ANOMALY DETECTION

NGO NGUYEN THANH THANG - 20229549

DATA SET DESCRIPTION

unit number	cycle number	operational setting 1	operational setting 2	operational setting 3	sensor measurement 1	sensor measurement 2
1	1	34.9983	0.8400	100.0	449.44	555.32
1	2	41.9982	0.8408	100.0	445.00	549.90
1	3	24.9988	0.6218	60.0	462.54	537.31
1	4	42.0077	0.8416	100.0	445.00	549.51
1	5	25.0005	0.6203	60.0	462.54	537.07
...	...	...	...	...	...	...
260	312	20.0037	0.7000	100.0	491.19	608.79
260	313	10.0022	0.2510	100.0	489.05	605.81
260	314	25.0041	0.6200	60.0	462.54	537.48
260	315	25.0033	0.6220	60.0	462.54	537.84

- The data consists of multiple multivariate time series, each representing a different engine.
- All engines are of the **same type**, each with **unknown initial wear** and **manufacturing variations** considered **normal**
- Engines operate under three **operational settings** affecting performance
- Data is contaminated with **sensor noise**
- Data set has 26 features

Each time series begins with normal operation and develops a fault that leads to system failure in the training set.

The data is already divided into **training** and **test** subsets.

In the test set, the series ends before system failure.

	cycle number
unit number	
1	149
2	269
3	206
4	235
5	154
...	...
256	163
257	309

OBJECTIVES

DETECT ANOMALY

PREDICT THE **REMAINING USEFUL LIFECYCLE (RUL)** FOR ENGINES IN THE TEST SET

BASIC EDA

- Lacking information regarding the variables (features)
- Cannot extract any impactful insight

	operational setting 3	sensor measurement 18	sensor measurement 19
count	20631.0	20631.0	20631.0
mean	100.0	2388.0	100.0
std	0.0	0.0	0.0
min	100.0	2388.0	100.0
25%	100.0	2388.0	100.0
50%	100.0	2388.0	100.0
75%	100.0	2388.0	100.0
max	100.0	2388.0	100.0

MODEL BUILDING

- Objective 1: Detect any **anomaly** or **unusual occurrence** in the time series
- Model Name: Hugging Face's Zero-shot classification

HOW TO USE HUGGING FACE'S ZERO-SHOT CLASSIFICATION MODEL

HOW TO USE ZERO SHOT FOR ANOMALY DETECTION

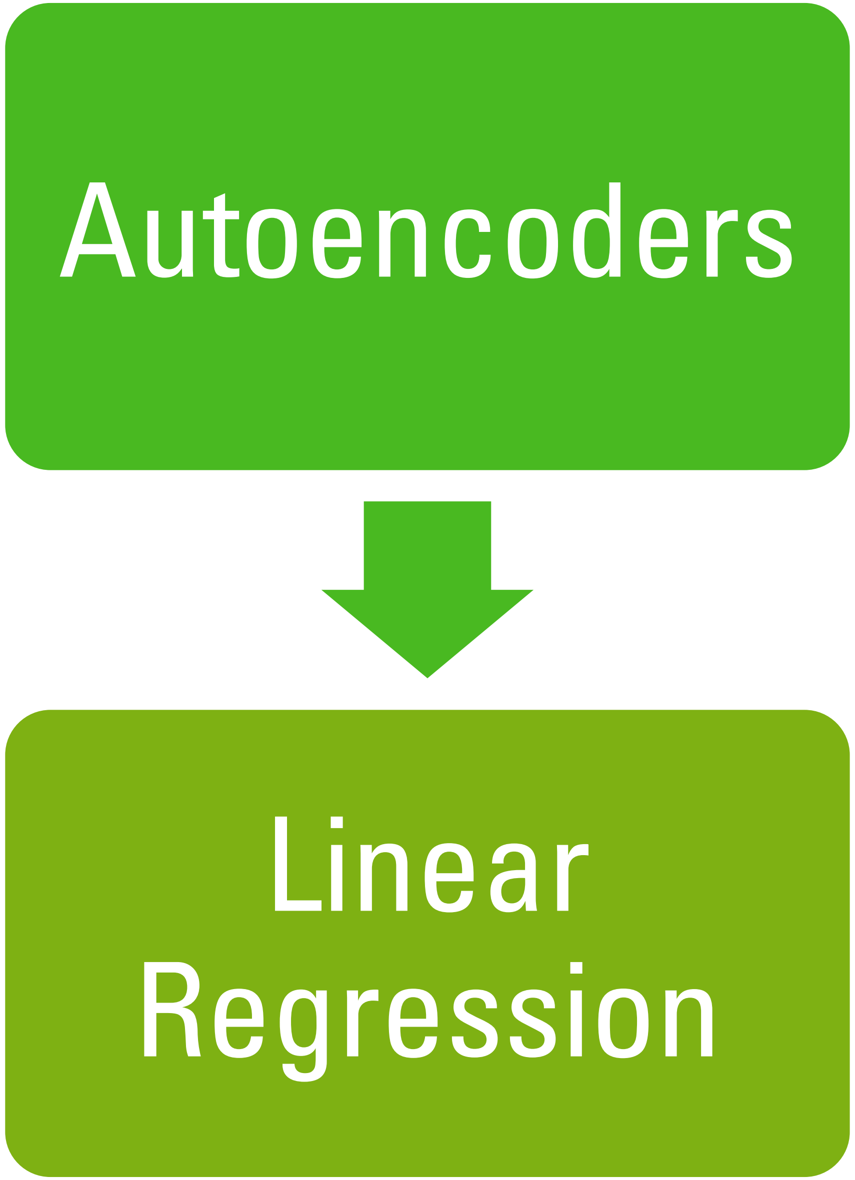
	unit number	cycle number	operational setting 1	operational setting 2	operational setting 3	sensor measurement 1	sensor measurement 2	sensor measurement 3
0	1	1	-0.0007	-0.0004	100.0	518.67	641.82	1589.70
1	1	2	0.0019	-0.0003	100.0	518.67	642.15	1591.82

