Access Latest Slide Version

CPE 342 MACHINE LEARNING (1/2025)

PROJECT INSTRUCTION

"Now is the moment to apply what you've learned"

From one ***decision tree*** to a ***random forest***,
from a single model to an **ensemble of solutions**.

Table of Content

Back to table of content

👋 Welcome to the ML Project Challenge!

> *The project is designed to give you hands-on experience with real-world machine learning problems.*

What You'll Experience

- Working with messy, real-world datasets
- Building ML models 5 different tasks!!!
- Making decisions about model selection and evaluation
- Collaborating in a team like industry professionals

You'll Work in Groups of 5 !

- Real ML projects require teamwork
- Divide work efficiently across 5 tasks
- Learn from each other's approaches
- Practice collaboration skills

Back to table of content

Competition Enrollment

1 Team Register

- Fill in team info on the Google Sheet.
- Your "Team ID" will be the official team name.

Click here!

2 Join Competition

- Create or log in to your Kaggle account
- Register for the competition using this link

3 Team Setup

- Name your Kaggle team with the exact "Team ID" (e.g., ML001).
- Invite all teammates to the Kaggle team.

4 Participation

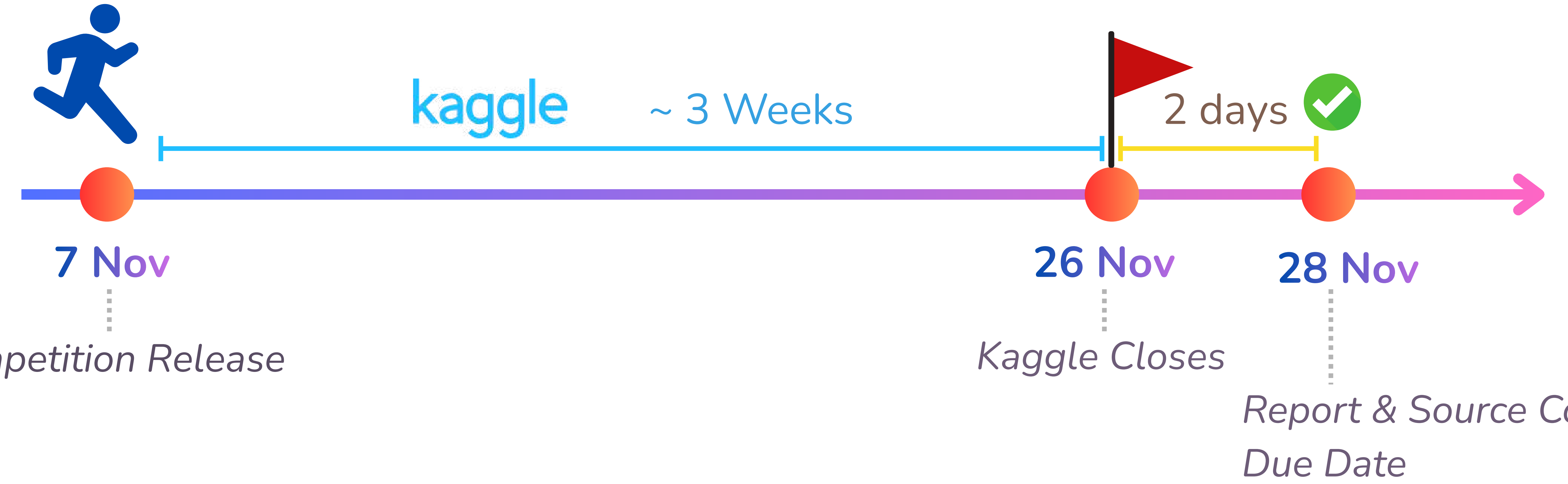
- Download the dataset
- Train your best model
- Upload your submission
- Check the leaderboard

Grading Criteria

• Performance on Kaggle leaderboard	70%
• Technical Report	20%
• Source Code	10%

"The technical report and source code submission will be posted on LEB2"

Timeline



Back to table of content

Submission Checklist

1. Kaggle

1 Join Competition

- Create or log in to your Kaggle account
- Register for the competition using this link

2 Team Register

- Fill in team info on the Google Sheet.
- Your "Team ID" will be the official team name.

3 Team Setup

- Name your Kaggle team with the exact "Team ID" (e.g., ML001).
- Invite all teammates to the Kaggle team.

4 Participation

Download the dataset, train your best model, and submit to Kaggle

2. Technical Report

Language: 🇬🇧 or 🇹🇭

1 Methodology

Organize subsections by task (Task 1–5)

- EDA findings
- Data Preprocessing
- Model Design
- Evaluation & Results
- Insights gained from model behavior
- Common mistakes or failed experiments with your solutions
- Domain interpretation of the results (link to game data and business impact)

2 Team Collaboration

- Role of each member
- Key takeaways from teamwork

3 Conclusion

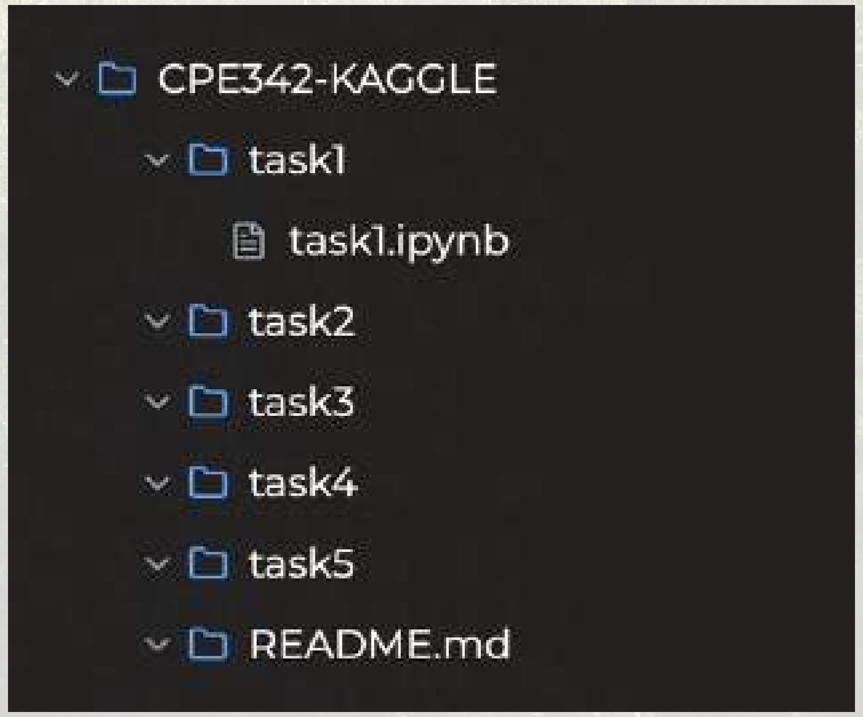
- Key lessons learned from model building and evaluation
- Insights on applying ML to the game industry
- Suggestions for future improvements or extensions

3. Source Code

1 Crate a Github Repository

Organize the code used to solve the 5 tasks into separate, clearly labeled folders within the repository (e.g., Task1, Task2, etc.).

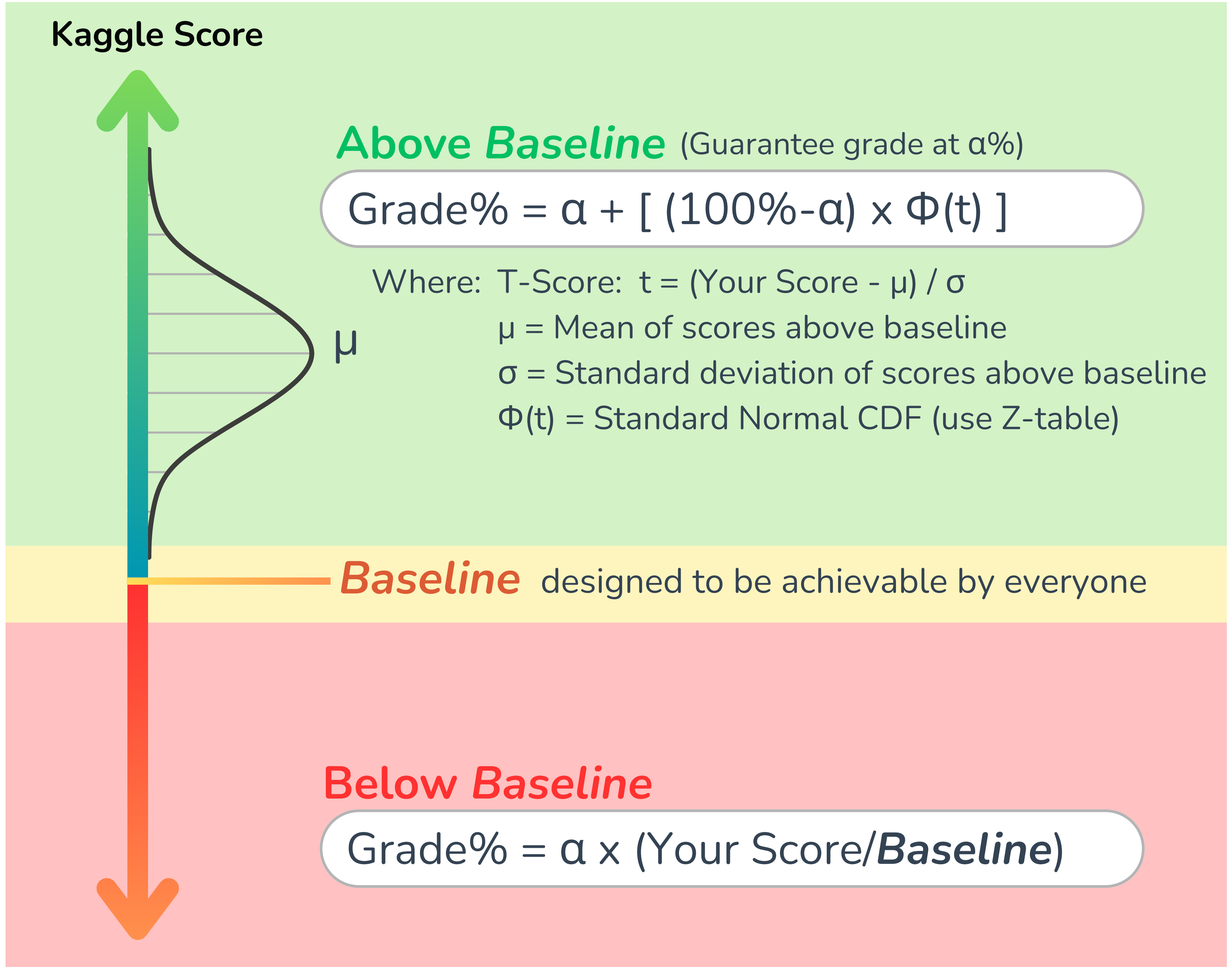
2 File Structure Example



```
CPE342-KAGGLE
  task1
    task1.ipynb
  task2
  task3
  task4
  task5
  README.md
```

