohost Phantom\_Command Mythril\_Q Stellar\_S Frenzy Anima\_Re Painkiller Cavalcade 5 4 3 1 11467 40443

8 4208243 Pyosik 46678118 31 6 Markus Axe Santa\_Mu Guardian Tactical\_V Music\_Bo Hiking Bo Painkiller Dine\_n\_D Camping Penny\_Pir 12 1 2 9 14395 32544

9 4201423 7 46678118 31 4 Magnus Hammer Monkey\_K Burgundy Crusader\_Burning\_H Blade\_Boc Contempt Carnivore Painkiller Dine\_n\_D 5 0 2 3 5216 47480

10 4098060 milo99 46678118 31 6 Aya Pistol AK-12 Phantom\_Legatus Radar Blade\_Boc Dismantle Carnivore Unwaverir Dine\_n\_D 12 6 3 2 22850 54776

11 3604479 penguinx 46678118 31 8 Garnet Bat Spy\_Umbr Ghost\_Bri Helm\_of\_E Burnished Steel\_Kne Thorn\_Sh Penny\_Pir Painkiller Dine\_n\_D 4 0 1 2 7601 47602

12 2891079 Undying 46678118 31 4 Katja Sniper\_Rif Blackfire\_Tactical\_A Chinese\_C Chain\_of\_M Mirage\_La Stopping\_ Overwatch Dismantle Anima\_Re 5 3 1 1 15616 65923

13 2872770 7 46678118 31 4 Shoichi Dagger Eclipse Couturier Chinese\_C Pharaoh's Legs\_of\_St Frenzy Anima\_Re Unwaverir Cavalcade 5 2 3 1 6570 114067

14 2862218 7 46678118 31 7 Hisui Two-hand Monohos\_ř\_o\_D Welding\_Plasma\_A White\_Rhi Contempt Anima\_Re Unwaverir Cavalcade 10 1 4 3 16327 32496

15 2674373 7 46678118 31 1 Adriana Throw Astrap\_ř\_o Blood\_Clo Persona Nightmare Delta\_Red Open\_Woi Recharge Dismantle Anima\_Re 17 3 3 10 29443 62708

16 2051088 7 46678118 31 8 Justyna Crossbow Phantom\_Blood\_Clo Persona Corrupting Delta\_Red Open\_Woi Power\_Cre Coin\_Toss Penny\_Pir 4 2 2 1 14896 49052

17 1776315 abconl 46678118 31 3 Haze Assault\_R Valkyrie Inquisitor Astronaut Eye\_of\_Ho Tachyon\_I Stopping\_ Power\_Cre Contempt Anima\_Re 3 0 5 3 11017 43818

18 1729906 7 46678118 31 6 Vanya Arcana The\_Emp Holy\_Orde Astronaut Tindalos\_Legs\_of\_St Frenzy Anima\_Re Embolden Dine\_n\_D 12 5 3 7 16275 42804

19 1376944 Casval 46678118 31 3 Yuki Two-hand Aurora\_Lo Spectral\_J Helm\_of\_E Laced\_Qu Wild\_Wall Painkiller Dine\_n\_D Circular\_S Overwatch 3 2 2 0 8854 42861

20 1169313 7 46678118 31 5 Leni Pistol Elegance Elf\_Dress Command Nightinga Tachyon\_I Coin\_Toss Penny\_Pir Circular\_SR\_echarge 19 2 1 16 13226 29907

21 906654 7 46678118 31 1 Luke Bat Monkey\_K Omert\_ř Tactical\_V Auto-arm Red\_Shoe Embolden Cavalcade Frenzy Anima\_Re 17 9 3 6 23367 59508

22 750800 7 46678118 31 2 Karla Crossbow Phantom\_Holy\_Orde Racing\_H Tellurian\_Gladiator Contempt Anima\_Re Unwaverir Cavalcade 5 1 4 3 12620 53742

23 265744 7 46678118 31 3 Barbara Pistol Altair Kabana Persona Tindalos\_Killer\_Hee Coin\_Toss Penny\_Pir Open\_Woi Power\_Cre 3 1 5 2 28246 117424

24 2104 Cardboard 46678118 31 8 Kenneth Axe Axe\_of\_Pa Racing\_Su Tactical\_V Plasma\_A Mythril\_B Embolden Dine\_n\_D Circular\_S Power\_Cre 4 2 2 1 6287 34158

25 5483809 Pieck 46678050 0 8 Emma Shuriken Fuhma\_Si Command Persona Corrupting SCV Circular\_SR\_echarge Unwaverir Cavalcade 9 2 4 3 9795 17128

26 5462932 Rokke 46678050 0 1 Vanya Arcana The\_Death\_Holy\_Orde Fencing\_N Tindalos\_Legs\_of\_St Frenzy Anima\_Re Embolden Dine\_n\_D 13 2 1 8 7690 36144

27 5383963 7 46678050 0 5 Celine Throw Ruthenium Blood\_Clo Persona Bracelet\_c Legs\_of\_St Stopping\_ Power\_Cre Contempt Anima\_Re 14 4 4 6 10171 44153

28 5257451 7 X 46678050 0 1 Camilo Dual\_Swo Phantom\_Phantom\_Legatus Radar Racing\_Bc Frenzy Anima\_Re Painkiller Cavalcade 13 9 1 3 22704 150481

29 5216788 nonenon 46678050 0 3 Rio Bow Twinbow Specter Legatus Radar Stellar\_Ste Dismantle Anima\_Re Stopping\_ Power\_Cre 5 1 3 4 6646 79906

