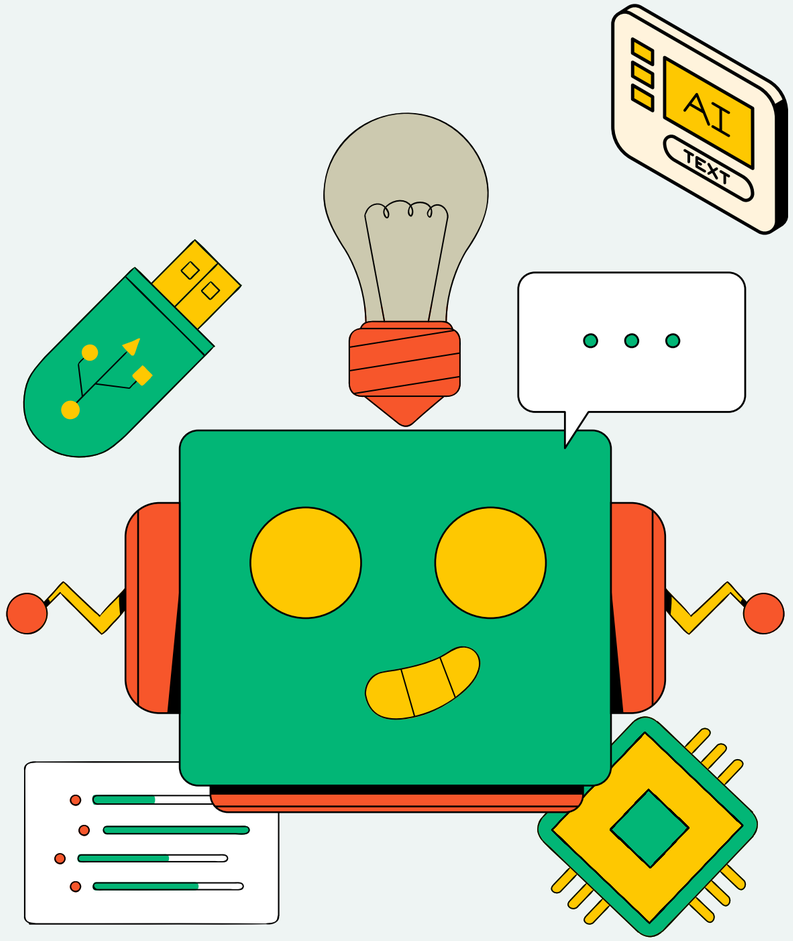
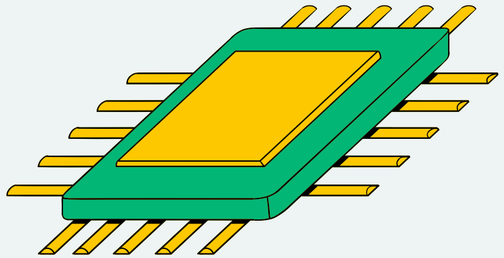


MACHINE LEARNING PROJECT

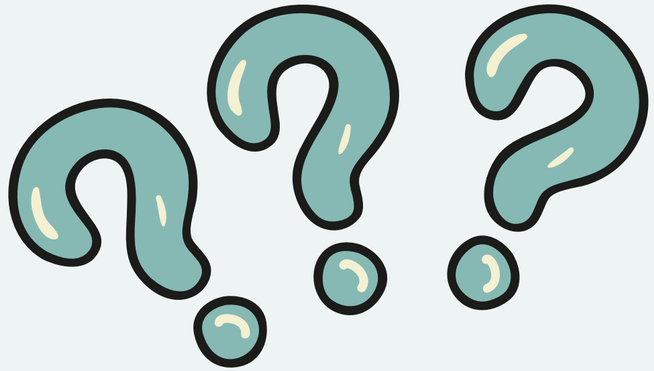
PRESENTATION

PRESENTED BY:

KERNEL LENNEL



ANALYSIS OF THE ANOMALY TYPES



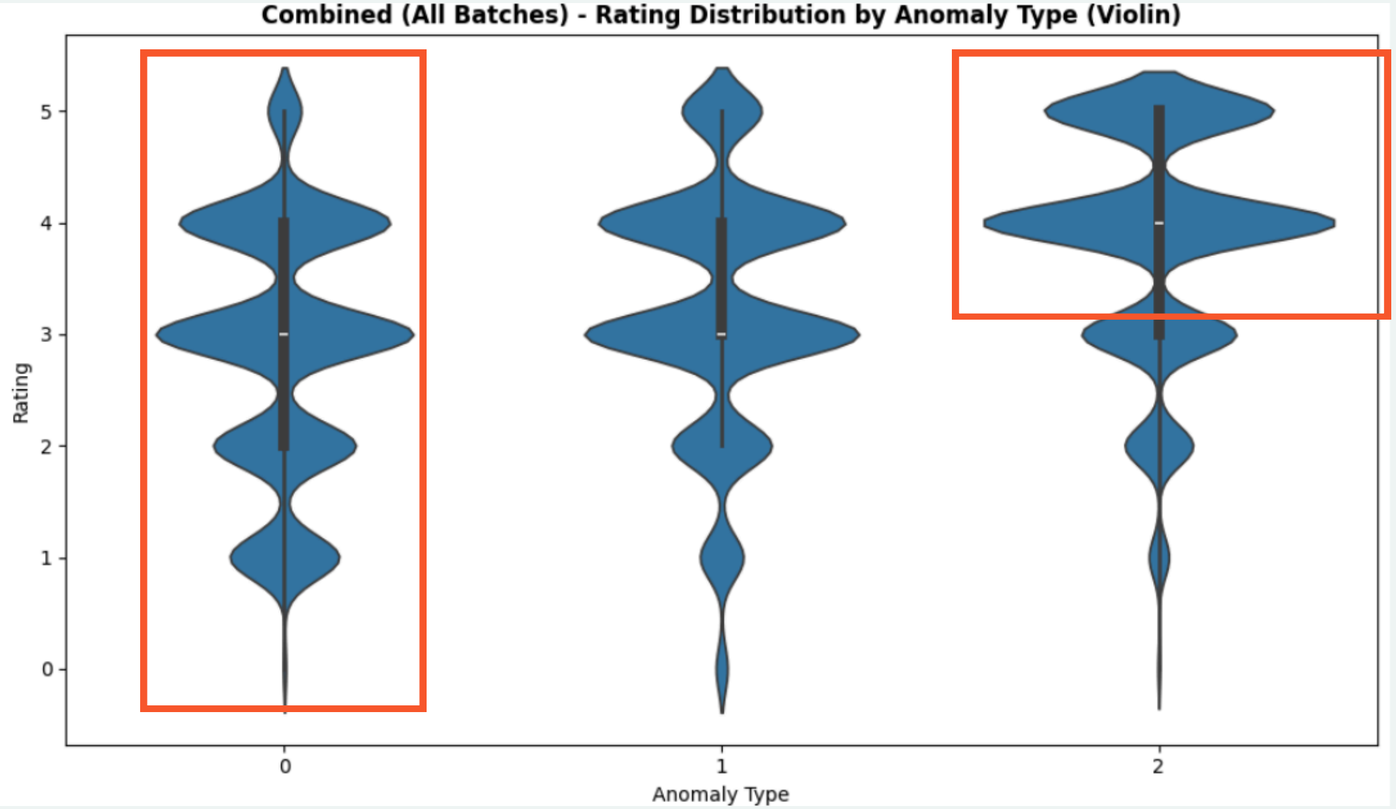
Type 0

Type 1

Type 2

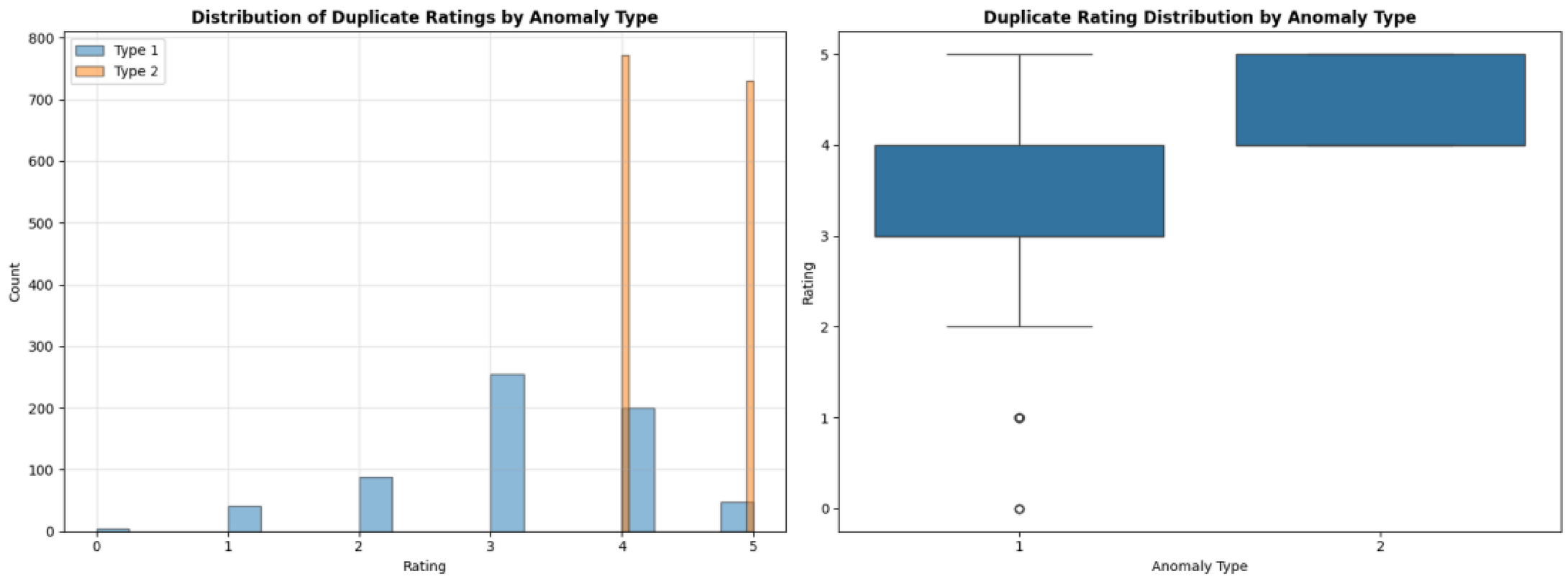
COMBINED STATISTICS

anomtype	n_ratings	mean	std	skew	n_unique_items	dup_rate	label_y
0	308.50	2.8490	1.0874	-0.3157	308.50	0.0000	0.5504
1	273.05	3.3332	0.9606	-0.4490	256.40	0.0533	0.4842
2	303.60	3.9041	0.8788	-0.6983	260.65	0.1460	0.5045



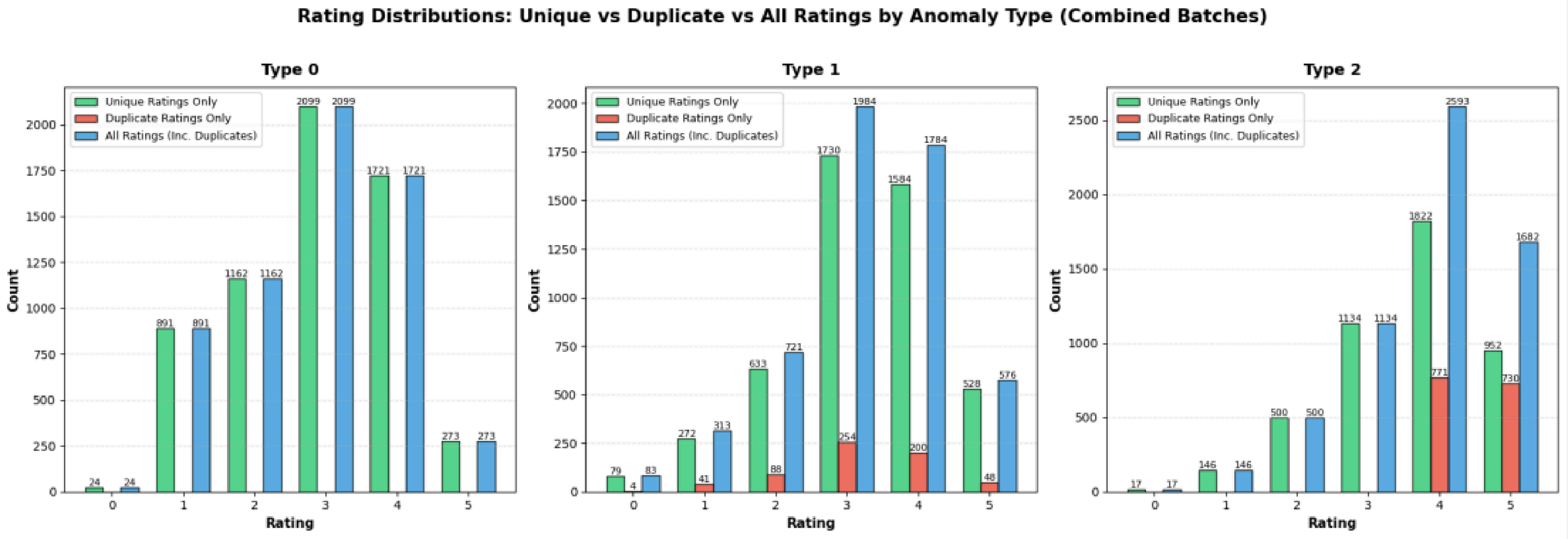
- Type 0 has no duplicates. Type 1 has some, Type 2 has the most
- Type 2 has more 4 and 5 ratings
- Type 2 has higher median and Upper Quartile Range
- Type 0 has more even distribution of ratings

DUPLICATE ANALYSIS



- Type 2 ONLY has 4's and 5's as duplicates, which also inflates its mean to be the higher
- Type 1 has more even distribution of duplicates, but more of 3's and 4's.