Back to table of content

How Your Kaggle Score Converts to a Grade



Above Baseline (Guarantee grade at α%)

$$\text{Grade\%} = \alpha + [\,(100\% - \alpha) \times \Phi(t)\,]$$

Where: T-Score: $t = (\text{Your Score} - \mu) / \sigma$

μ = Mean of scores above baseline

σ = Standard deviation of scores above baseline

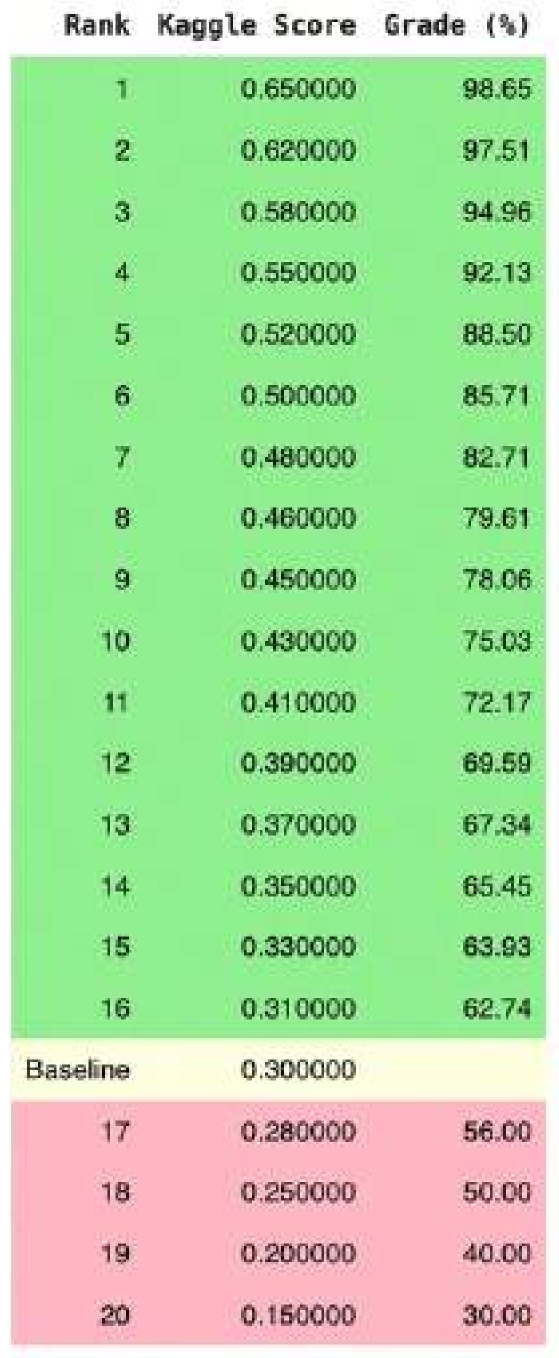
$\Phi(t)$ = Standard Normal CDF (use Z-table)

Baseline designed to be achievable by everyone

Below Baseline

$$\text{Grade\%} = \alpha \times (\text{Your Score}/\textbf{\textit{Baseline}})$$

Example Grading (α = 60%)



Rank	Kaggle Score	Grade (%)
1	0.650000	98.65
2	0.620000	97.51
3	0.580000	94.96
4	0.550000	92.13
5	0.520000	88.50
6	0.500000	85.71
7	0.480000	82.71
8	0.460000	79.61
9	0.450000	78.06
10	0.430000	75.03
11	0.410000	72.17
12	0.390000	69.59
13	0.370000	67.34
14	0.350000	65.45
15	0.330000	63.93
16	0.310000	62.74
Baseline	0.300000	
17	0.280000	56.00
18	0.250000	50.00
19	0.200000	40.00
20	0.150000	30.00

Back to table of content

Need a Hand? We're Here to Help!

Your Teaching Assistants (TAs) are your first stop for questions, road bumps, and feedback. We're excited to support your team!

P' Chokun
email: chotanansub.soph@kmutt.ac.th

P' Noina
email: phanita.siri@kmutt.ac.th

Time to reveals the project...

Karena Player Intelligence System

ML meets Game Industry

Free Fried

7-11 Knights

FiveN

PUBE Mobile

Roblock

This isn't just about algorithms - it's about solving business problems with data.

Situation

Your team consists of **Machine Learning Engineers** at **Karena Thailand** - a gaming platform company (similar to Garena [1]) that develop and publishes multiple online game for example PUBE Mobile, Free Fried, FiveN, Roblock, 7-11 Knight, etc

Karena

Publishing games

PUBE Mobile

Free Fried

7-11 Knights

Roblock

FiveN

[1] A famous game developer and publisher company see more info

The Challenge

Karena needs an AI-powered Player Intelligence System to

Maintain game integrity across all platforms

Understand player behavior and preferences

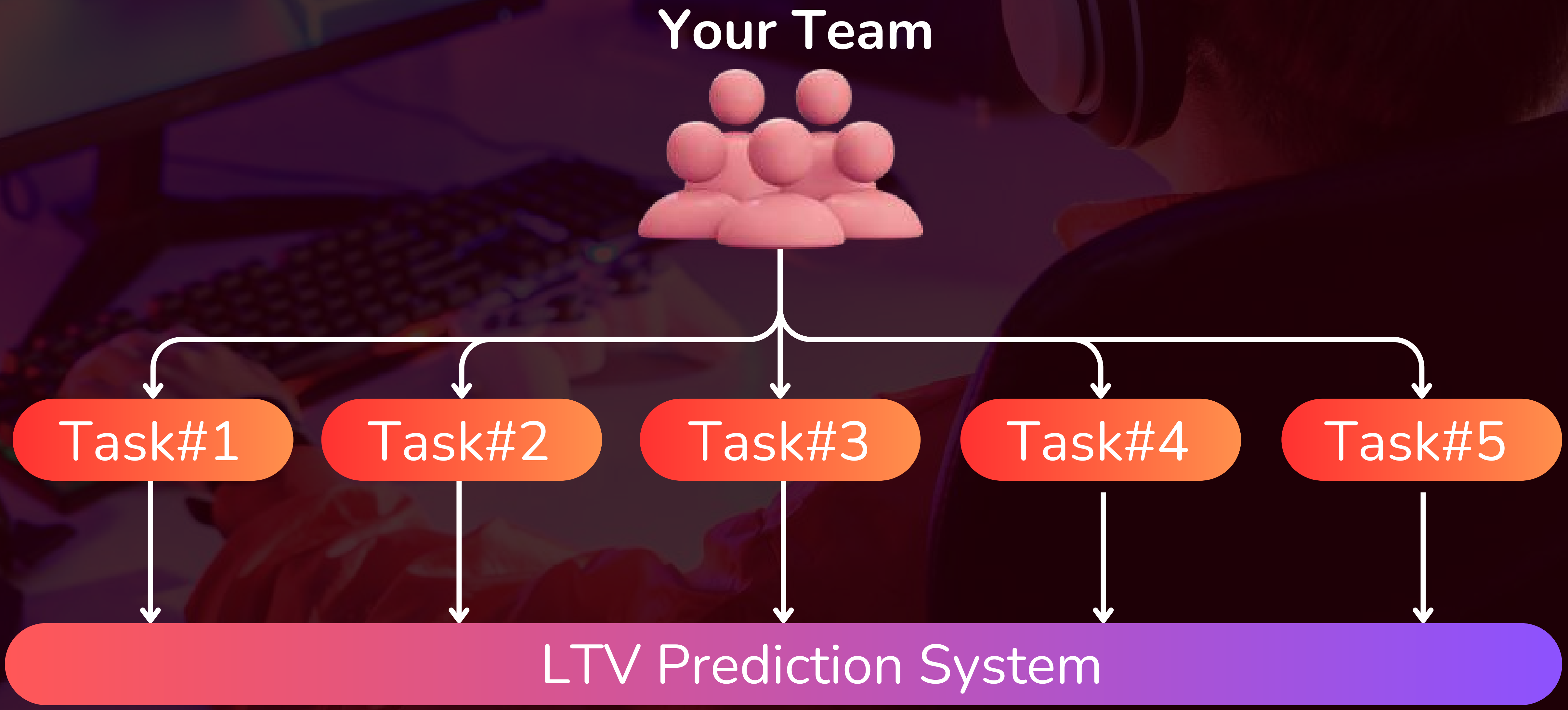
Optimize monetization strategies

Improve player retention and lifetime value

Back to table of content

Your Team's Mission

Build a comprehensive ML pipeline with 5 tasks that feed into a final Player Lifetime Value (LTV) Prediction System to help the business make data-driven decisions.



Back to table of content

What You'll Build

Your team will build a complete ML project with 5 interconnected models:

1. Binary Classification
2. Muti-Class Classification
3. Regression
4. Image Classification
5. Anomaly Detection (Unsupervised)