30 5005855 7 46678050 0 2 Debi\_&\_M Two-hand Aurora\_Lo Phantom\_Command Moonlight Wild\_Wall Camping Penny\_Pir Embolden Dine\_n\_D 8 2 2 5 10830 107257

31 4930709 7 46678050 0 7 Haze Assault\_R Gold\_Rusl Blood\_Clo Imperial\_I Nightmare Delta\_Red Open\_Woi Power\_Cre Dismantle Anima\_Re 9 3 4 3 7317 39623

32 4733798 yumemin 46678050 0 4 Yuki Two-hand Aurora\_Lo Spectral\_J Helm\_of\_E Auto-arm Wild\_Wall Painkiller Dine\_n\_D Circular\_S Overwatch 5 2 2 3 6893 48839

33 4501945 moyu233 46678050 0 6 Irem Throw Sticky\_Bor Holy\_Orde Imperial\_(Eye\_of\_Ho Delta\_Red Stopping\_ Overwatch Contempt Endorphin 7 3 2 1 7526 23473

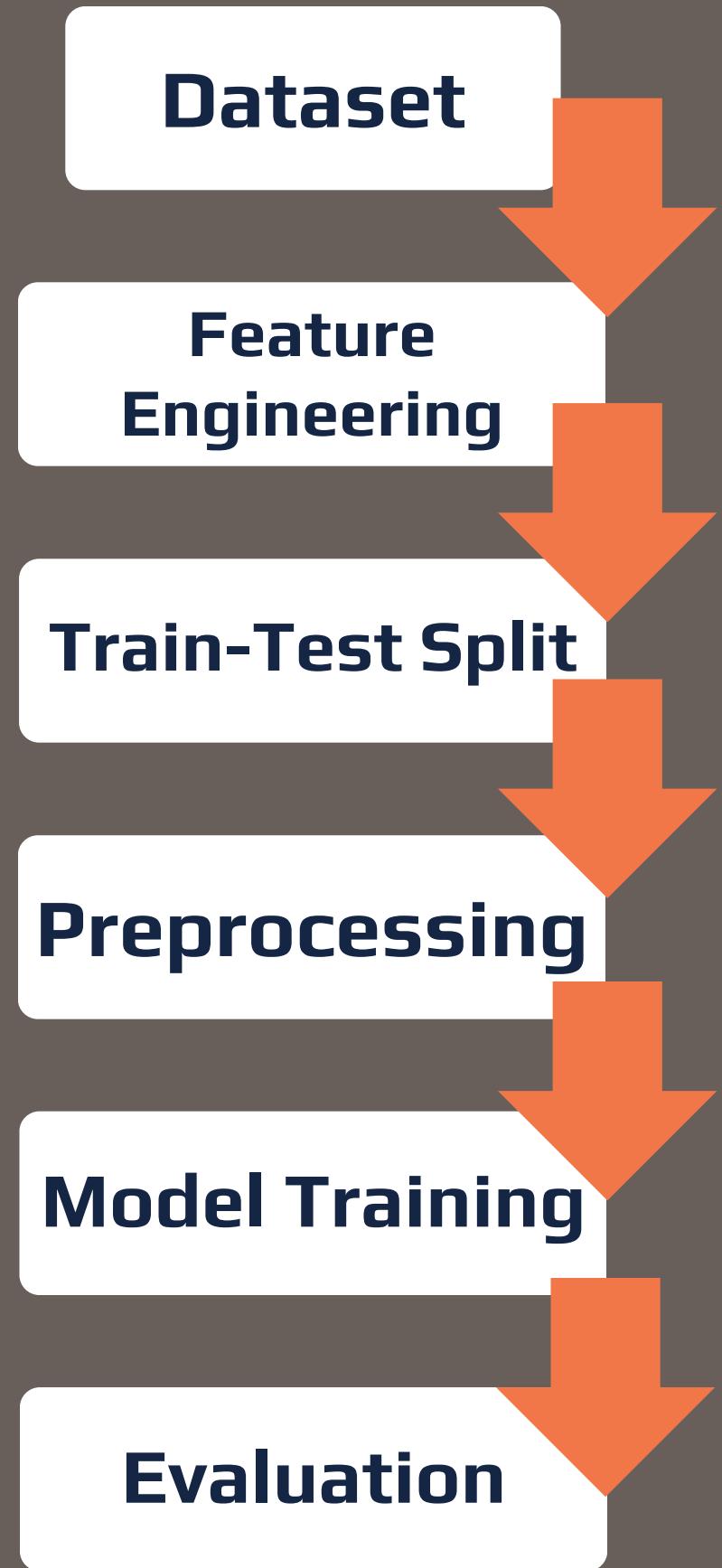
34 4414670 nkkmdmr 46678050 0 2 Bianca Arcana The\_Moor Elegant\_G Persona Nightmare Legs\_of\_St Thrill\_of\_t Penny\_Pir Open\_Woi Power\_Cre 8 3 4 2 9037 31915

35 4256256 7 46678050 0 4 Karla Crossbow Blood\_Hal\_Holy\_Orde Racing\_H Tellurian\_Racing\_Bc Contempt Anima\_Re Unwaverir Cavalcade 5 3 1 0 4295 124239

36 4189667 7 46678050 0 3 Elena Rapier Sword\_of\_Bazinga\_D Command Burnished Wild\_Wall Painkiller Cavalcade Coin\_Toss Penny\_Pir 5 1 3 3 5816 41170