DUPLICATE ANALYSIS



- Different Types have different distributions
- Type 1's duplicates seem to follow its natural distribution

USER BASED FEATURES

USER MEAN, STD DEV, MIN, MAX, SKEW	MEAN, STD DEV, MIN, MAX, SKEW OF USER RATINGS
USER COUNT	NUMBER OF RATINGS BY THE USER
RATING MODE	MOST FREQUENT RATING GIVEN BY THE USER
RATING ENTROPY	SHANNON ENTROPY OF THE USER'S RATING DISTRIBUTION HIGH ENTROPY = MORE DIVERSE, LOW ENTROPY = MORE BIASED TOWARDS A FEW RATINGS $-\sum_{i=0}^{5} p_i log_2(p_i)$ WHERE P_i IS THE PROBABILITY OF THE USER RATING = i
ENTROPY COUNT	INTERACTION TERM BETWEEN RATING ENTROPY AND USER COUNT CAPTURES DIVERSITY WEIGHTED BY ACTIVITY
USER Z-SCORE (MEAN & STD DEV)	MEAN AND STD DEV OF USER Z-SCORE $\frac{user\ rating - \ user\ rating\ mean}{user\ rating\ std\ dev}$
USER BELOW/ABOVE 3	PROPORTION OF RATINGS THE USER GAVE THAT WERE BELOW/ABOVE 3
EXTREME RATING RATIO	PROPORTION OF RATINGS THE USER GAVE THAT WERE EITHER 1 OR 5

GLOBAL/ITEM BASED FEATURES

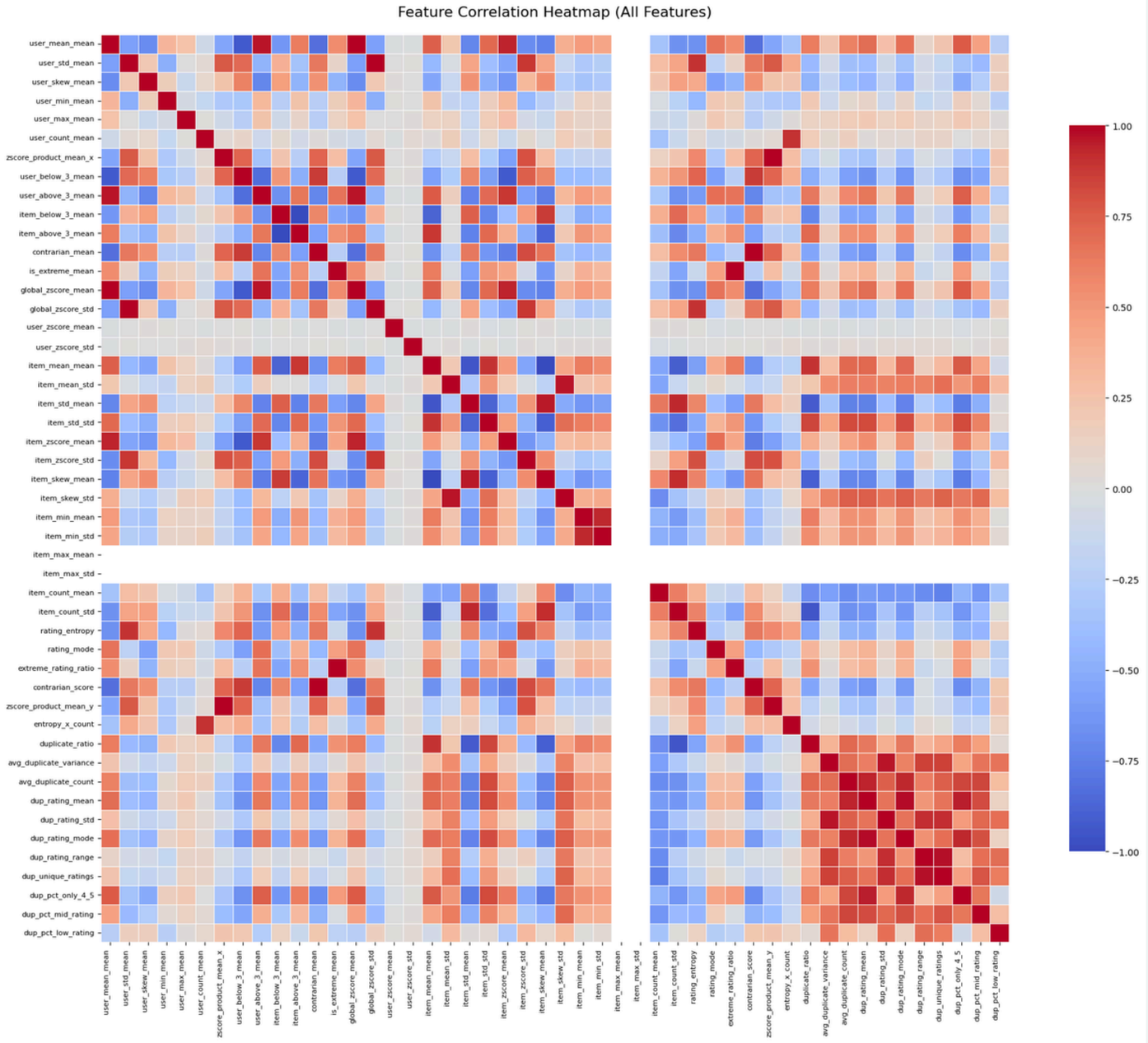
ITEM MEAN, STD DEV, MIN, MAX, SKEW, COUNT (MEAN & STD DEV)	MEAN AND STD DEV OF VARIOUS ITEM STATISTICS FOR EACH USER
ITEM BELOW/ABOVE 3	PROPORTION OF ITEMS RATED BY THE USER THAT HAVE MEAN RATING BELOW/ABOVE 3
ITEM Z-SCORE (MEAN & STD DEV)	MEAN AND STD DEV OF ITEM Z-SCORE OF ALL ITEMS RATED BY USER $\frac{user\ rating - item\ rating\ mean}{item\ rating\ std\ dev}$
CONTRARIAN MEAN	PROPORTION OF RATINGS WHERE THE USER'S RATING WAS OPPOSITE THE ITEM MEAN (USER BELOW 3 & ITEM ABOVE 3) OR (USER ABOVE 3 & ITEM BELOW 3)
GLOBAL Z-SCORE (MEAN & STD DEV)	MEAN AND STD DEV OF THE Z-SCORE OF THE USER'S RATINGS RELATIVE TO THE GLOBAL RATING MEAN $\frac{user\ rating - all\ ratings\ mean}{all\ ratings\ std\ dev}$
Z-SCORE PRODUCT MEAN	MEAN OF THE PRODUCT BETWEEN GLOBAL Z-SCORE AND ITEM Z-SCORE FOR ALL RATINGS FOR THE USER POSITIVE VALUE IMPLIES THAT THE GLOBAL Z-SCORE IS CONSISTENT WITH THE ITEM Z-SCORE NEGATIVE VALUE IMPLIES THAT THEY ARE INCONSISTENT

GLOBAL/ITEM STATISTICS ARE CALCULATED FROM TRAINING SET ONLY

USER DUPLICATE RATING BASED FEATURES

DUPLICATE RATIO	RATIO OF DUPLICATE RATINGS TO TOTAL RATINGS
AVERAGE DUPLICATE VARIANCE	AVERAGE STD DEV OF DUPLICATED RATINGS PER ITEM RATED BY THE USER VALUE CLOSE TO 0 IMPLIES THAT DUPLICATED RATINGS ARE ALMOST IDENTICAL
AVERAGE DUPLICATE COUNT	AVERAGE NUMBER OF DUPLICATED RATINGS PER ITEM RATED BY THE USER
DUPLICATE RATING MEAN, STD DEV, MODE, RANGE	VARIOUS STATISTICS OF DUPLICATED RATINGS
DUPLICATE UNIQUE RATINGS	NUMBER OF UNIQUE RATING VALUES AMONGST DUPLICATED RATINGS
DUPLICATE PCT ONLY 4 OR 5	PERCENTAGE OF DUPLICATED RATINGS THAT ARE 4 OR 5
DUPLICATE PCT MID RATING	PERCENTAGE OF DUPLICATED RATINGS THAT ARE 3 OR 4
DUPLICATE PCT LOW RATING	PERCENTAGE OF DUPLICATED RATINGS THAT ARE 1 OR 2