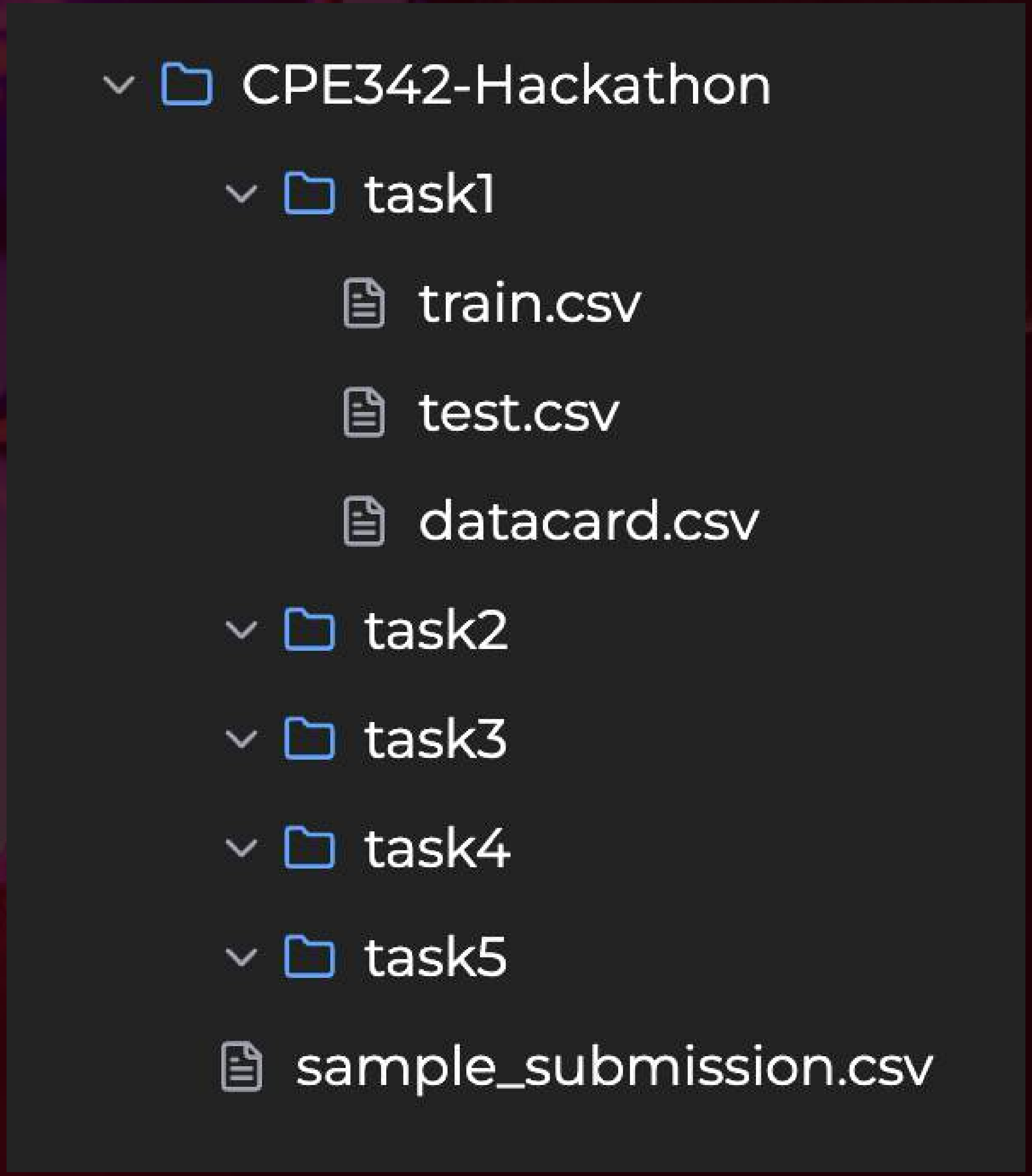
Back to table of content

Instruction Structure

In each task instruction consists of

- Situation
- Mission
- Task Category
- Evaluation Metric
- Example Features
- Hints & Best Practices

Dataset Structure: kaggle



```
CPE342-Hackathon
    task1
        train.csv
        test.csv
        datacard.csv
    task2
    task3
    task4
    task5
    sample_submission.csv
```

Back to table of content

Competition Rules

Data

- Use provided datasets only
- Train and Val sets allowed for training
  - ✅ You can combine Train and Val for training
  - ❌ No Test set usage for training purpose pseudo-labeling, fine-tuning, etc.)

Collaboration

- Work only with your team
- Concept discussions OK, but no sharing code or predictions
- All members must contribute meaningfully

Model Development

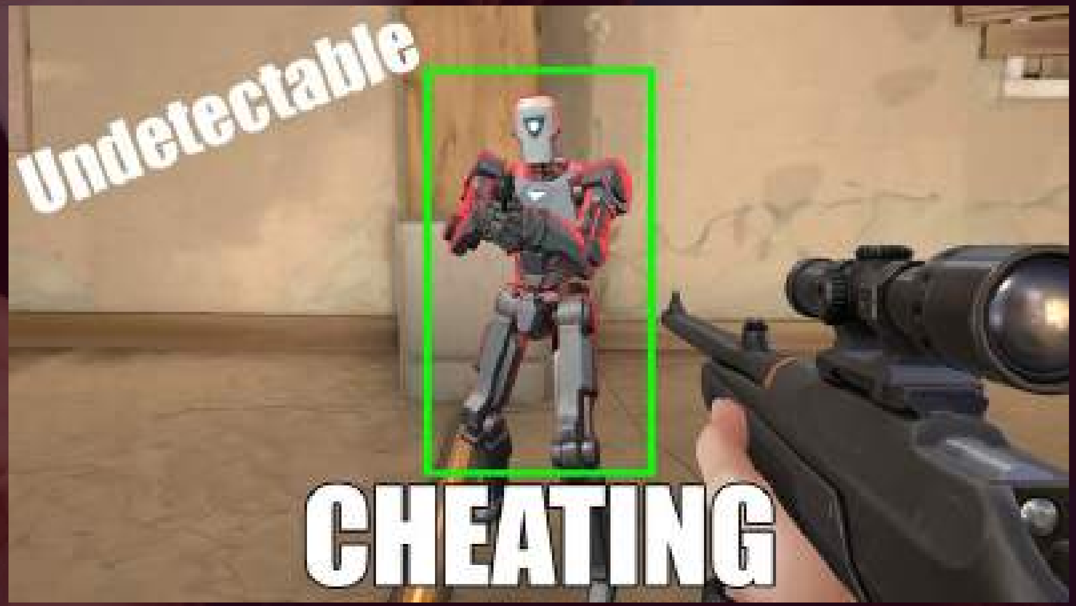
- Task 4 (Image-Classifcation): Pre-trained models allowed
- Tasks 1, 2, 3, 5: Train from scratch
- Feature engineering & ensembling: Encouraged
- AutoML
- ❌ Prohibited: AutoML framework
- ✅ Allowed: Hyperparameter optimization tools

Back to table of content

Task 1: Anti-Cheat Pre-Filter (1/4)

Situation:

Karena Thailand receives over 10,000 cheat reports daily across all games, but the manual review team can only handle 500 cases per day, creating a massive backlog of 45,000+ pending reports. Many reports are false positives from frustrated players who lost matches fairly, while real cheaters continue to damage game experience and drive legitimate players away. The current automated system has a 70% false positive rate, meaning innocent players get flagged and sometimes banned, leading to customer service nightmares and permanent player loss. Meanwhile, sophisticated cheaters who use subtle aimbots [1] or wallhacks [2] slip through undetected. Player retention has dropped 15% in competitive game modes due to cheating complaints, directly impacting revenue by approximately ฿15 million daily. The company needs an intelligent pre-filtering system that can accurately identify high-probability cheaters for priority manual review, reducing workload while catching real threats



24 KILLS
WALLHACK + AIMBOT CHEATER!! | 24 KILLS SOLO VS SQUAD | PUBG...
Watch >

[1] Aimbot: Cheating software that automatically aims at enemies
[2] Wallhack: Exploit allowing players to see through walls

Back to table of content

Task 1: Anti-Cheat Pre-Filter (2/4)

Your Task:

Build a binary classifier to automatically pre-filter reports and prioritize high-confidence suspicious accounts for manual review.

ML Task:

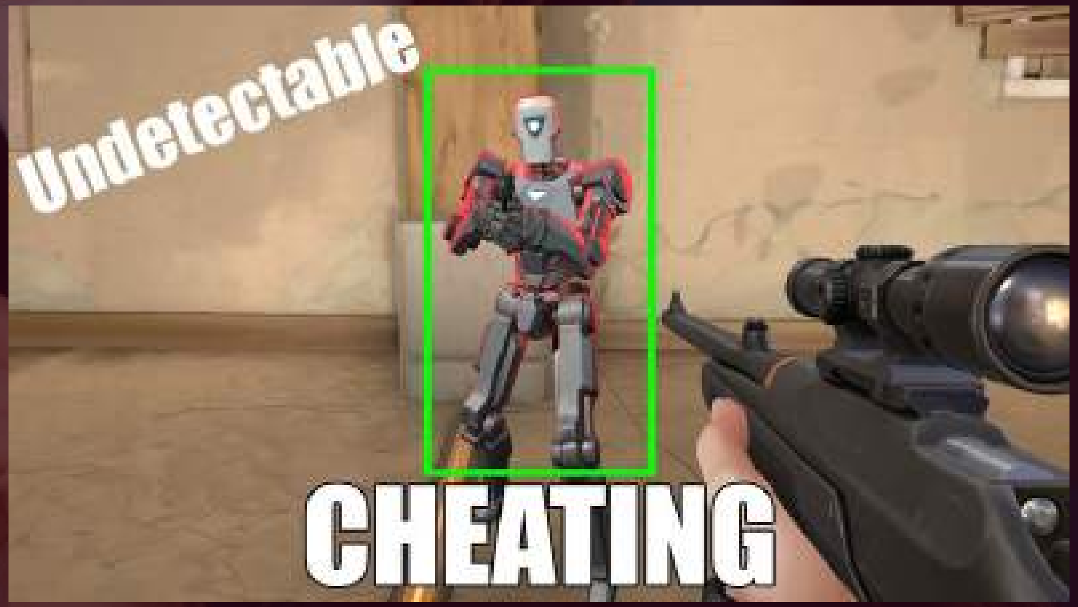
Binary Classification

Evaluation Metric: F2

$$F_2 = 5 \cdot \frac{\text{Precision} \cdot \text{Recall}}{4 \cdot \text{Precision} + \text{Recall}}$$

Task Structure:

```
.
└── task1
    ├── train.csv
    ├── test.csv
    └── datacard.txt
```



WALLHACK + AIMBOT CHEATER!! | 24 KILLS SOLO VS SQUAD | PUBG...

Back to table of content

Task 1: Anti-Cheat Pre-Filter (3/4)

The dataset contains 33 features
consists of 32 predictive features plus 1 target variable.

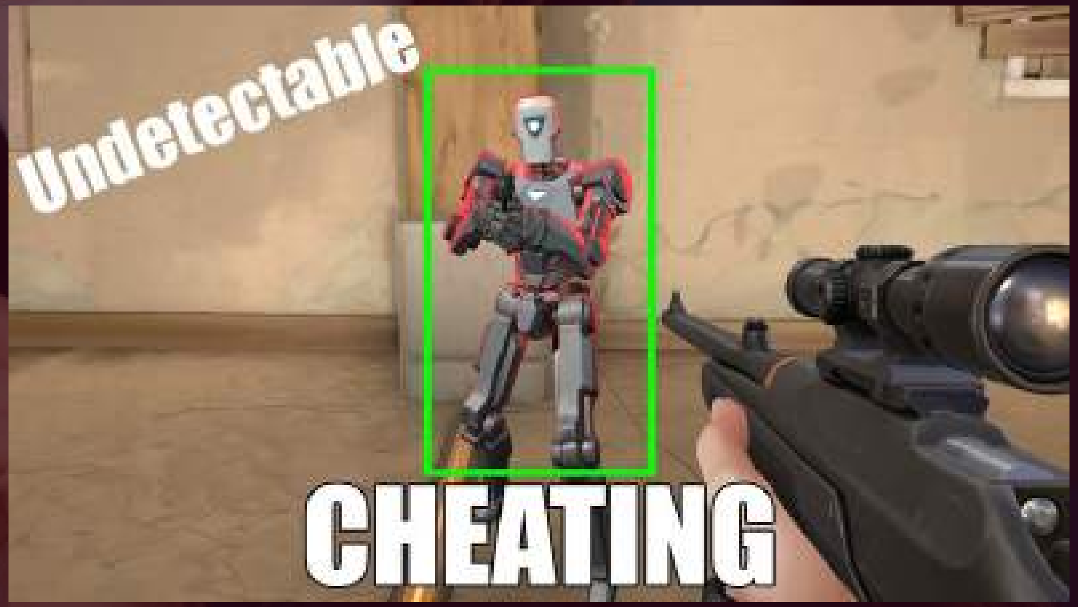
Target Variable

is_cheater (0=Legitimate, 1=Cheater)