# Data Gathering

# Model Overview



# Models

## Classification

- RandomForestClassifier (from sklearn.ensemble)
- GradientBoostingClassifier (from sklearn.ensemble)
- AdaBoostClassifier (from sklearn.ensemble)
- ExtraTreesClassifier (from sklearn.ensemble)
- LogisticRegression (from sklearn.linear\_model)
- DecisionTreeClassifier (from sklearn.tree)
- SVC (from sklearn.svm)
- KNeighborsClassifier (from sklearn.neighbors)
- XGBClassifier (from xgboost)
- **[USED]** LGBMClassifier (from lightgbm)

## Regression

- LinearRegression (from sklearn.linear\_model)
- Ridge (from sklearn.linear\_model)
- Lasso (from sklearn.linear\_model)
- ElasticNet (from sklearn.linear\_model)
- DecisionTreeRegressor (from sklearn.tree)
- **[USED]** RandomForestRegressor (from sklearn.ensemble)
- GradientBoostingRegressor (from sklearn.ensemble)
- **[USED]** HistGradientBoostingRegressor (from sklearn.ensemble)
- AdaBoostRegressor (from sklearn.ensemble)
- ExtraTreesRegressor (from sklearn.ensemble)
- XGBRegressor (from xgboost)
- LGBMRegressor (from lightgbm)

Disclaimer: I do not claim full knowledge of the models listed. For the most part, I simply used and evaluated their results.

## 1. Load dataset

	sum nickname	gameId	seasonId	gameRank	character	weapon	\
0	426809	마사	46678118	31	2	Cathy	Dagger
1	5296453	쿠로미여동생	46678118	31	1	Hisui	Two-handed_Sword
2	5162955	우주생명마음	46678118	31	5	Adina	Arcana
3	5140849	只玩过卡拉彼丘	46678118	31	7	Ly Anh	Dagger
4	5051669	초고교금마다기	46678118	31	7	Tsubame	Shuriken
	itemWeapon	itemChest	itemHead	...	Trait_Main1	\	
0	Asura	The_Revenant	Imperial_Crown	...	Unwavering_Mentality		
1	Aurora_Longsword	Ao_Dai	Blaster_Helmet	...	Contempt_for_the_Weak		
2	The_Moon	Blood_Cloak	Persona	...	Open_Wounds		
3	Maharaja	Dragon_Dobok	Cowboy_Hat	...	Frenzy		
4	Petal_Torrent	Changpao	Crystal_Tiara	...	Dismantle_Goliath		
	Trait_Main2	Trait_Sub1	Trait_Sub2	TeamKill	Kill	Death	\
0	Cavalcade	Coin_Toss	Penny_Pincher	5	0	3	
1	Anima_Reaper	Unwavering_Mentality	Cavalcade	17	5	4	
2	Overwatch	Dismantle_Goliath	Anima_Reaper	19	12	2	
3	Carnivore	Painkiller	Dine_n'_Dash	18	8	3	
4	Anima_Reaper	Unwavering_Mentality	Cavalcade	18	9	2	
	Assist	Dmg_Player	Dmg_Monster				
0	5	12865	53392				
1	7	15397	79563				
2	6	36449	67664				
3	6	6315	23338				
4	8	17556	58226				
	[ 5 rows x 22 columns ]						

One row = One player

## 2. Clean data and format dataframe

Team stats with separated characters:

	Team_ID	gameId	Kill	Assist	Death	Dmg_Player	Dmg_Monster	\
0	46678029_1_0	46678029	13	21	11	62112	208177	
1	46678029_2_0	46678029	7	10	12	38223	153835	
2	46678029_3_0	46678029	11	13	10	46516	173631	
3	46678029_4_0	46678029	9	15	5	38821	181832	
4	46678029_5_0	46678029	8	12	10	57173	90100	

teamPlacement char1 char2 char3

0	1	Adina	Lenore	Li Dailin
1	2	Alonso	Adela	Yumin
2	3	Katja	Yuki	Cathy
3	4	Celine	Alonso	Hart
4	5	Emma	Estelle	Katja

One row = One team.