CORRELATION MATRIX



PART 1 REGRESSION

PART 1 METHODOLOGY

SEEDING

Seed = 42

TRAIN TEST SPLIT

Train 80%
Test 20%

DATA TRANSFORMATION

Feature Engineering

CROSS VALIDATION

5 fold
Train 80%
Validation 20%

SCALING WITHIN FOLD

Scale on Train
Transform test

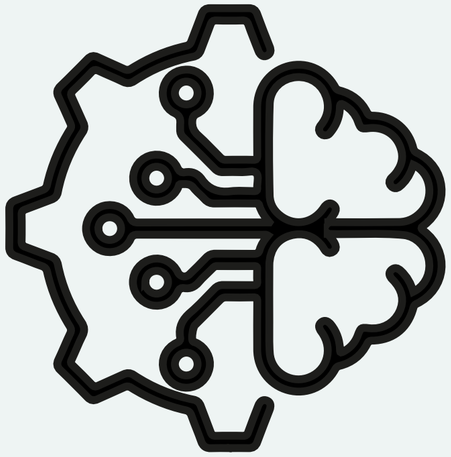
HYPERPARAMETER TUNING

Greedy Grid Search

PART 1 MODELS

MODEL NAME	PARAMS	MEAN/STD DEV
LINEAR REGRESSION	NIL	0.0858/0.0025
RIDGE REGRESSION	POLY DEGREE, ALPHA	0.0568/0.0016
LASSO REGRESSION	POLY DEGREE, ALPHA	0.0596/0.0021
ELASTIC NET REGRESSION	POLY DEGREE, ALPHA, L1/L2 RATIO	0.0567/0.0022
RANDOM FOREST	NO. OF LEARNERS, MAX DEPTH, MAX FEATURES	0.0596/0.0014
ADAPTIVE BOOSTING	NO. OF LEARNERS, LEARNING RATE, LOSS TYPE	0.0729/0.0025
EXTREME GRADIENT BOOSTING	NO. OF LEARNERS, LEARNING RATE, MAX DEPTH	0.0591/0.0012
LIGHT GRADIENT BOOSTING	NO. OF LEARNERS, LEARNING RATE, MAX DEPTH	0.0598/0.0012
DENSE NEURAL NETWORK	HIDDEN LAYER UNITS, BATCH SIZE, LEARNING RATE, L2 REGULARIZATION	0.0510/0.0016

NEURAL NETWORK HYPERPARAMETERS



Batch Size: 64

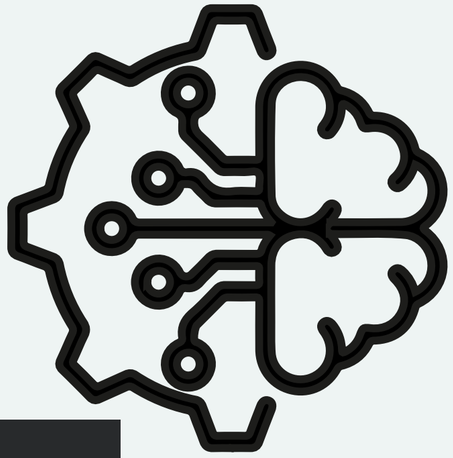
Hidden Layer 1: 4

Hidden Layer 2: 64

L2 Regularisation: 0.001

Learning Rate: 0.001

NEURAL NETWORK ARCHITECTURE



```
model = models.Sequential([
    layers.Input(shape=(48,)),
    layers.Dense(units=units1, activation='relu',
                 kernel_regularizer=regularizers.l2(l2_reg)),
    layers.Dense(units=units2, activation='relu',
                 kernel_regularizer=regularizers.l2(l2_reg)),
    layers.Dense(1, activation='sigmoid')
])
```

FEATURE OPTIMISATION

DUPLICATE

- zscore_product_mean
- contrarian_mean
- is_extreme_mean

REDUNDANT

- item_max_mean
- item_max_std

Removing these features gave a **worse** MAE

Mean CV MAE

Before: 0.0510
After : 0.0516

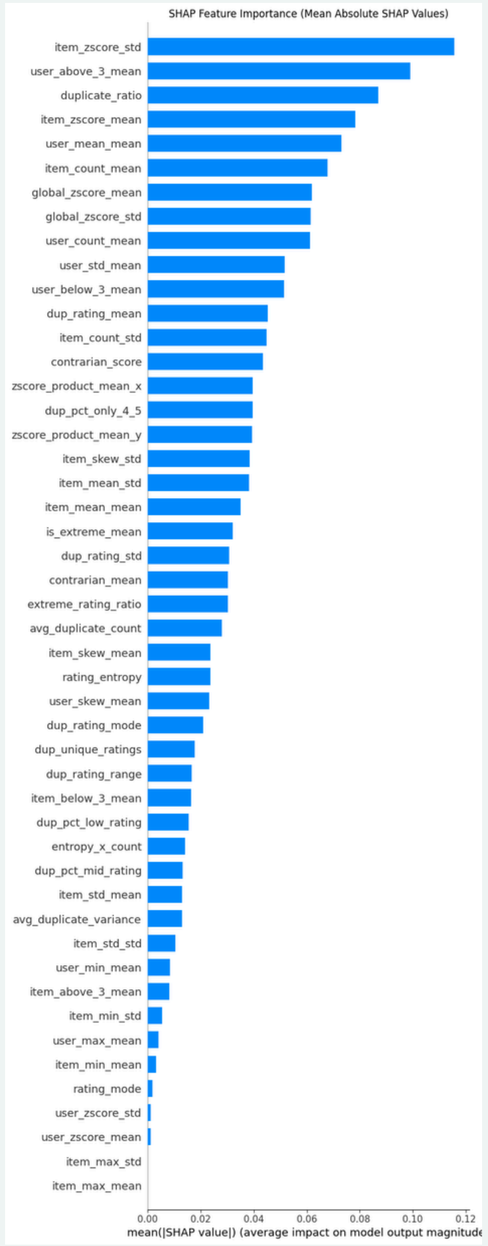
FEATURE IMPORTANCE

CV LEAVE 1 OUT

- do the 5 fold validation again, taking out 1 feature at a time
- higher drop in inaccuracy → more important feature

SHAP

- predicted value = sum of all SHAP values of the features + base value
- higher absolute value = good



Mean CV MAE

Before: 0.0510
After : 0.0542

FIT

OVERFITTING:

TRAIN = 0.0473

MEAN CV = 0.0510

TEST = 0.0507

UNDERFITTING:

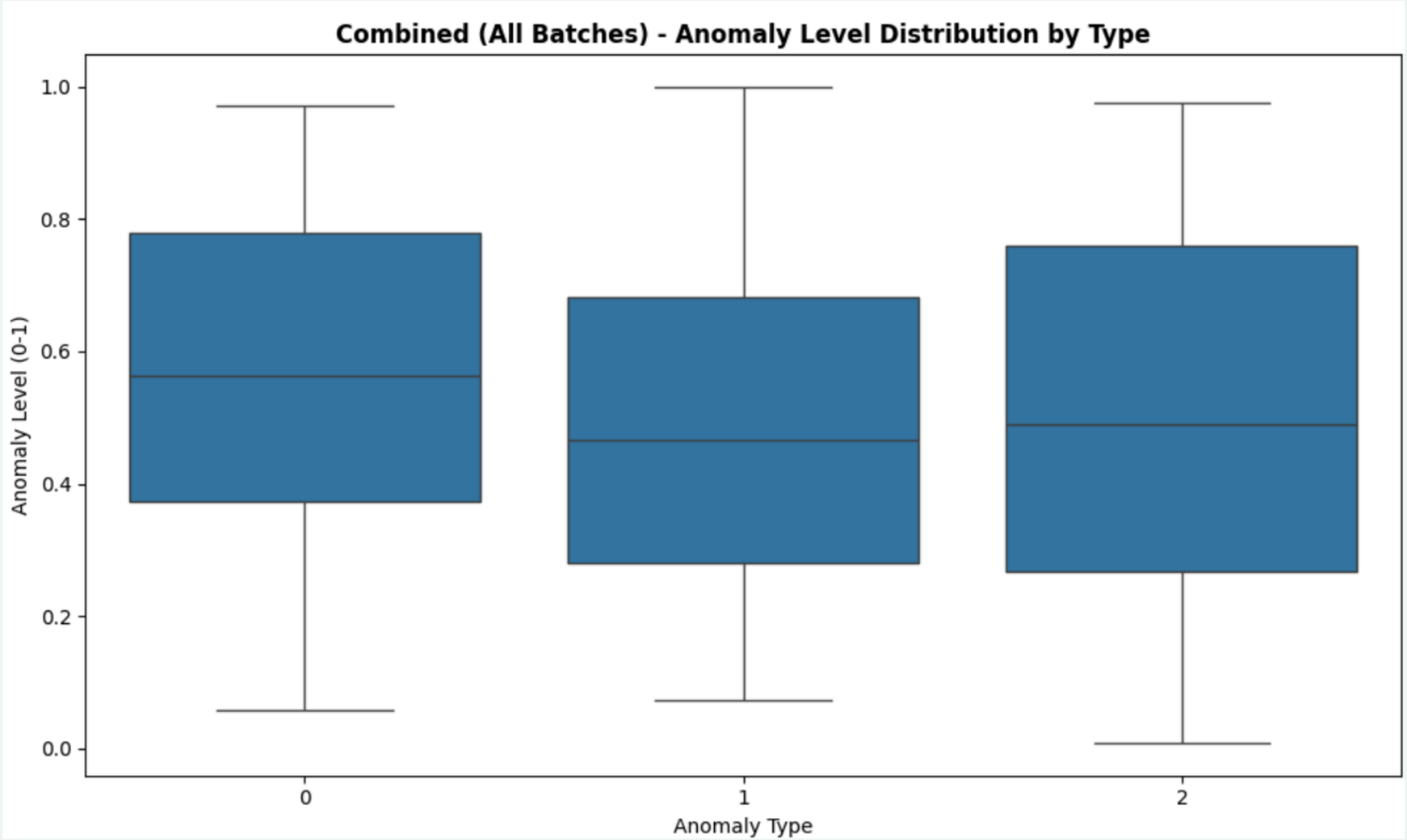
BEST: 0.0511

OURS: 0.0539

AVERAGE: 0.0907

PART 2 CLASSIFICATION

ANOMALY LEVEL VS TYPE



PART 2 METHODOLOGY

SEEDING

Seed = 42

TRAIN TEST SPLIT

Train 80%
Test 20%

DATA TRANSFORMATION

Feature Engineering

CROSS VALIDATION ON LABELLED DATA

5 fold
Train 80%
Validation 20%

SCALING WITHIN FOLD

Scale on Labelled Train
Transform Labelled Val
Transform Unlabelled

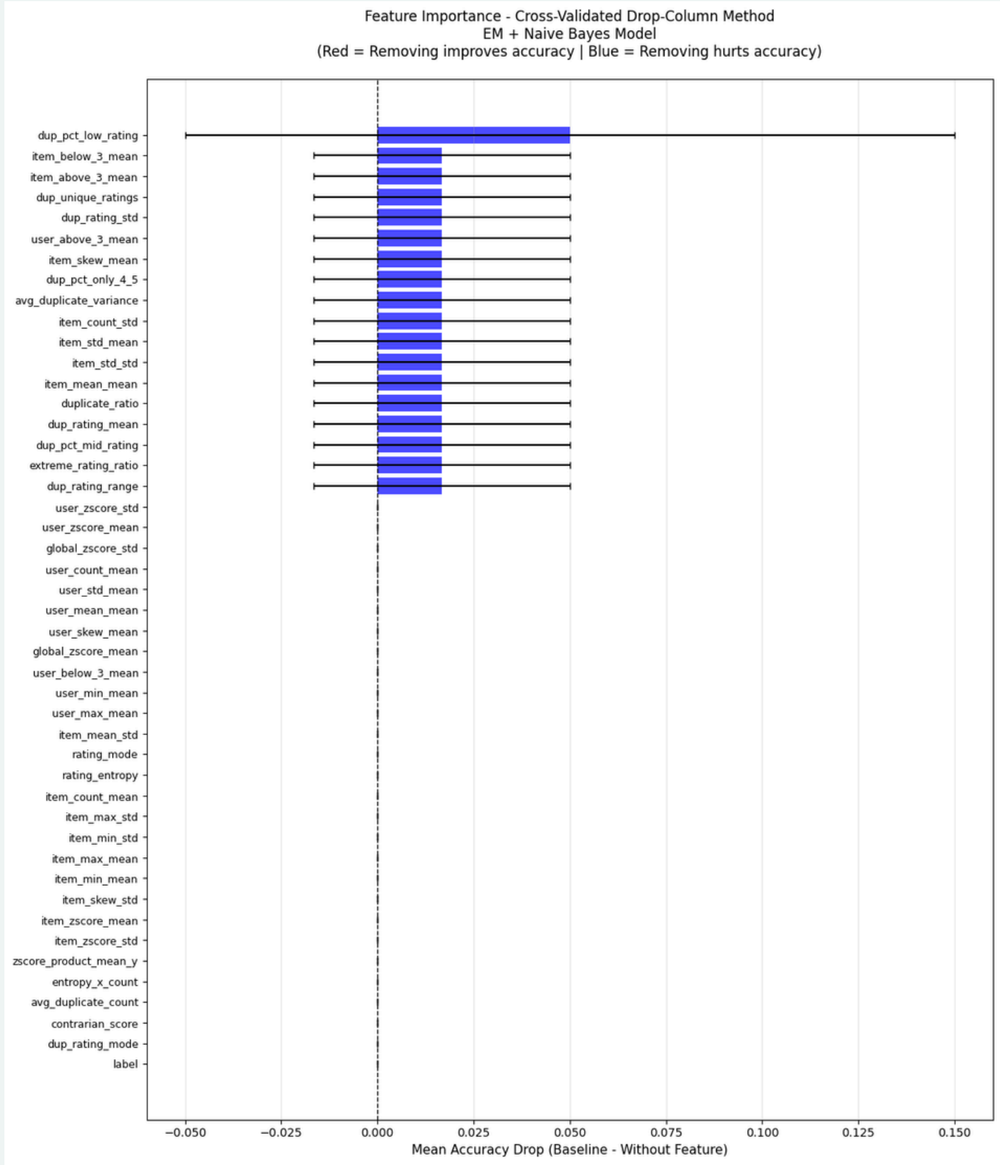
HYPERPARAMETER TUNING

Greedy Grid Search

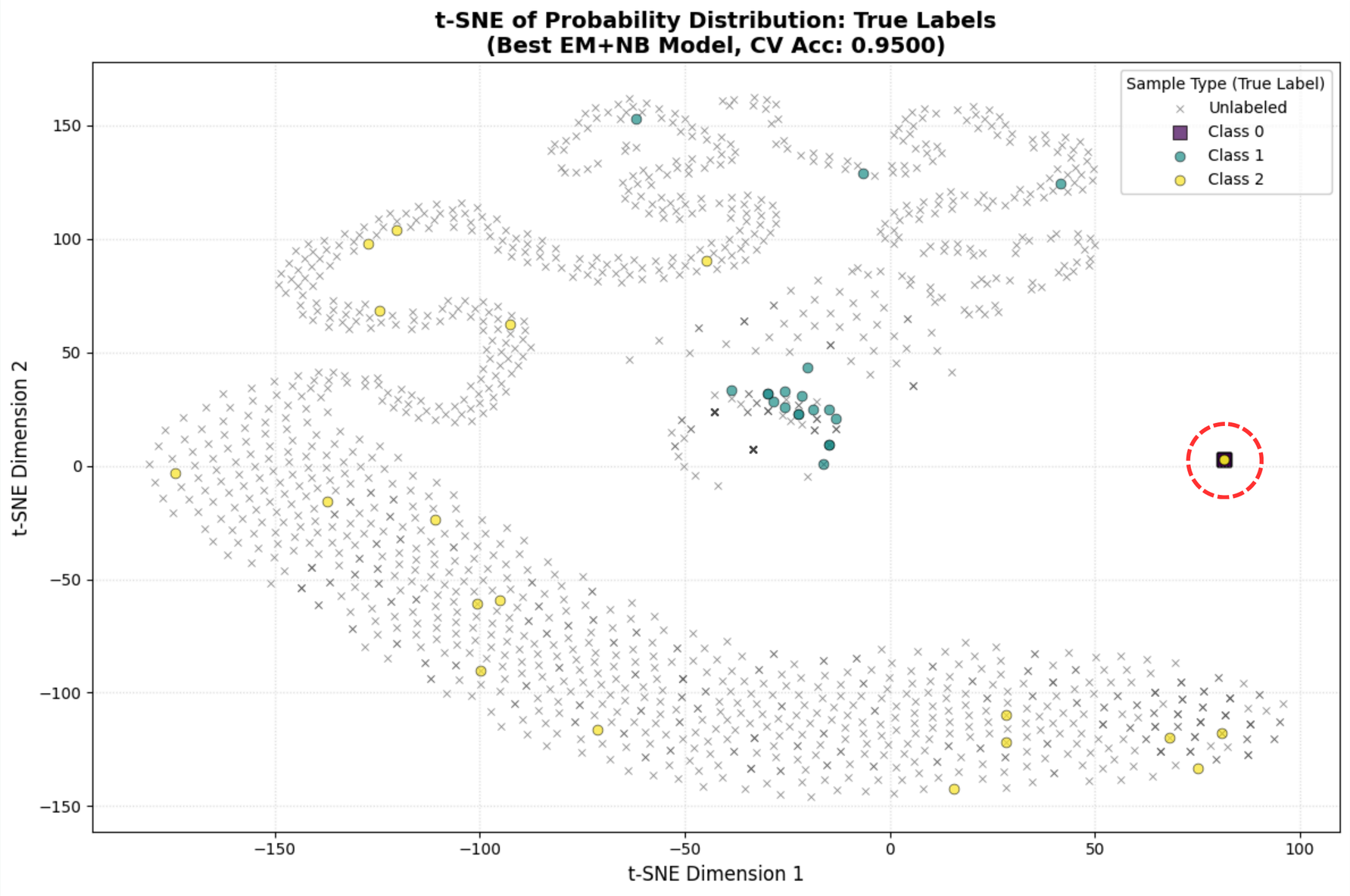
PART 2 MODELS

MODEL NAME	PARAMS	MEAN/STD DEV
LABEL SPREADING	KERNEL, NO. OF NEIGHBOURS, ALPHA, GAMMA	0.9167/0.0527
LABEL SPREADING (SHAP)	KERNEL, NO. OF NEIGHBOURS, ALPHA, GAMMA	0.9333/0.0624
LABEL SPREADING (CV LEAVE-ONE-OUT)	KERNEL, NO. OF NEIGHBOURS, ALPHA, GAMMA	0.9167/0.0527
SELF TRAINING RF	NO. OF LEARNERS, MAX DEPTH, MAX FEATURES, MAX ITER, CONFIDENCE THRESHOLD	0.9333/0.0333
SELF TRAINING RF (INBUILT FEATURE IMPORTANCE)	NO. OF LEARNERS, MAX DEPTH, MAX FEATURES, MAX ITER, CONFIDENCE THRESHOLD	0.9167/0.0527
SELF TRAINING HGB	NO. OF LEARNERS, MAX DEPTH, LEARNING RATE, SUBSAMPLE, MAX ITER, CONFIDENCE THRESHOLD	0.9167/0.0527
SELF TRAINING HGB (INBUILT FEATURE IMPORTANCE)	NO. OF LEARNERS, MAX DEPTH, LEARNING RATE, SUBSAMPLE, MAX ITER, CONFIDENCE THRESHOLD	0.9167/0.0527