Features Example

- **kill_death_ratio** (float): Ratio of kills to deaths across recent matches
- **headshot_percentage** (float): Percentage of kills that are headshots
- **win_rate** (float): Overall win rate percentage across all matches
- **accuracy_score** (float): Shot accuracy percentage (hits/shots fired)
- **reaction_time_ms** (float): Average reaction time in milliseconds

For complete feature descriptions please refer to the data card.
task1/datacard.txt

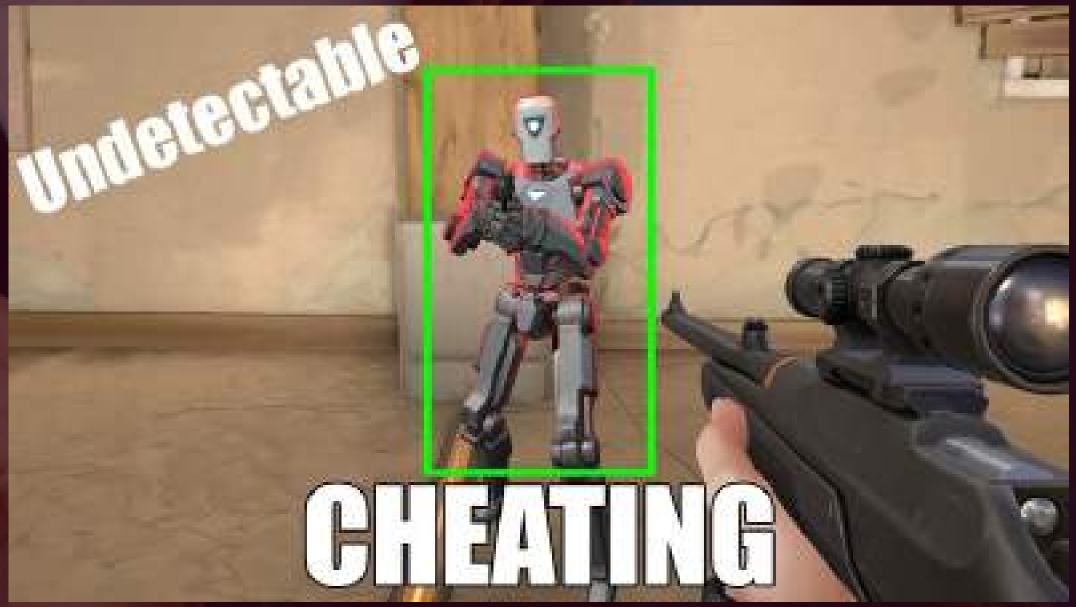


WALLHACK + AIMBOT CHEATER!! | 24 KILLS SOLO VS SQUAD | PUBG...

Task 1: Anti-Cheat Pre-Filter (4/4)

💡 Hints & Best Practices

- Dealing with Missing Value is crucial. missing data can distort results. Use imputation or remove incomplete rows wisely see the example
- Handle Class Imbalance! Imbalanced data can bias predictions. Try resampling, class weights, or SMOTE. see the example
- Feature Engineering is Key! try create some interaction features
  - e.g. skill_consistency = game_sense / accuracy
- Cross-Validation for Reliable Performance see the example
- Try decision threshold tuning see the example
- Ensemble for Robustness see the example
- The hint in Task 2 could be useful—make sure to check it out as well!



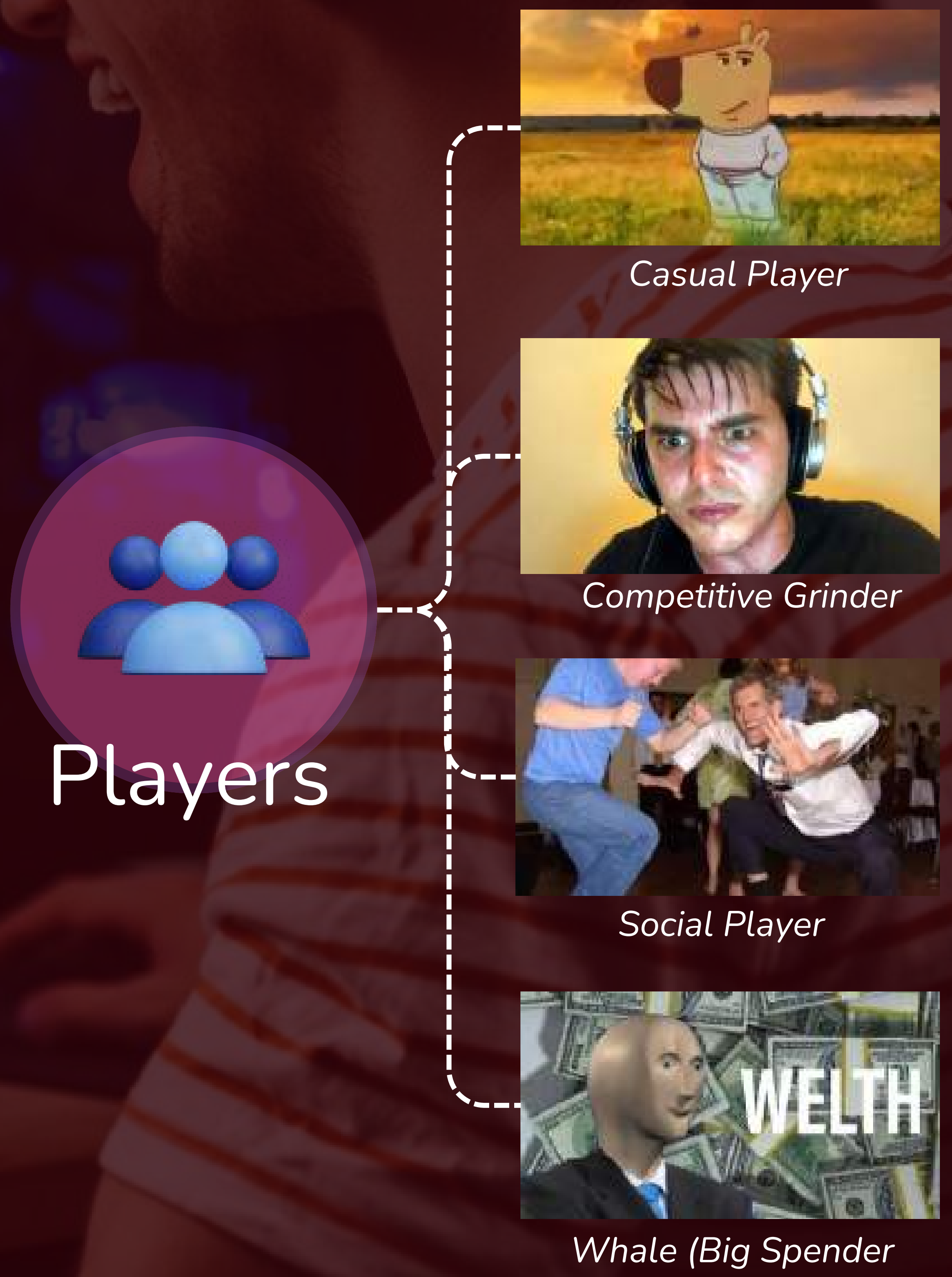
24 KILLS
WALLHACK + AIMBOT CHEATER!! | 24 KILLS SOLO VS SQUAD | PUBG...
Watch >

Back to table of content

Task 2: Player Segment Classification (1/4)

Situation:

Karena Thailand's marketing team currently runs generic campaigns across all players, resulting in extremely low engagement rates of only 2-5%. They spend approximately ฿60 million monthly on promotional campaigns, but most fall flat because they're not personalized. For example, competitive esports tournament announcements are shown to casual mobile players who just want to relax, cosmetic item sales are advertised to players who have never made a purchase, and FiveN roleplay event invitations go to hardcore PUBE competitive grinders. This misalignment wastes massive advertising budgets and annoys players with irrelevant content. The top 5% of players (whales) generate 60% of total revenue, but the company can't identify them reliably using simple metrics like playtime alone. A high-playtime player could be a competitive grinder seeking glory, a social player enjoying friend time, or a whale collecting items—each needs completely different messaging. Proper segmentation could increase campaign effectiveness by 3-4x, translating to millions of baht in additional revenue and improved player satisfaction.



Task 2: Player Segment Classification (2/4)

Your Task:

Classify each player into one of four behavioral segments to enable targeted marketing and personalized experiences.

ML Task:

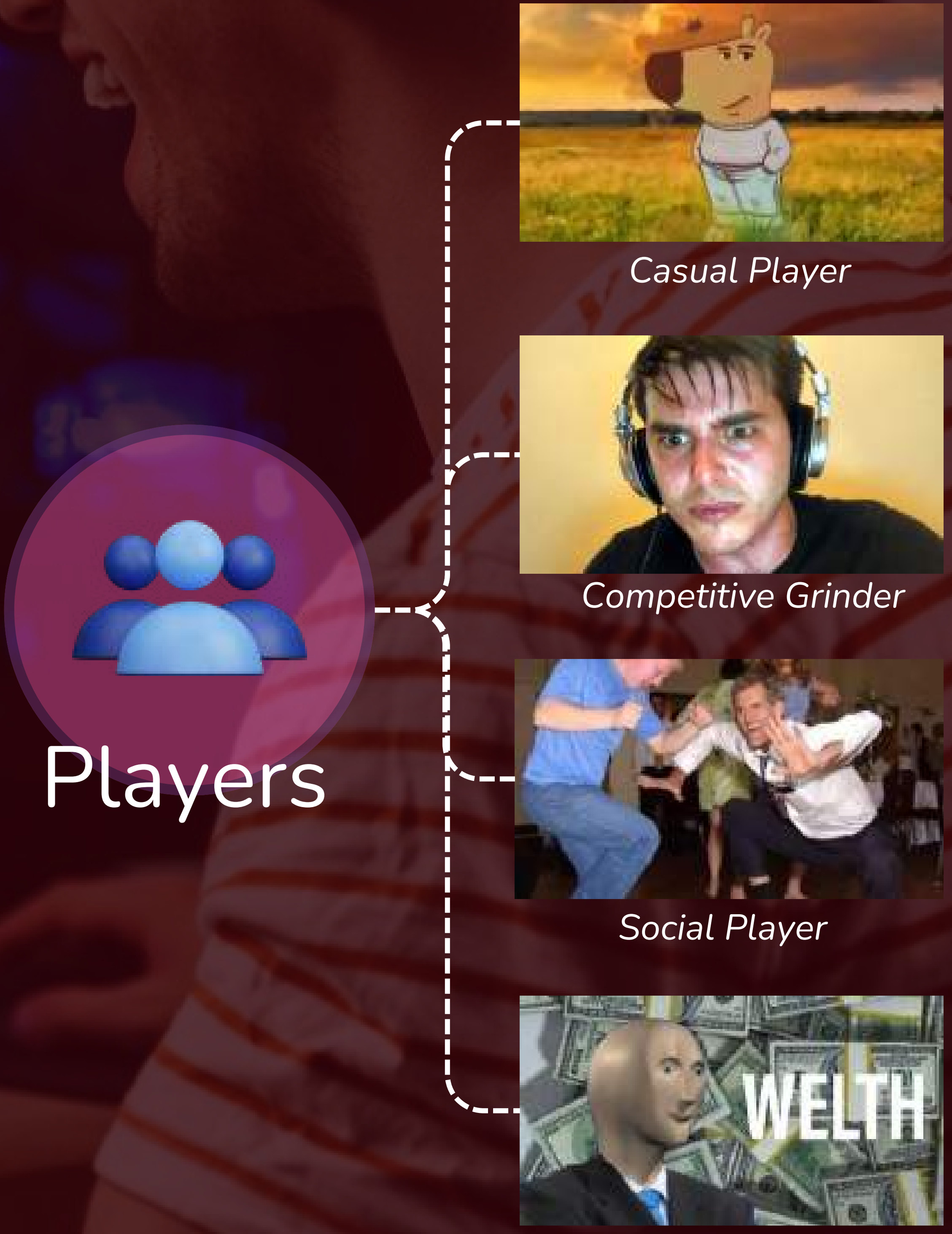
Multi-class Classification

Evaluation Metric: F1

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Project Structure:

```
.
└── task2
    ├── train.csv
    ├── test.csv
    └── datacard.txt
```



Task 2: Player Segment Classification (3/4)

The dataset contains 45 features
consists of 44 predictive features plus 1 target variable.

Target Variable

segment (0=Casual, 1=Grinder, 2=Social, 3=Whale)

Features Example

- **play_frequency** (float): Average days per week player logs in
- **avg_session_duration** (float): Average play session duration in minutes
- **total_spending_thb** (float): Lifetime spending on in-game purchases
- **avg_monthly_spending** (float): Average monthly spending
- **friend_count** (float): Number of in-game friends
- **team_play_percentage** (float): Percentage of matches played in groups

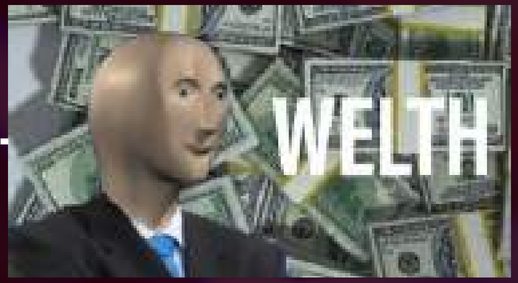
For complete feature descriptions please refer to the data card.
task2/datacard.txt

Players

Casual Player

Competitive Grinder

Social Player



Whale (Big Spender

Back to table of content

Task 2: Player Segment Classification (4/4)

Hints & Best Practices

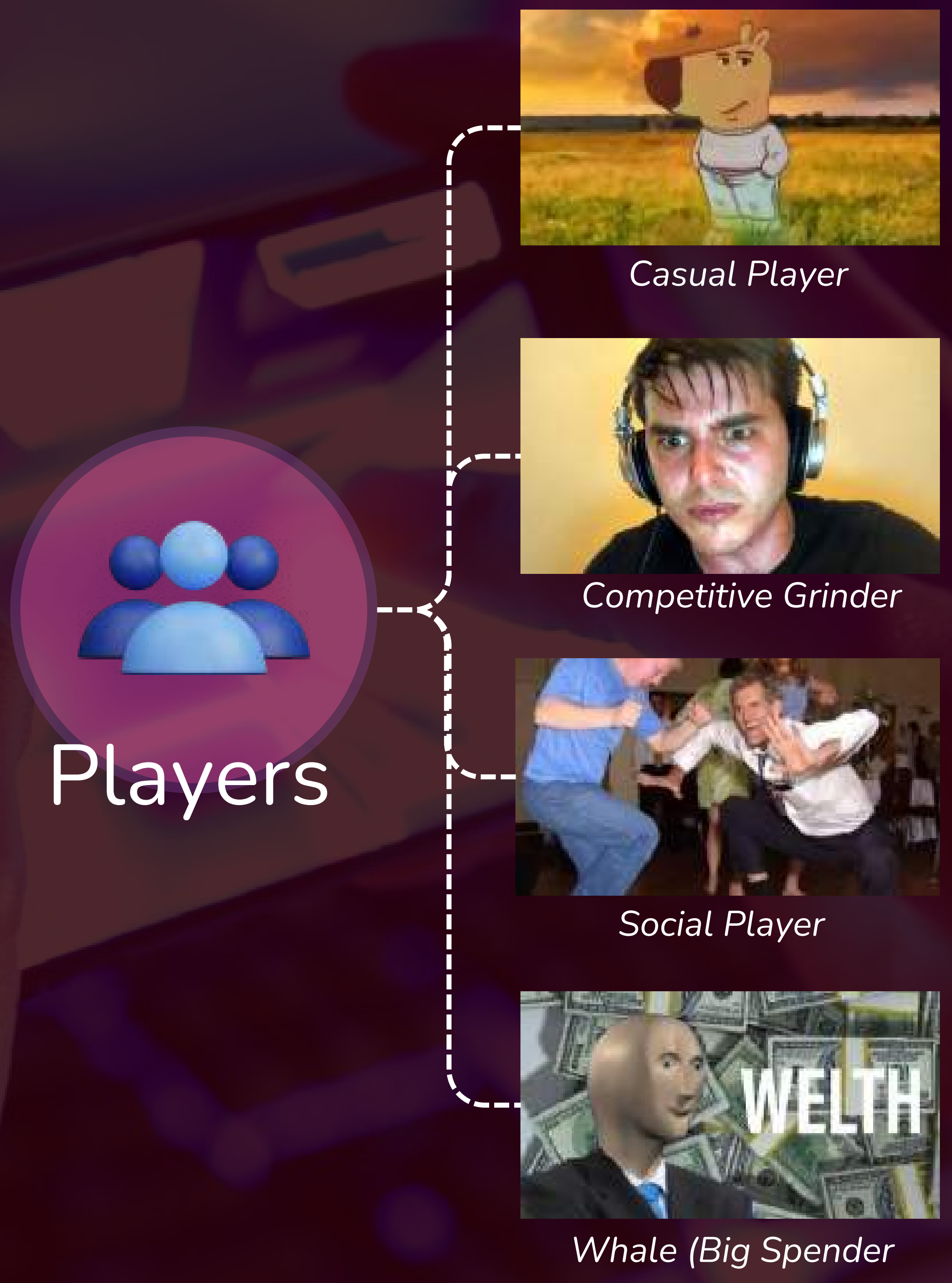
Since Task 1 already includes some binary classification tips, you can leverage those as well. In the Task 2 hint, we'll provide more examples and a notebook for multi-class classification.

- Missing Value Handling Source1 Source2
- Class Imbalance & SMOTE Source1 Source2 Source3
- Feature Engineering Source
- Cross-Validation Source
  - Use Stratified K-Fold to preserve class distribution
- Ensemble Methods Source1 Source2

Advanced Models to Try

- XGBoost Source
- LightGBM Source
- CatBoost Source

Hint: These models are powerful and even more robust when you stack them together! 🙂



Back to table of content

Task 3: Player Monthly Spending Prediction (1/4)

Situation:

Karena Thailand's finance team struggles with revenue forecasting, consistently missing quarterly targets by 35-40%, which impacts budgeting, hiring decisions, and investor confidence. Without accurate player-level spending predictions, multiple departments are operating blindly. The VIP support team of 20 people provides white-glove service but doesn't know which players actually deserve priority treatment—last quarter they spent significant resources on 150 high-playtime players who collectively spent only ฿12,000, while a whale who spent ฿255,000 waited 3 days for ticket response and subsequently left for a competitor. The marketing team sends promotional discount codes randomly, often giving 50% off to whales who would have paid full price anyway, resulting in ฿6 million+ monthly revenue loss. Customer service wastes time on players unlikely to ever monetize while high-value customers receive generic treatment. Additionally, the company needs to optimize their limited-time event scheduling and new content releases based on when high-spending players are most active. Accurate spending predictions would enable proper resource allocation, targeted offers, revenue forecasting, and VIP program optimization across the organization.

Back to table of content

Task 3: Player Monthly Spending Prediction (2/4)

Your Task:

Predict how much (in THB) each player will spend on in-game currency and items in the next 30 days.

ML Task:

Regression

Evaluation Metric: Normalized MAE

$$\text{Normalized MAE} = \frac{1}{1 + \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{\sum_{i=1}^{N} y_i}}$$

Project Structure:

```
.
└──── task3
     ├──── train.csv
     ├──── test.csv
     └──── datacard.txt
```

Back to table of content

Task 3: Player Monthly Spending Prediction (3/4)

The dataset contains 33 features
consists of 32 predictive features plus 1 target variable.

Target Variable

spending_30d (Continuous, THB)

Features Example

- **friend_count** (float): Number of in-game friends
- **event_participation_rate** (float): Percentage of limited-time events joined
- **daily_login_streak** (float): Current consecutive daily login streak
- **avg_session_length** (float): Average session duration in minutes
- **historical_spending** (float): Historical spending over past periods

For complete feature descriptions please refer to the data card.

task3/datacard.txt

Back to table of content

Task 3: Player Monthly Spending Prediction (4/4)