TeamWin

# TeamWin

## 3. Setup pre-processing Pipeline

```
# Preprocessing pipelines
numeric_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline([
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
    ('onehot', OneHotEncoder(handle_unknown='ignore', sparse_output=False))
])

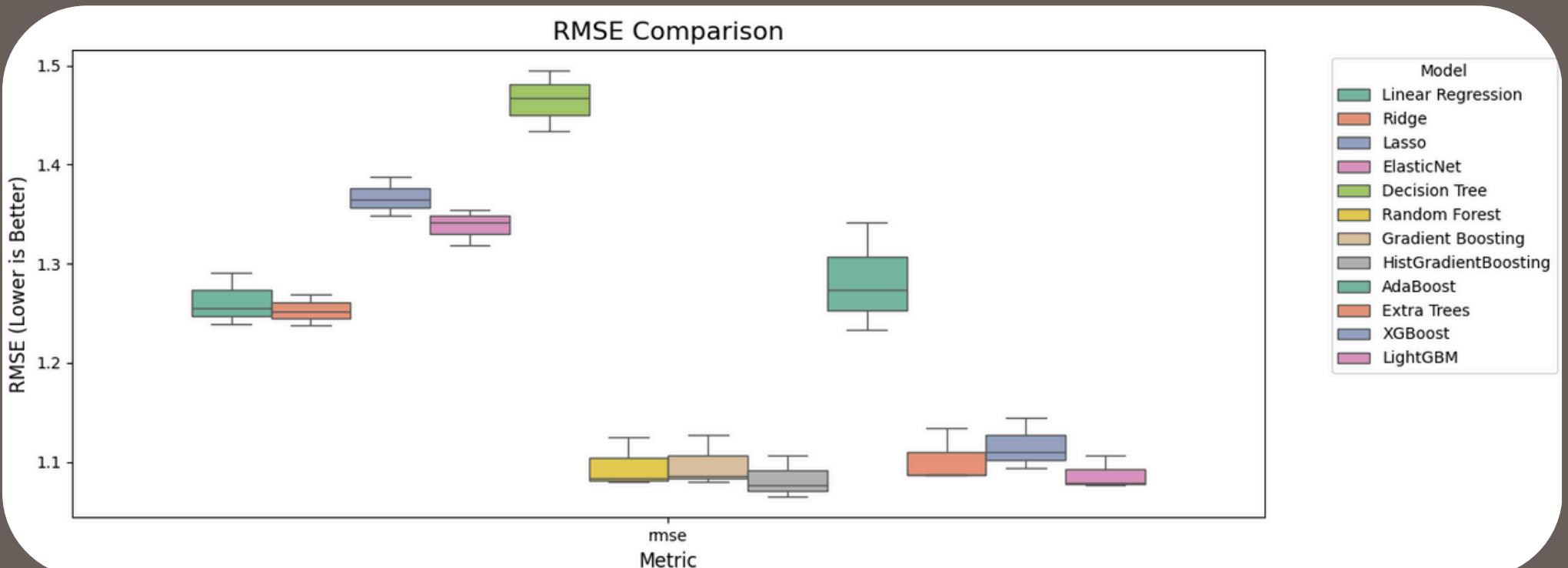
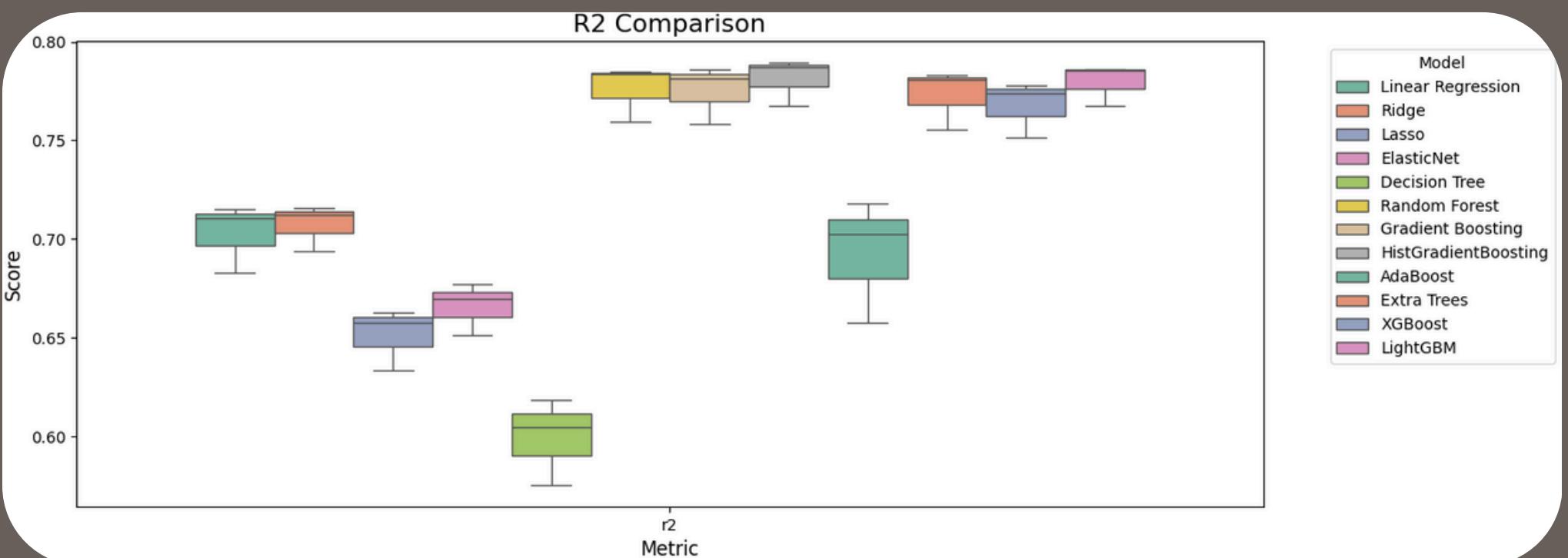
preprocessor = ColumnTransformer([
    ('num', numeric_transformer, Numeric),
    ('cat', categorical_transformer, Categorical)
])
0.0s
```

```
# Insert model here
TeamWin_Pipeline = Pipeline([
    ('preprocessor', preprocessor),
    ('regressor', RandomForestRegressor(n_jobs=-1, random_state=42))
])
0.0s
```