Text format: unit number:1.0 at cycle number:1.0, operational setting 1:-0.0007, …. sensor measurement 26: 0.02

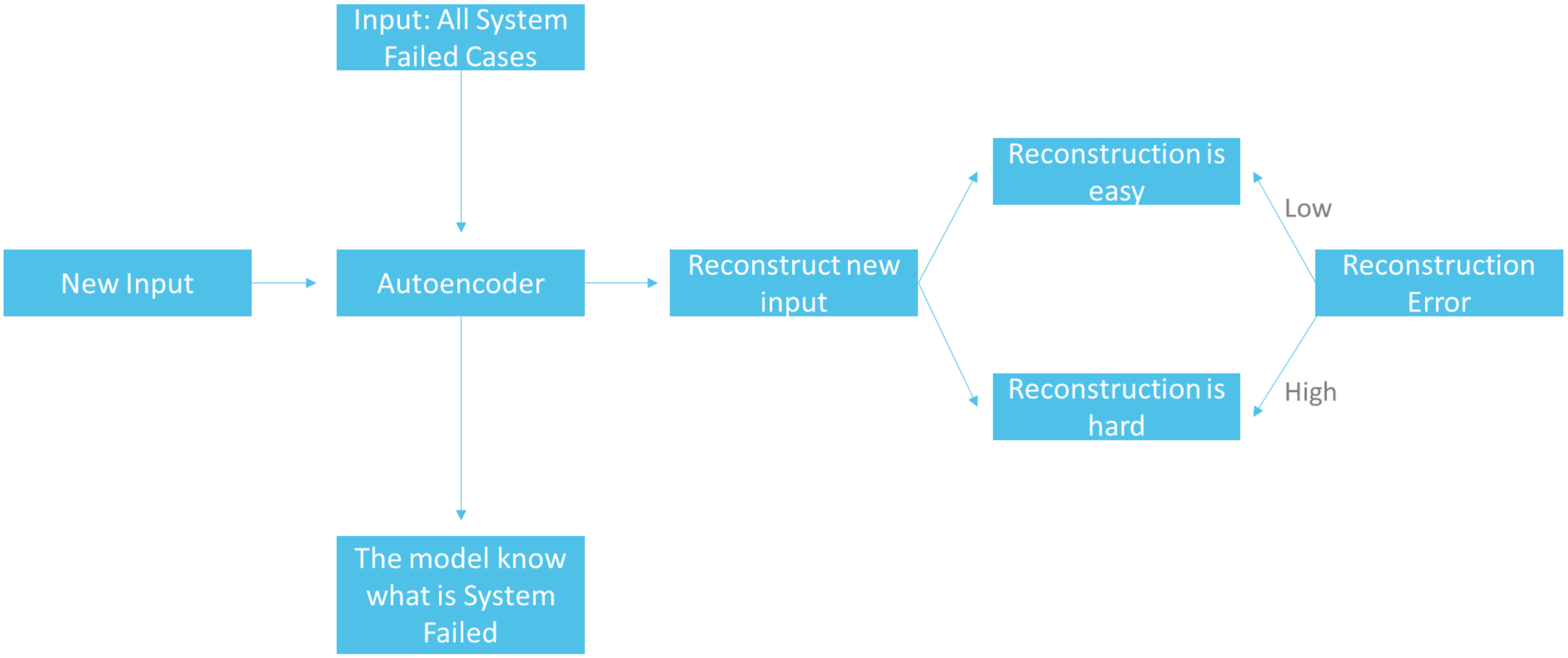
```python
from transformers import pipeline
# Initialize the zero-shot classification pipeline
classifier = pipeline("zero-shot-classification", model="facebook/bart-large-mnli")

# Define the possible candidate labels (categories)
candidate_labels = ["Normal", "Early Warning", "Moderate Anomaly", "Critical Anomaly"]

# Perform zero-shot classification on the engine data
result = classifier(engine_data, candidate_labels)

# Print the result
print(f"Input data: {engine_data}")
print(f"Predicted Label: {result['labels'][0]}")
print(f"Confidence Score: {result['scores'][0]:.4f}")
```

```
Input data: unit number:1.0 at cycle number:1.0, operational setting 1:-0.0007, operation
Predicted Label: Moderate Anomaly
Confidence Score: 0.4406
```