💡 Hints & Best Practices

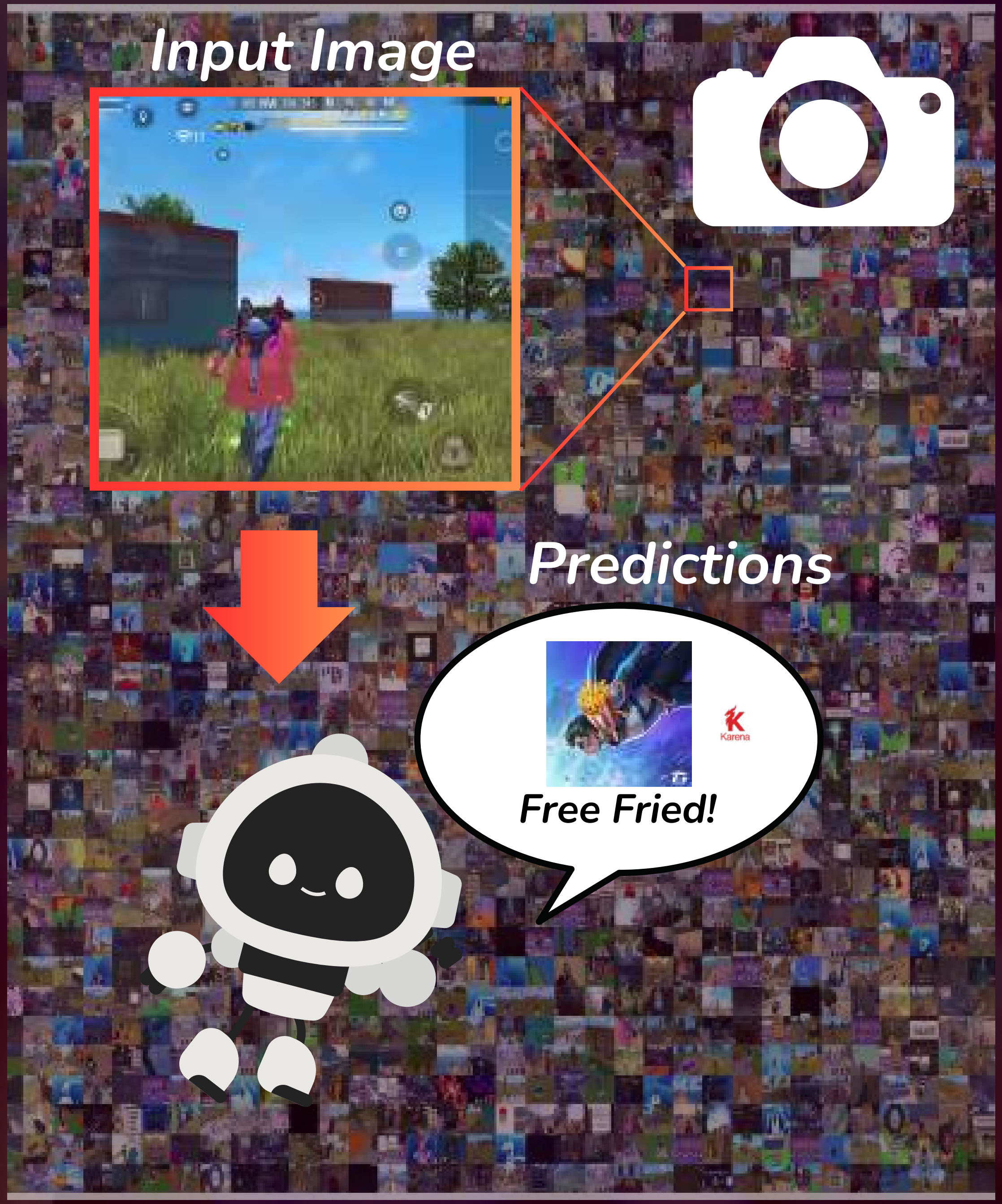
- Zero-Inflated Data Source
  - Hint: Two-stage ??? --> (1) Will spend? (2) How much?
- Feature Engineering Source1 Source2
- Log Transformation Source
- Advanced Models to Try
  - XGBoost Regressor Source
  - LightGBM Regressor Source
  - CatBoost Regressor Source
  - Ensemble/Stacking Source
- Common Mistakes

Back to table of content

Task 4: Game Title Detection (1/4)

Situation:

Karena Thailand operates five distinct games but can't automatically identify which game is being played from screenshots. The customer support team receives 2,000 daily tickets with images and wastes 10+ minutes per ticket figuring out which game before routing to specialists. The "Play Multiple Games for Rewards" campaign can't verify 50,000+ screenshot submissions, forcing manual checks of only 3% (leading to fraud). Players post 100,000+ gameplay images weekly across platforms that can't be auto-tagged for contests or moderation. Marketing can't track which games are trending without manual review. An automated detection system would enable faster support, verified campaigns, scalable content moderation, and cross-game insights.



Back to table of content

Task 4: Game Title Detection (2/4)

Your Task:

Build an image classifier that identifies which Karena game is being played from a single screenshot.

ML Task:

Image Classification (Multi-Class)

Evaluation Metric: F1 (Macro)

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Project Structure:

```
.
└── task4/
    ├── train/
    ├── eval/
    ├── test/
    ├── train.csv
    ├── test.csv
    └── datacard.txt
```

PUBE Mobile

Free Fried

7-11 Knights

Roblock

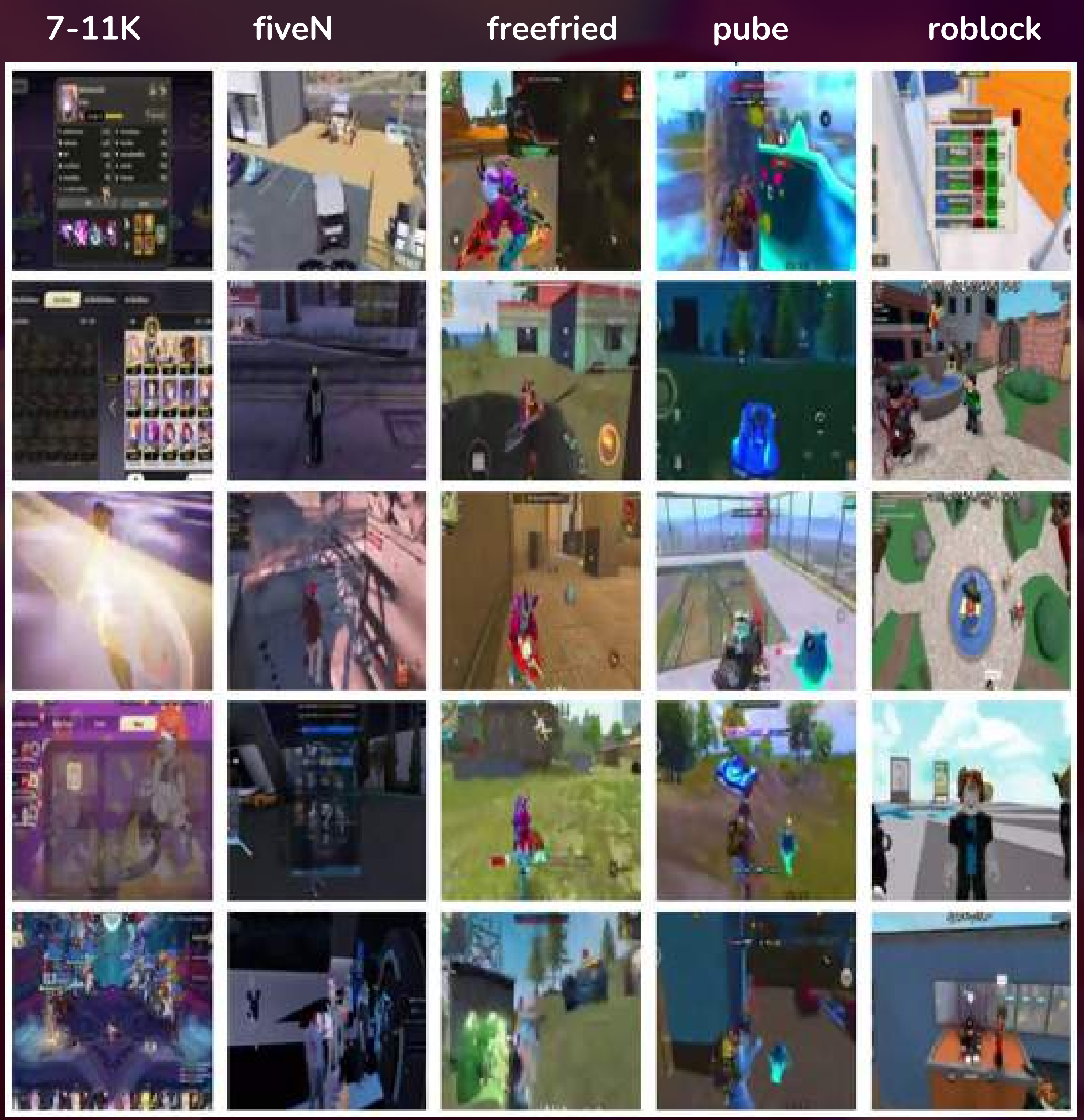
FiveN

Back to table of content

Task 4: Game Title Detection (3/4)

Data Description:

- Image dataset: Gameplay screenshots (144p and centered crop to 230x120)
- Sources: Live streams, player submissions, community forums, support tickets
- Variety: Different maps, game phases, weather conditions, UI states
- Data Class (5 Classes)
    - 0: 7-11K
    - 1: fiveN
    - 2: freefried
    - 3: pube
    - 4: roblock



Back to table of content

Task 4: Game Title Detection (4/4)

Hints & Best Practices

- Image Classification Basics Source Source
- Class Imbalance in Images Source
- Data Augmentation Source
- Cross-Validation for CNNs Source
- Ensemble Methods Source

Pretrained Models to Try

- VGG16/VGG19
- Vision Transformer (ViT)
- Swin Transformer

You may need a GPU for this task. Colab and Kaggle offer free GPU options (such as T4)

NOTE: Kaggle provides a predictable quota of approximately 30 hours per week resetting weekly, while Colab's access is dynamic, with no fixed limit and high usage leading to temporary restrictions.