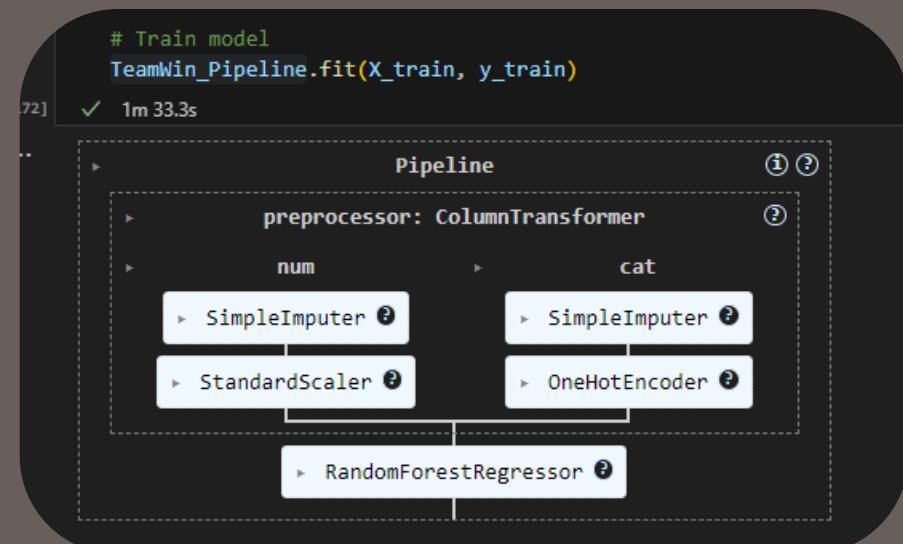
## 4. Train Test Split

```
# Train/test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
✓ 0.1s
```

## 5. Evaluate Models with sample



## 6. fit pipeline with chosen model using full data



## 7. Evaluate

```
# Evaluation
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r2 = r2_score(y_test, y_pred)
print(f"RMSE: {rmse:.2f}")
print(f"R^2 Score: {r2:.2f}")

[175]    ✓  0.0s
```

... RMSE: 1.12  
R^2 Score: 0.75

## 8. Test

```
#test:
TestCode_Predict('Li Dailin', 'Adina', 'Lenore')

✓ 0.7s
```

Exact Team Match Found for Li Dailin, Adina, Lenore  
Games Played Together : 82  
1st Place Finishes : 1  
2nd-3rd Place Finishes : 17  
Total Top 3 Finishes : 18  
Win Rate (Top 3) : 21.95%  
Average Placement : 4.24  
  
Combined Stats for the Team:  
Total Kills : 9.17  
Total Assists : 14.90  
Total Deaths : 8.17  
Total Dmg (Player) : 45214.74  
Total Dmg (Monster): 142537.22  
Predicted Placement (From Model): 3.91

# TeamWin

# CharWin

1. Load dataset

2. Clean Dataset

3. Establish Features  
and Target

```
# Features and target
features = [
    'character', 'weapon', 'itemWeapon', 'itemChest', 'itemHead',
    'itemArm', 'itemLeg', 'Trait_Main1', 'Trait_Main2', 'Trait_Sub1', 'Trait_Sub2',
    'TeamKill', 'Kill', 'Death', 'Assist', 'Dmg_Player', 'Dmg_Monster'
]
target = 'gameRank'

x = df[features]
y = df[target]

0.0s
```

