Metro PT3 Final Project

6005 - Real Time Analytics

Presented by

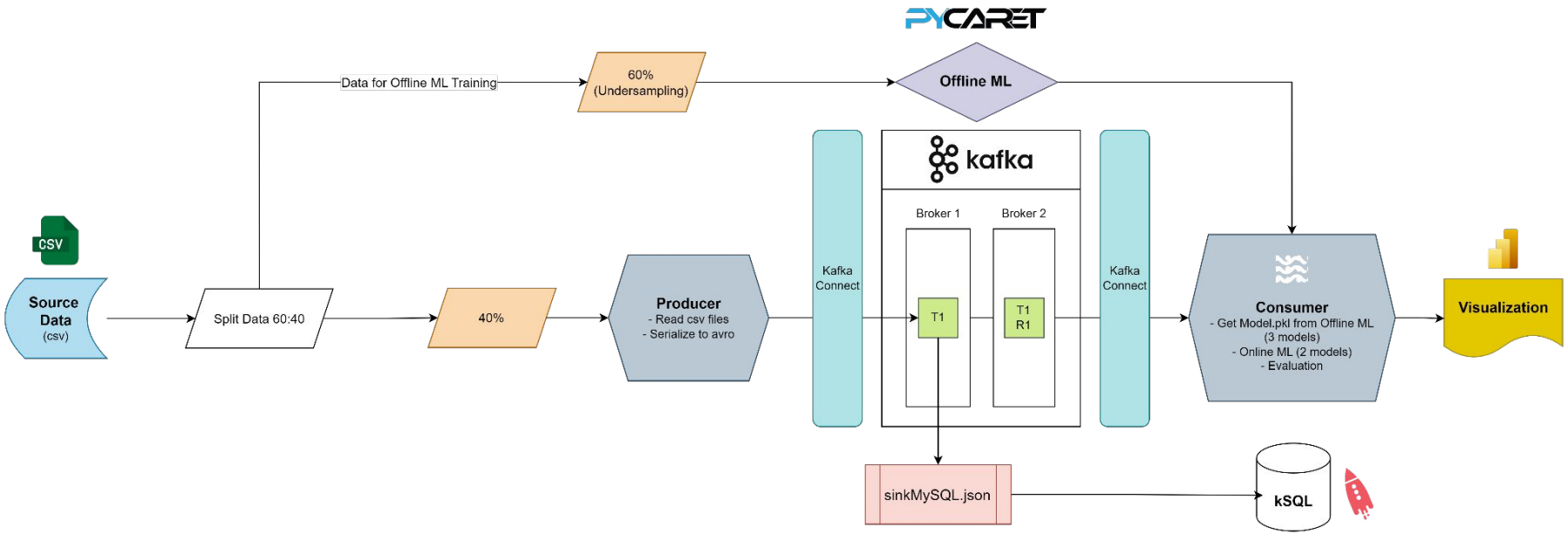
6520422022 Pakkawat Raksasin | 6520422024 Pitchayapa Chuwathanawit | 6520422030 Thanyalak Limsukhawat

Agenda

- Flow Diagram
- Exploratory Data Analysis (EDA)
- Real Time Application
  - Offline ML
  - Online ML
  - System Output
- System Simulation
- Summary and Suggestion