PUBE Mobile

Free Fried

7-11 Knights

Roblock

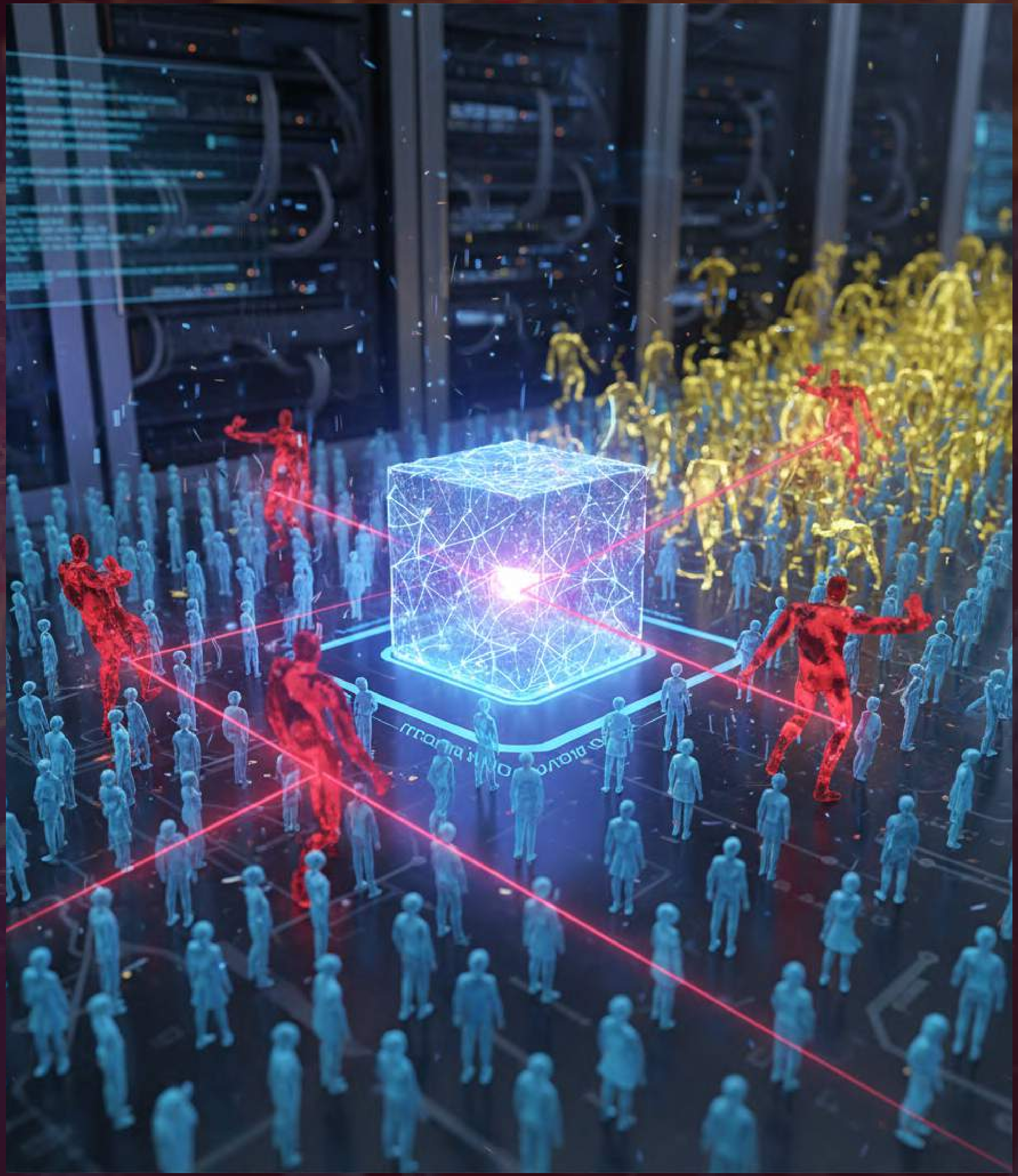
FiveN

Back to table of content

Task 5: Account Security Monitoring (1/4)

Situation

Karena Thailand faces over 5,000 confirmed account thefts monthly (actual number likely 2x higher), making it the #1 customer support issue consuming 60% of team resources. Beyond theft, sophisticated account selling rings destroy competitive integrity, bot farms operate 24/7 causing game economy inflation, and players exploit VPNs for regional pricing (70% discounts), costing millions of baht annually. The current rule-based security system has 60% false positive rate—flagging legitimate travelers while missing smart attackers who wait 24 hours between actions. Simple thresholds flag players who legitimately improve while missing gradual botting patterns. The security team drowns in alerts while real threats slip through. The company needs an intelligent unsupervised anomaly detection system that learns normal behavior patterns and flags truly unusual activities without rigid rules or labeled data.



Back to table of content

Task 5: Account Security Monitoring (2/4)

Your Task:

Build an unsupervised anomaly detection model to flag accounts exhibiting unusual behavior patterns. You must use anomaly detection techniques (Isolation Forest, Autoencoder, etc.) to identify outliers without relying on labeled training data.

ML Task:

Anomaly detection (Unsupervised)

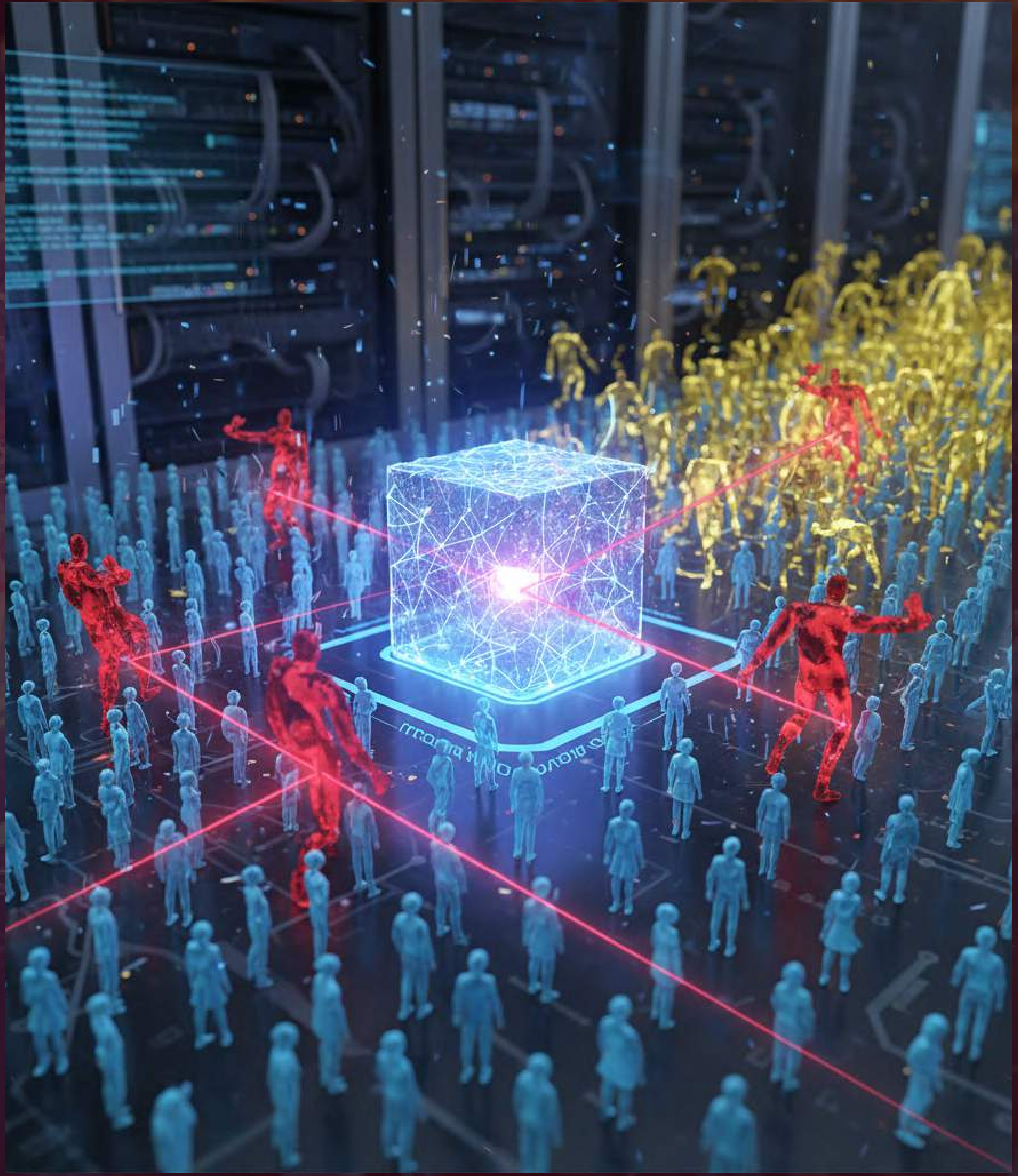
Evaluation Metric: F3

$$F_3 = 10 \cdot \frac{\text{Precision} \cdot \text{Recall}}{9 \cdot \text{Precision} + \text{Recall}}$$

Project Structure:

```
.
└── task5/
    ├── test.csv
    └── datacard.txt
```

Since it's unsupervised, we don't have a labeled training set!



Back to table of content

Task 5: Account Security Monitoring (3/4)

The dataset contains 33 features
consists of 32 predictive features plus 1 target variable.

Target Variable

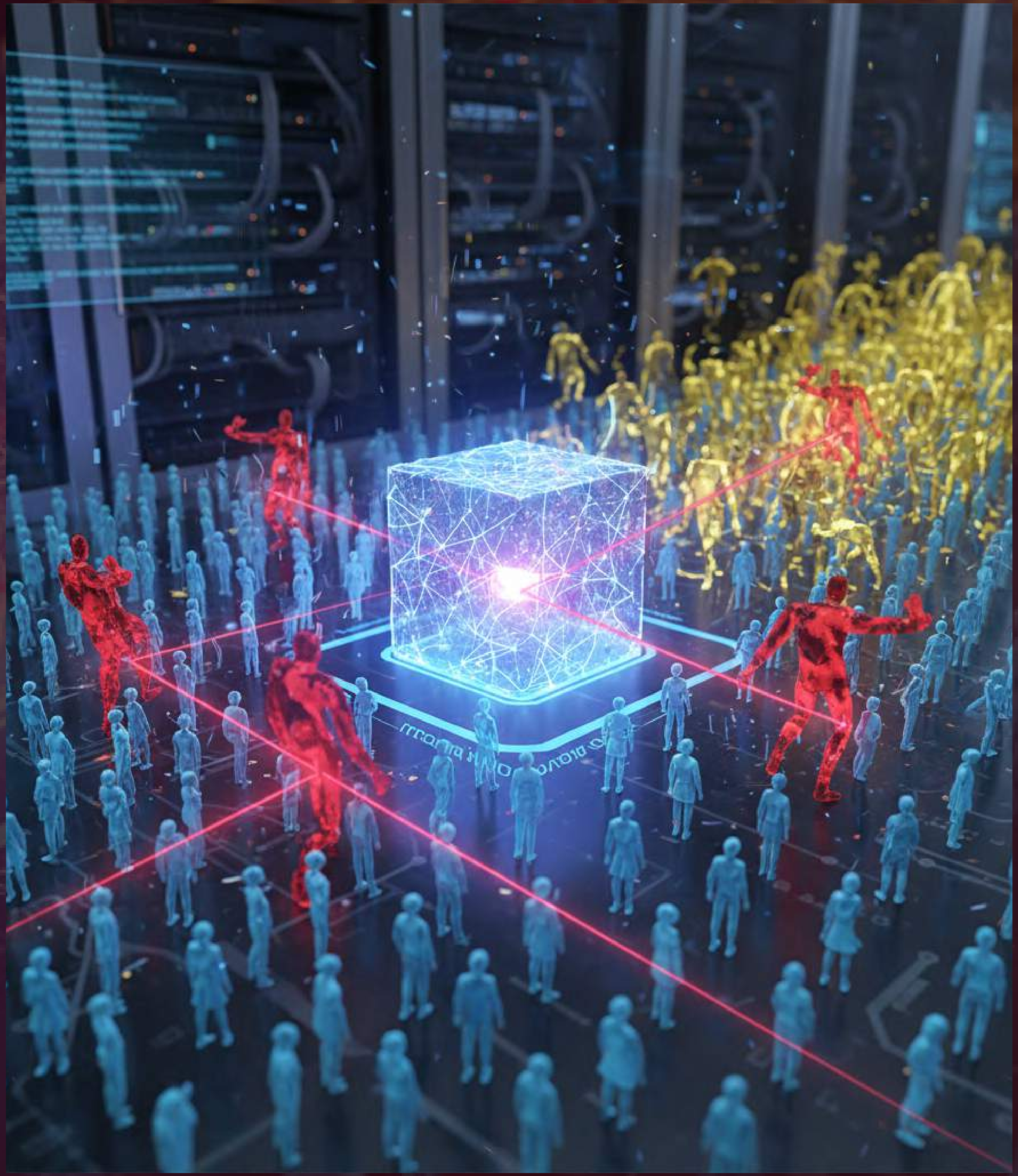
is_anomaly (0=Normal, 1=Anomalous)