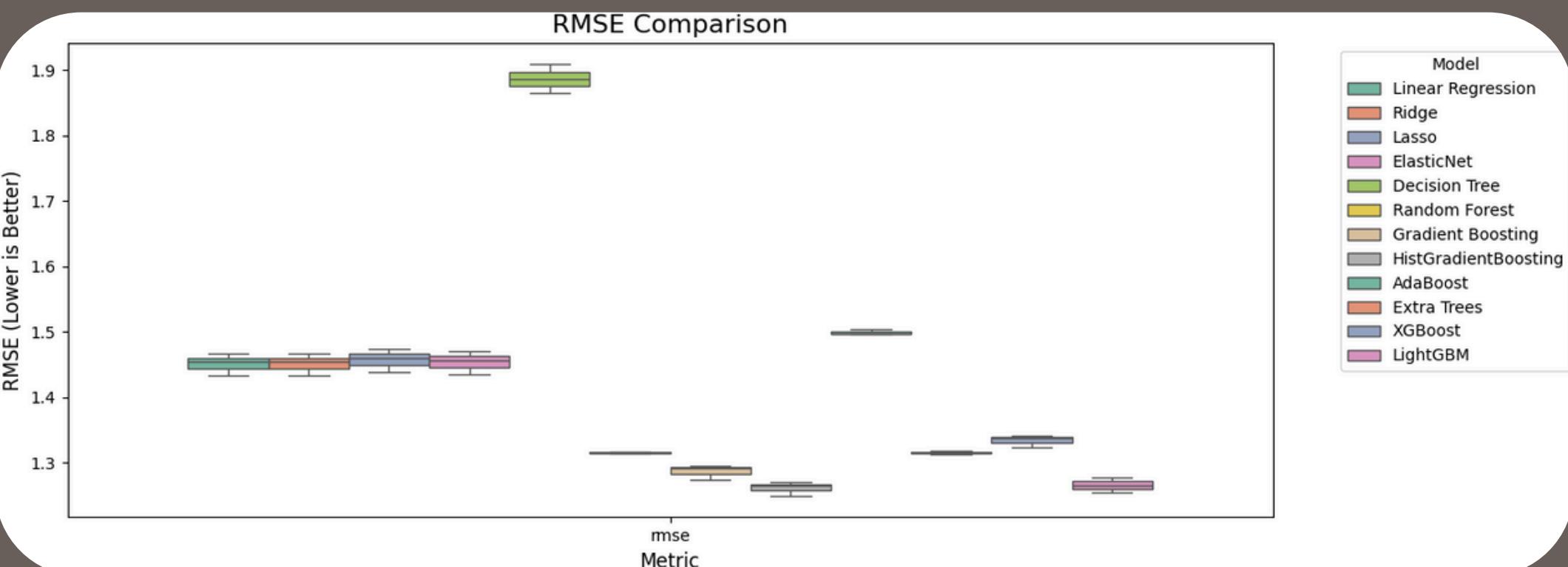
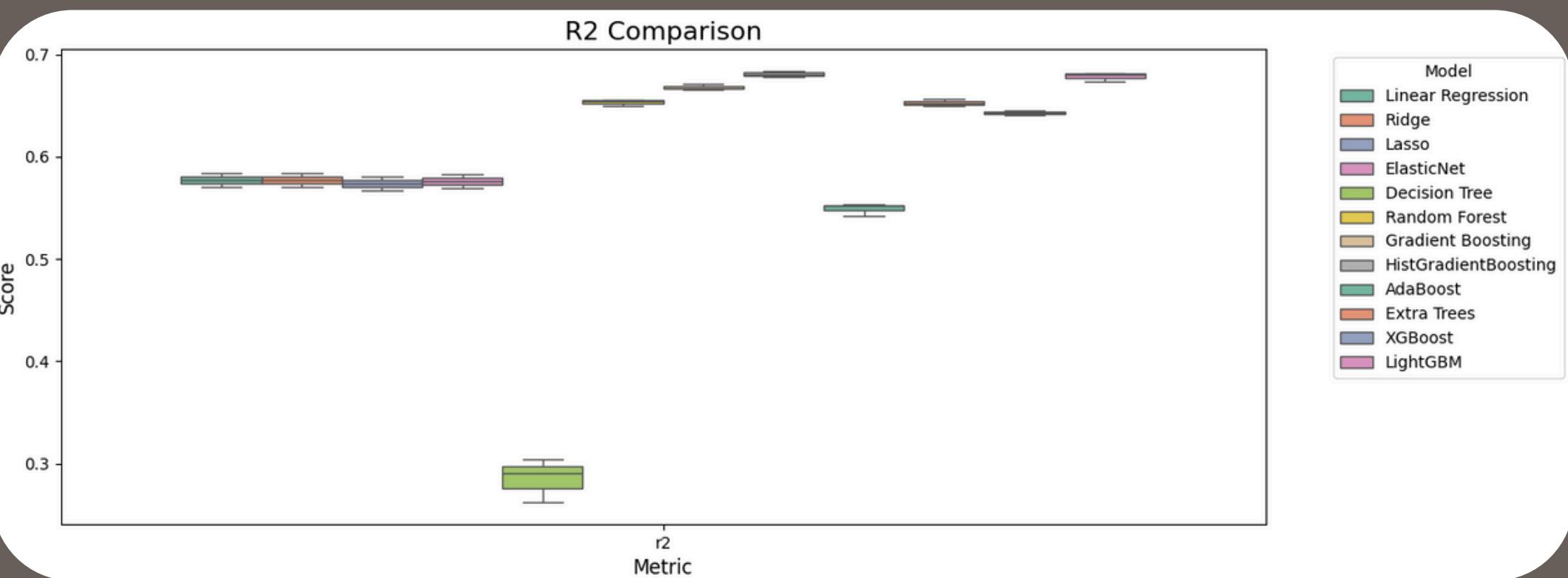
4. Setup Pre-processing  
Pipeline

```
# Preprocessor: Ordinal encode categoricals, passthrough numericals
preprocessor = ColumnTransformer(
    transformers=[
        ('cat', OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1), Categorical),
        ('num', 'passthrough', Numeric)
    ]
)

# Complete pipeline
CharWin_Pipeline = Pipeline(steps=[
    ('preprocessing', preprocessor),
    ('regressor', HistGradientBoostingRegressor(max_iter=100, random_state=42))
])

0.0s
```

5. Train Test Split  
6. Evaluate Models with sample



# CharWin

## 7. Fit with chosen model with full data

```
▶ 
#fit the model
CharWin_Pipeline.fit(X_train, y_train)
[75] ✓ 2.6s

...
Pipeline (i) (?)
  preprocessing: ColumnTransformer (i) (?)
    cat → num
      ▶ OrdinalEncoder (?) → passthrough
      ▶ HistGradientBoostingRegressor (?)
```

## 9. Test

```
TestCode("Cathy")
✓ 0.0s

Performance Summary: Cathy
Historical Average Rank : 4.30
Historical Win Rate (%) : 39.89%
Predicted Win Rate (%) : 25.95%
Projected Wins in 50 Games: 12 Wins
```

## 8. Evaluate

```
▶ 
#Score
y_pred = CharWin_Pipeline.predict(X_test)
print("R² Score:", round(r2_score(y_test, y_pred), 4))
print("RMSE:", round(mean_squared_error(y_test, y_pred, squared=False), 4))
[76] ✓ 0.5s

...
R² Score: 0.7135
RMSE: 1.196
```