EM NAIVE BAYES	VAR SMOOTHING, MAX ITER, USE WEIGHTS, CONVERGENCE TOLERANCE, CONFIDENCE THRESHOLD	0.9167/0.0527
EM NAIVE BAYES (SHAP)	VAR SMOOTHING, MAX ITER, USE WEIGHTS, CONVERGENCE TOLERANCE, CONFIDENCE THRESHOLD	0.9167/0.0527
EM NAIVE BAYES (CV LEAVE-ONE-OUT)	VAR SMOOTHING, MAX ITER, USE WEIGHTS, CONVERGENCE TOLERANCE, CONFIDENCE THRESHOLD	0.9500/0.0408

EM NAIVE BAYES CV LEAVE-ONE-OUT



EM NAIVE BAYES T-SNE



NAIVE BAYES

$$\hat{c} = \arg\max_{c} P(c|\mathbf{x})$$

$$P(c|\mathbf{x}) = \frac{P(\mathbf{x}|c) \cdot P(c)}{P(\mathbf{x})}$$

$P(\mathbf{x}|c)$: The **likelihood** of observing the features $\mathbf{x}$ given the class c.

$P(c)$: The **prior probability** of class c.

$P(\mathbf{x})$: The **evidence** (probability of the features $\mathbf{x}$ occurring).

$$P(\mathbf{x}|c) = P(x_1, x_2, \ldots, x_n|c) \approx \prod_{i=1}^{n} P(x_i|c)$$

Assuming all features are conditionally independent given class c

$$P(\mathbf{x}) = \sum_{c} P(c)P(\mathbf{x}|c) = \sum_{c} P(c) \prod_{i=1}^{n} P(x_i|c)$$

$$\hat{c} = \arg\max_{c} \left(\frac{P(c) \prod_{i=1}^{n} P(x_i|c)}{P(\mathbf{x})} \right)$$

EM NAIVE BAYES

Step 1:

Calculate $P(c)$, $P(x_i|c)$, $P(x)$ using labelled data

Step 2 (E step):

Calculate soft labels for unlabelled data

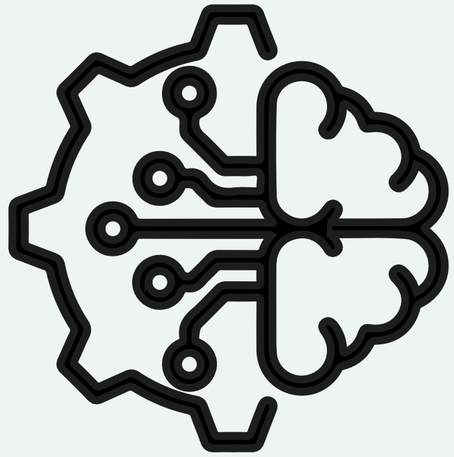
Step 3 (M step):