Note: Features have suffixes _1, _2, _3, _4 representing 4 consecutive time periods

Features Example

- **login_count_1/2/3/4** (float): Number of login events in each time period
- **login_lat_1/2/3/4** (float): Latitude coordinate of login locations
- **login_lon_1/2/3/4** (float): Longitude coordinate of login locations
- **location_changes_1/2/3/4** (float): Number of significant location changes
- **device_count_1/2/3/4** (float): Number of unique devices used

For complete feature descriptions please refer to the data card.
task5/datacard.txt

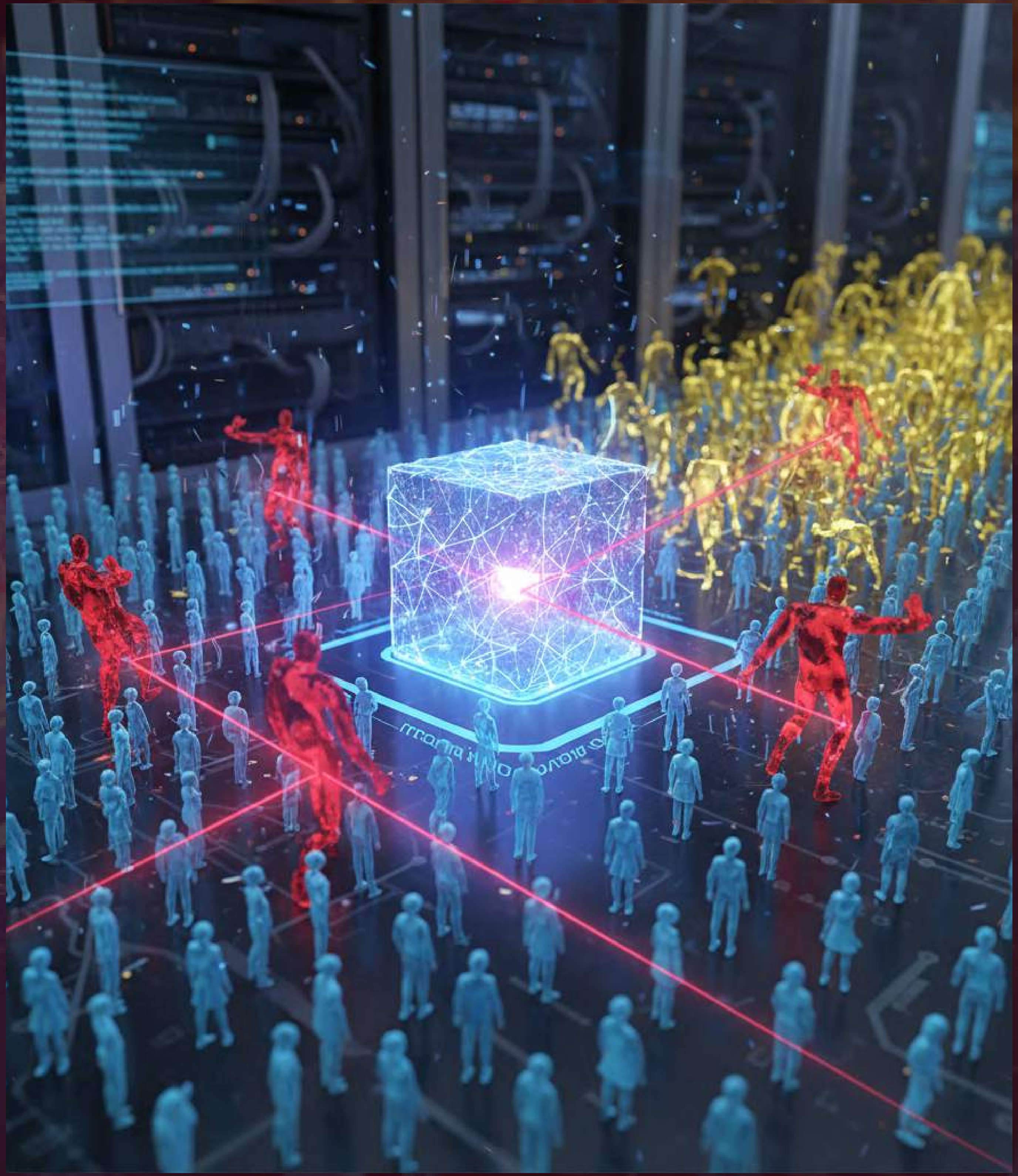


Back to table of content

Task 5: Account Security Monitoring (4/4)

Hints & Best Practices

- Well knowns unsupervised anomaly detection model
  - Isolation Forest Source
  - One-Class SVM Source
  - Auto-Encoder Source
- Even if it's unsupervised, you can still ensemble multiple models, but you may need to use a voting technique. Read more about soft and hard voting Source



Submission Format

Fill your answer in sample_submission.csv

sample_submission.csv

id	task1	task2	task3	task4	task5
ANS00001	0	2	0	freefried	1
ANS00002	1	3	10000	fiveN	0
ANS00003					
…	…	…	…	…	…
ANS25889					

25889

5

(25889,6)

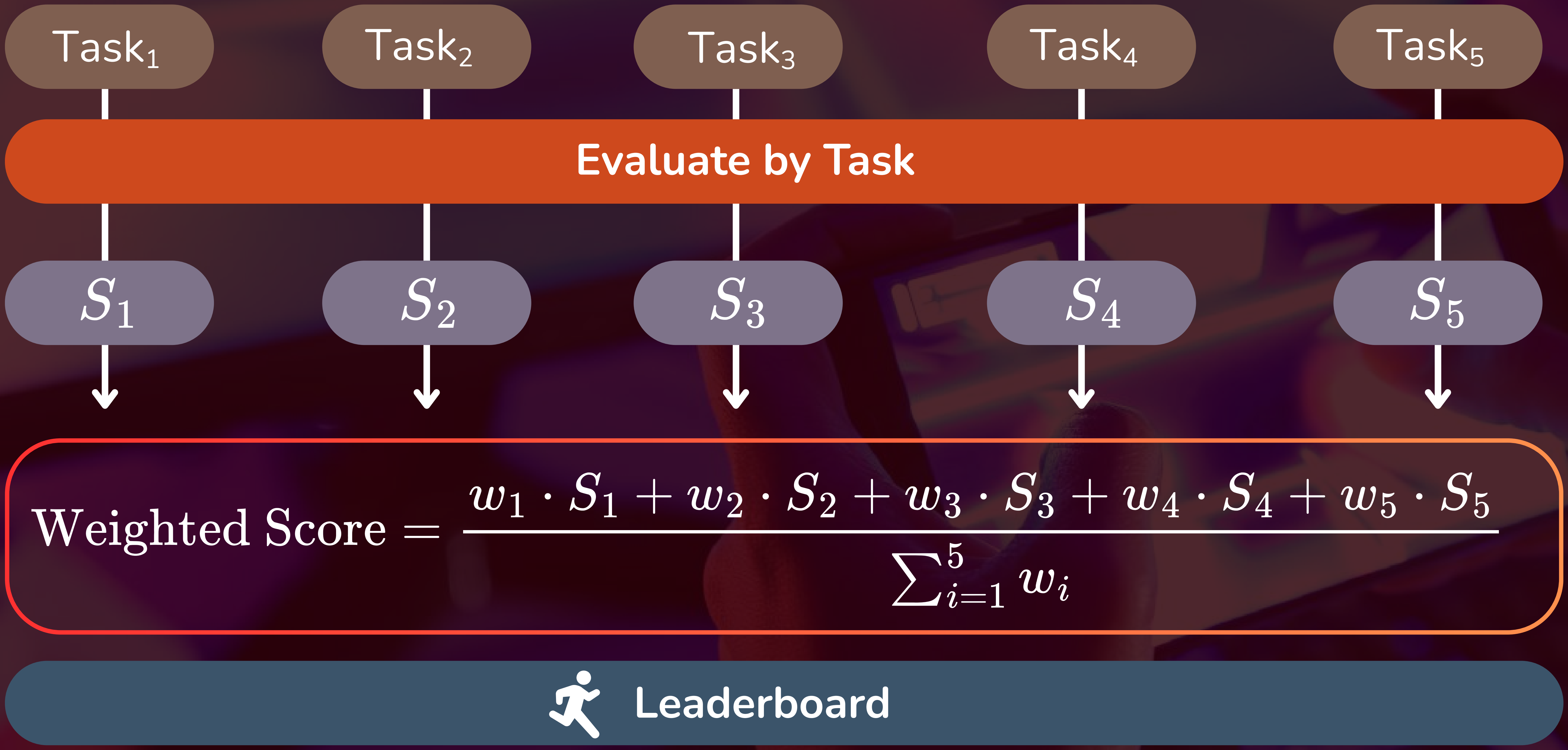
Back to table of content

Evaluation Metrics

Metric	Definition	Applied Task
F_1	$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$	Task2, Task4
F_2	$F_2 = 5 \cdot \frac{\text{Precision} \cdot \text{Recall}}{4 \cdot \text{Precision} + \text{Recall}}$	Task1
F_3	$F_3 = 10 \cdot \frac{\text{Precision} \cdot \text{Recall}}{9 \cdot \text{Precision} + \text{Recall}}$	Task5
NMAE	$\text{Normalized MAE} = \frac{1}{1 + \frac{\sum_{i=1}^{N} \lvert y_i - \hat{y}_i \rvert}{\sum_{i=1}^{N} y_i}}$	Task3

Back to table of content

Kaggle Score Calculation



$$\text{Weighted Score} = \frac{w_1 \cdot S_1 + w_2 \cdot S_2 + w_3 \cdot S_3 + w_4 \cdot S_4 + w_5 \cdot S_5}{\sum_{i=1}^{5} w_i}$$

range: [0,1]

Baseline Model

We builded the first baseline using only the models we covered in class

Weighted Score (Public) = 0.68 (Ground Baseline)

e.g. Logistic Regression | Random Forest | CNN | Isolation Forest

The challenge baseline selects a model and leverages techniques from the hints provided in this instruction slide.

Weighted Score (Public) = 0.75 (Challenge Baseline)

To go beyond the baseline, try experimenting with cutting-edge models and techniques. We encourage you and your team to explore and experiment with new ideas!

let’s get our hands dirty!

“Good luck!” - Aj & TA