# CharBuild

## 1. Load data set

```
bound method NDFrame.head of      userNum    nickname  gameId seasonId gameRank character
0      5426009      마사  46678118      31        2   Cathy
1      5296453  쿠로미여동생  46678118      31        1   Hisui
2      5162955      우주생명마음  46678118      31        5   Adina
3      5140049  只玩过卡拉彼丘  46678118      31        7 Ly Anh
4      5051669      초고교급마다기  46678118      31        7 Tsubame
...
311260     ...     ...
311261  1931832 Enakyira  48012635      0        2   Rozzi
311261  1306446 eternalqueues  48012635      0        2   Camilo
311262     396298 키르디  48012635      0        2   Celine
311263     102184      다다  48012635      0        1   Daniel
311264     71827  역까는당하는사람  48012635      0        1   Yuki

      weapon      itemWeapon      itemChest \
0      Dagger      Asura      The_Revenant
1 Two-handed_Sword Aurora_Longsword      Ao_Dai
2      Arcana      The_Moon      Blood_Cloak
3      Dagger      Maharaja      Dragon_Dobok
4      Shuriken      Petal_Torrent      Changpao
...
311260     ...
311261  Dual_Swords Nosferatu_-Crimson      Ao_Dai
311262      Throw      Chaser      Bikini
311263      Dagger Scarlet_Dagger_-Crimson      Tactical_Armor
311264 Two-handed_Sword      Phantom_Blade      Burgundy_47
...
311263     25832      298
311264     26804      8283

[11265 rows x 22 columns]
```

## 2. Clean data and format dataframe

▼ 0.0%

Total rows before dropping: 311265

Rows with missing traits: 41064 (13.19%)

Rows with all traits filled: 270201 (86.81%)

	AllTraits	BuildPerformance
0	Unwavering_Mentality_Cavalcade_Coin_Toss_Penny...	Good
1	Contempt_for_the_Weak_Anima_Reaper_Unwavering...	Good
2	Open_Wounds_Overwatch_Dismantle_Goliath_Anima...	Average
3	Frenzy_Carnivore_Painkiller_Dine_n'_Dash	Bad
4	Dismantle_Goliath_Anima_Reaper_Umwavering_Ment...	Bad
...	...	...
311252	Stopping_Power_Power_Crescendo_Dismantle_Golia...	Bad
311253	Dismantle_Goliath_Anima_Reaper_Stopping_Power...	Bad
311254	Thorn_Shackles_Penny_Pincher_Painkiller_Dine_n...	Good
311255	Painkiller_Dine_n'_Dash_Circular_System_Overwatch	Bad
311256	Stopping_Power_Power_Crescendo_Dismantle_Golia...	Average

[270201 rows x 24 columns]

# Model Pre-processing

## 3. Clean data and format dataframe

## 4. Establish target/features

```
#Features  
Categorical = ['character', 'weapon', 'itemWeapon', 'itemChest', 'itemHead', 'itemArm', 'itemLeg', 'AllTraits']  
Numerical = ['Kill', 'Assist', 'Death', 'TeamKill', 'Dmg_Player', 'Dmg_Monster']  
  
#Target  
target = 'BuildPerformance'  
  
< 0.0s
```

## 5. Split train test

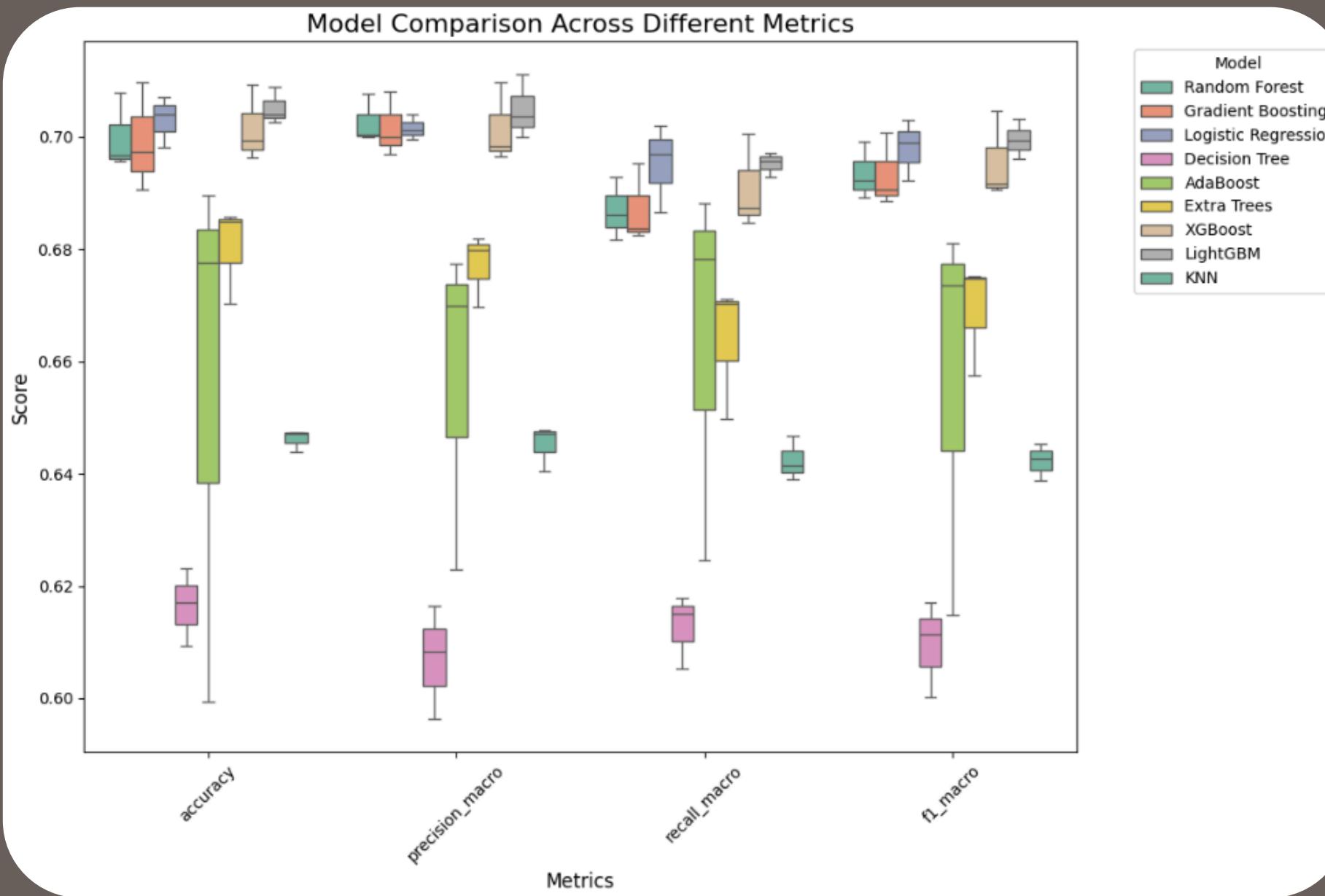
```
#Split dataset  
X = df[Categorical + Numerical]  
y = df[target]  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)  
  
< 0.3s
```