Update $P(c)$, $P(x_i|c)$, $P(x)$ using labelled and confident unlabelled data

Step 4:

Repeat step 2–3 until it converges

EM NAIVE BAYES HYPERPARAMETERS



Variance Smoothing: 1.0e-07

Max Iterations: 50

Convergence Tolerance: 0.1

Confidence Threshold: 0.99

Use Weights: False

FIT

OVERFITTING:

TRAIN = 0.9333

MEAN CV = 0.9500

NO TEST

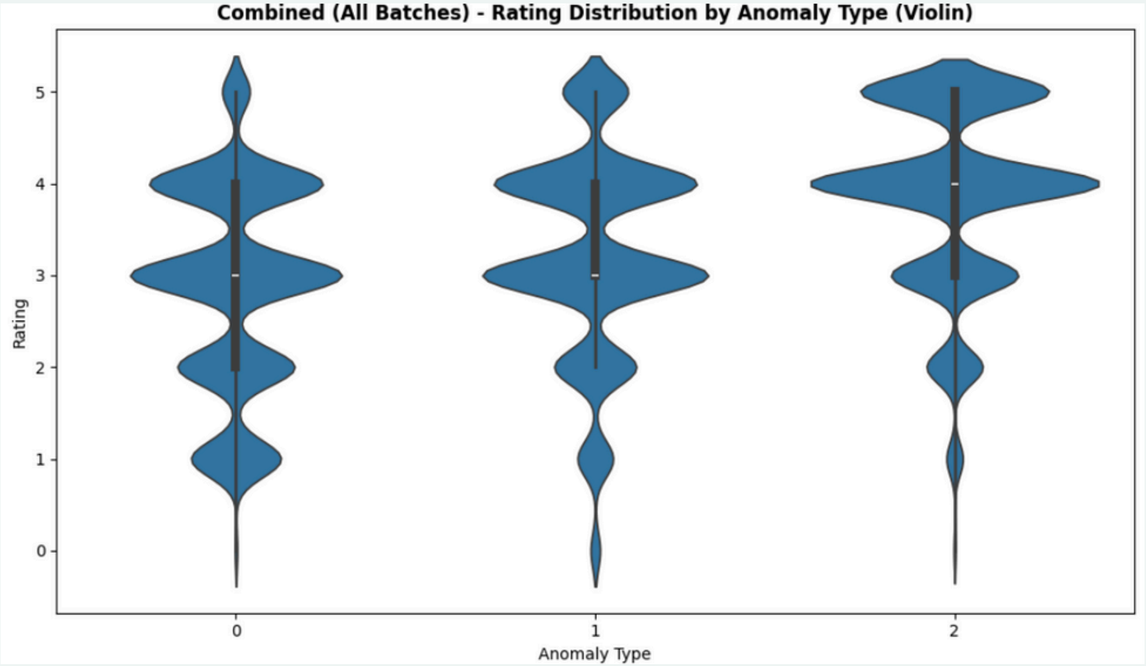
UNDERFITTING:

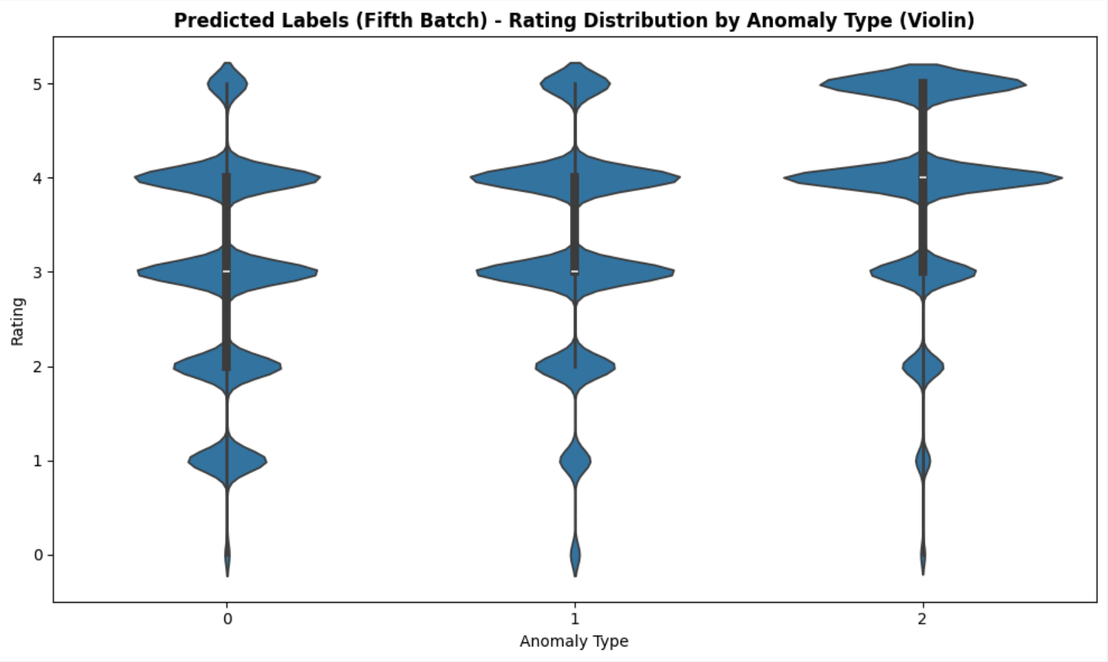
BEST: 0.9189

OURS: 0.9189

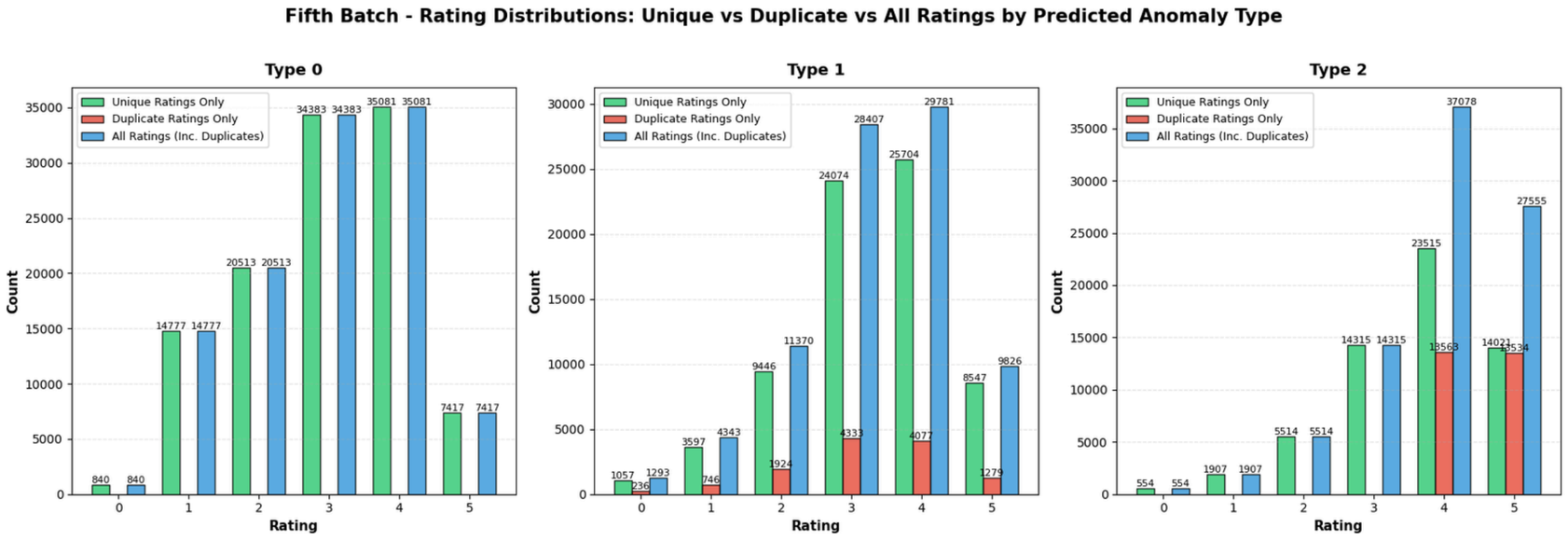
AVERAGE: 0.7161

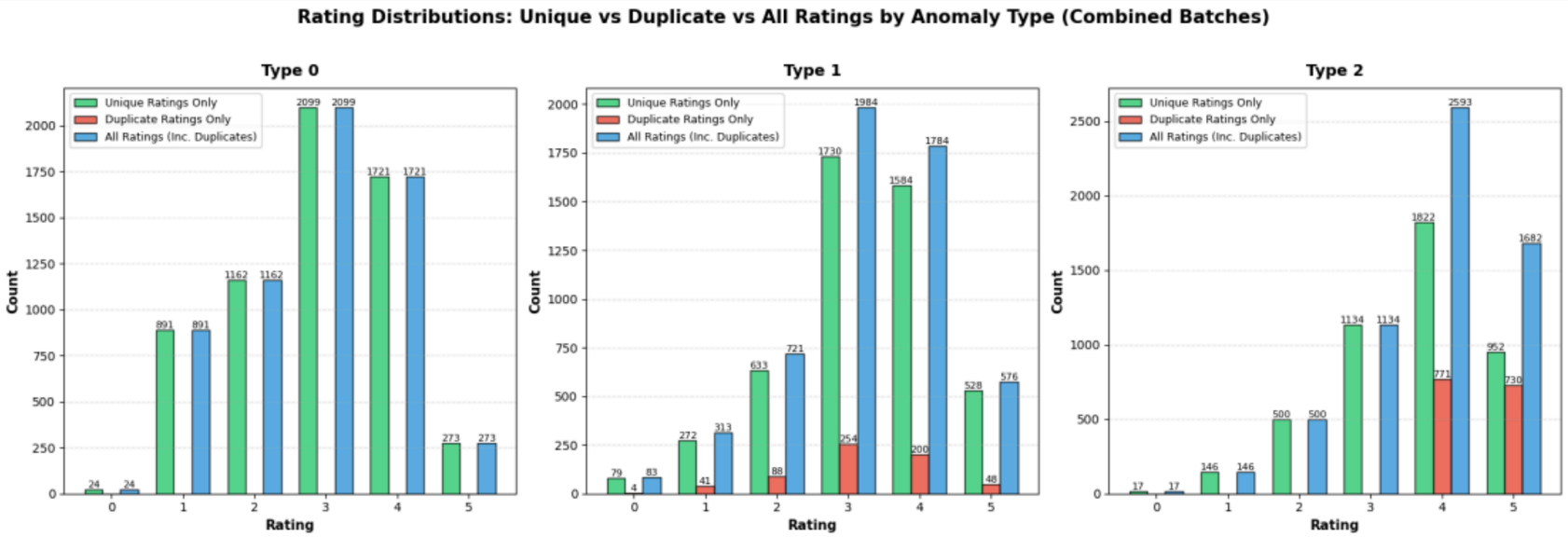
CONFIRMING FINDINGS (5TH BATCH PREDICTION)





CONFIRMING FINDINGS (5TH BATCH PREDICTION)





TYPE 0

Real Movie Watchers

- Ratings are relatively evenly distributed, with more 3’s and 4’s.
- Few or no duplicate ratings.
- Suggests natural, non-anomalous rating behavior.

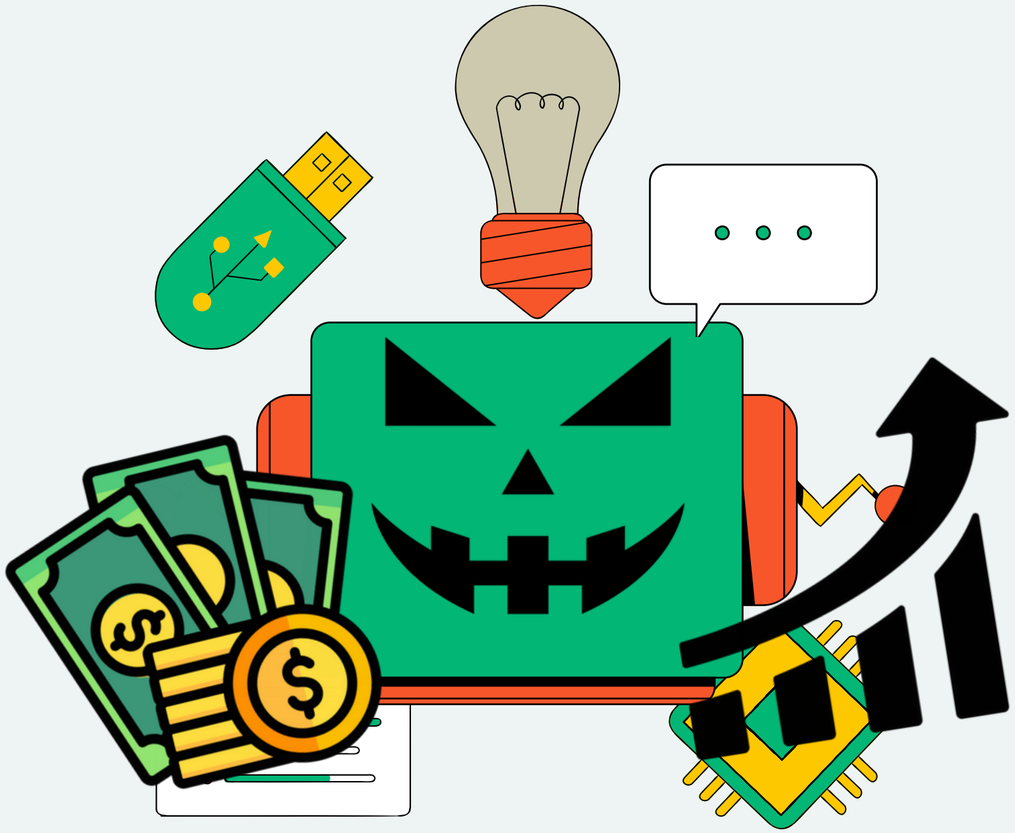
TYPE 1

Slightly Suspicious Users

- Duplicate ratings appear across the full 0–5 range, most concentrated around 3–4
- Duplicates mirrors the overall rating distribution
- Duplication pattern indicates users repeatedly rating same movie
- Possible signs of engagement manipulation or data quality issues

TYPE 2

HIGHLY Suspicious Users



- Strongly skewed distribution, heavily concentrated on 4’s and 5’s
- Extreme positive bias
- Very high volume of duplicate ratings, almost exclusively 4’s and 5’s
- Possible signs of botting or review-boosting

FUTHER EXPLORATION

- more extensive grid search
- other models