OBJECTIVE 2: PREDICT THE REMAINING USEFUL LIFECYCLE (RUL) FOR ENGINES IN THE TEST SET



AUTOENCODER



SCALE RECONSTRUCTION ERROR TO PROBABILITY FORMAT

Prob_score: the probability that the system would fail and shut down

prob_score
0.434556
0.312216
0.480827
0.417941
0.364827

CALCULATE THE REMAINING USEFUL LIFECYCLE (RUL) FOR THE TRAIN SET

RUL = The Cycle in which the system failed – the current cycle

	cycle number
unit number	
1	192

cycle number	RUL
1	191
2	190
3	189
4	188
5	187

BUILD MULTIVARIATE LINEAR REGRESSION TO PREDICT RUL (REMAINING USEFUL LIFECYCLE)

Regress all variables (features) against RUL

```
==============================================
 R-squared:                        0.656
 Adj. R-squared:                   0.656
 F-statistic:                      2060.
 Prob (F-statistic):                0.00
 Log-Likelihood:             -1.0499e+05
 AIC:                          2.100e+05
 BIC:                          2.102e+05


-------------------------------------------------
```

Reduced 26 features to 10 features using PCA then perform regression

```
==============================================
 R-squared:                        0.652
 Adj. R-squared:                   0.651
 F-statistic:                      3838.
 Prob (F-statistic):                0.00
 Log-Likelihood:             -1.0513e+05
 AIC:                          2.103e+05
 BIC:                          2.104e+05


==============================================
```

BUILD MULTIVARIATE LINEAR REGRESSION TO PREDICT RUL (REMAINING USEFUL LIFECYCLE)

Choose Regress all variables (features) against RUL

```
==============================================
R-squared:                            0.656
Adj. R-squared:                       0.656
F-statistic:                          2060.
Prob (F-statistic):                    0.00
Log-Likelihood:                 -1.0499e+05
AIC:                              2.100e+05
BIC:                              2.102e+05

--------------------------------------------------
```

- Lacking information regarding variables (features)
- Increase the accuracy
- Accept the risk of over-fitting

FIT TEST DATA TO THE MULTIVARIATE LINEAR REGRESSION MODEL

	unit_number	prob_score	RUL
0	1	0.434556	186.682192
1	1	0.312216	195.958447
2	1	0.480827	176.964339
3	1	0.417941	184.346987
4	1	0.364827	193.219749
...	...	...	...
13091	100	0.837774	26.223130
13092	100	0.840024	26.175452
13093	100	0.803511	27.002617
13094	100	0.832147	24.360208
13095	100	0.900106	11.815974

HOW TO PREDICT THE RUL

Predict RUL using only the last recorded cycle of each unit

- If unit 1 has 200 recorded cycle, then we will use the RUL computed from the 200th cycle as our final prediction.

Predict using the smallest RUL found for each unit number

- If unit 1 has 200 recorded cycle, then we will use the smallest RUL computed from the 200 cycles as our final prediction (choose the smallest RUL among the 200 computed RULs)

HOW TO PREDICT THE RUL

Predict RUL using only the last recorded cycle of each unit

Average of:

(Predicted RULs – Real RULs)

```
option_1 = last_rul_by_unit - RUL_real_array
option_1 = option_1.astype(int)
np.mean(option_1)
```

```
15.34
```

Produces higher margin of error

Predict using the smallest RUL found for each unit number

Average of:

(Predicted RULs – Real RULs)

```
option_2 = min_rul_by_unit_array - RUL_real_array
option_2 = option_2.astype(int)
np.mean(option_2)
```

```
10.25
```

Produces lower margin of error

HOW TO PREDICT THE RUL

Predict RUL using only the last recorded cycle of each unit

Advantage:

More Cost-effective, as the RULs become longer you don't have to check the system too often.

Drawback:

Suffer higher risk that the system would fail

Predict using the smallest RUL found for each unit number

Advantage:

The risk that the system would fail is lower

Drawback:

Higher maintenance cost

LIMITATION

- The model’s fit is not high (R^2 = 0.65)

=> Have to build other models or compute other appropriate variables.

REFERENCES

- ChatGPT
- Gemini