## 6. Setup Pre-processing Pipeline

```
#Build Preprocessing Pipeline  
  
#Categorical pipeline  
categorical_pipeline = Pipeline(steps=[  
    ('imputer', SimpleImputer(strategy='constant', fill_value='Unknown')),  
    ('encoder', OneHotEncoder(handle_unknown='ignore'))  
])  
  
#Numerical pipeline  
numerical_pipeline = Pipeline(steps=[  
    ('imputer', SimpleImputer(strategy='median')),  
    ('scaler', StandardScaler())  
])  
  
#Combine both pipelines  
preprocessor = ColumnTransformer(transformers=[  
    ('cat', categorical_pipeline, Categorical),  
    ('num', numerical_pipeline, Numerical)  
])
```

# Model Pre-processing

## 7. Evaluate Models with sample



## 8. Fit with chosen model with full data

```
#Train the model
character_build.fit(X_train, y_train)

✓ 5.5s
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.209404 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 2495
[LightGBM] [Info] Number of data points in the train set: 216160, number of used features: 945
[LightGBM] [Info] Start training from score -0.953810
[LightGBM] [Info] Start training from score -1.523215
[LightGBM] [Info] Start training from score -0.924524
```

The figure shows a code snippet for training a LightGBM model and a corresponding pipeline diagram. The code uses the `character_build` object to fit the model on the training data (`X_train, y_train`). The pipeline diagram illustrates the data processing steps: it starts with a `ColumnTransformer` containing two parallel paths for categorical (`cat`) and numerical (`num`) features. The `cat` path includes a `SimpleImputer` and a `OneHotEncoder`. The `num` path includes a `SimpleImputer` and a `StandardScaler`. Both paths converge to a final `LGBMClassifier`.

```
graph TD
    CT[preprocess: ColumnTransformer] --> SI_cat[SimpleImputer]
    CT --> OH[OneHotEncoder]
    CT --> SI_num[SimpleImputer]
    CT --> SS[StandardScaler]
    SI_cat --> OH
    OH --> SI_num
    SI_num --> SS
    SS --> LGBM[LGBMClassifier]
```

# Model Pre-processing

## 9. Evaluate

Classification Report:

	precision	recall	f1-score	support
Average	0.63	0.68	0.65	20820
Bad	0.73	0.64	0.68	11782
Good	0.81	0.80	0.80	21439
accuracy			0.72	54041
macro avg	0.72	0.71	0.71	54041
weighted avg	0.72	0.72	0.72	54041

## 10. Test

```
Build_test("Istvan", top_n=3)
```

✓ 0.0s

Recommended Build #1 for Istvan:

```
ItemWeapon : Spear_of_Longinus_-_Dawn
ItemChest  : Burgundy_47
ItemHead   : Laurel_Wreath
ItemArm    : Auto-arms
ItemLeg    : Blade_Boots
Traits     : Contempt_for_the_Weak, Anima_Reaper, Embolden, Cavalcade
```

Recommended Build #2 for Istvan:

```
ItemWeapon : Spear_of_Longinus
ItemChest  : Omertà
ItemHead   : Crusader_Helmet
ItemArm    : Auto-arms
ItemLeg    : Blade_Boots
Traits     : Contempt_for_the_Weak, Anima_Reaper, Embolden, Cavalcade
```

Recommended Build #3 for Istvan:

```
ItemWeapon : Spear_of_Longinus_-_Crimson
ItemChest  : Racing_Suit
ItemHead   : Chinese_Opera_Mask
ItemArm    : Bloodripper
ItemLeg    : Boots_of_Hermes
Traits     : Painkiller, Dine_n'_Dash, Stopping_Power, Recharger
```



# Conclusion

- System predicts builds & team outcomes based on historical match data
- The model is heavily skewed by current match trends
- Can players improve by learning from high performers
- Models are limited by dataset size and parameters
- Eternal Return's appeal is it's unpredictability.

# Future Work

1

**Consider additional performance metrics**

2

**Integrate into a GUI based program**

3

**Expand Algorithm to more games**



THANK YOU