Diagram

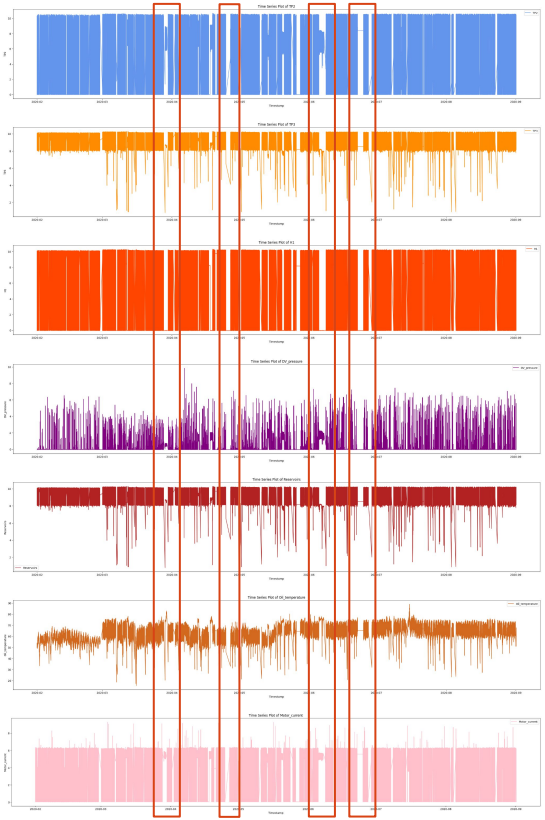
Flow Diagram



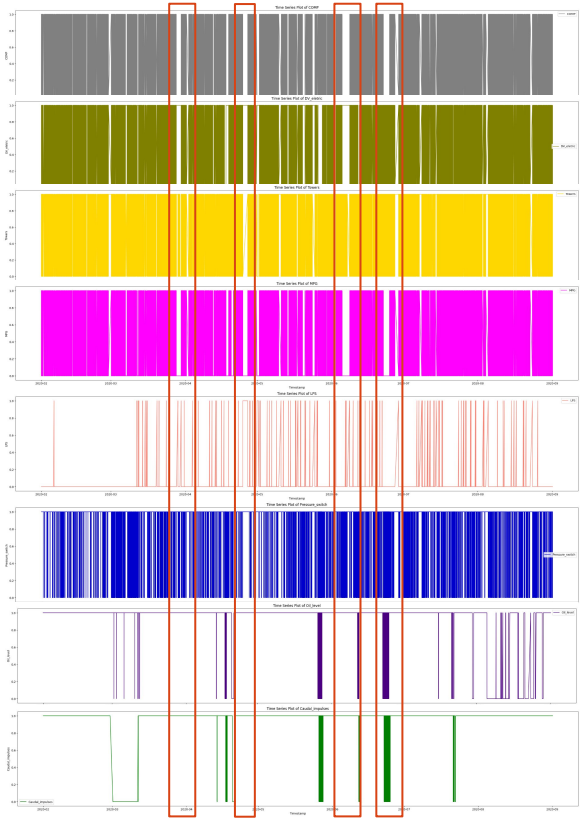
Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA)

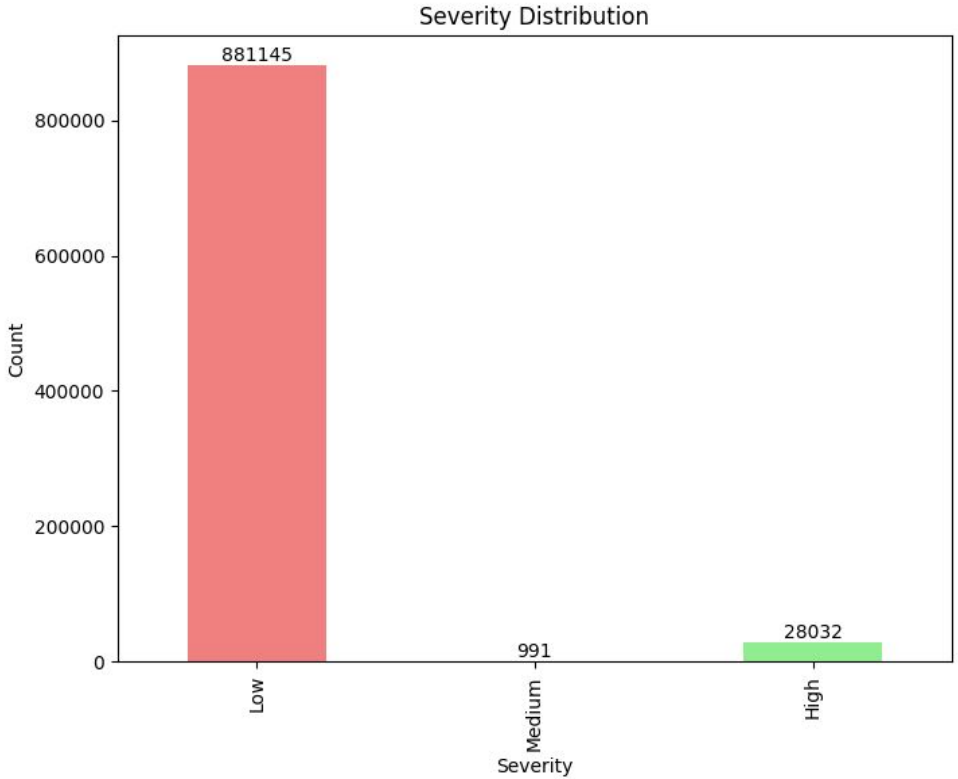
Devices



Censors



Exploratory Data Analysis (EDA)



Offline ML

Offline ML

Data Preparation:

- Drop unused features (e.g. Unnamed: 0)
- The timestamp was set as the index
- The target variable (severity) was found to be imbalance

```
import pandas as pd
import pycaret
from pycaret.classification import *
import numpy as np
from collections import Counter
from sklearn.datasets import make_classification
from imblearn.under_sampling import RandomUnderSampler

Offline_data = pd.read_csv('D:/NIDA/DADS6005/Finalproject/rev07_EDA/Offline_data_use.csv')
Offline_data = Offline_data.drop(columns=['Unnamed: 0'])
print(Offline_data.head())
# Create the DataFrame
df = pd.DataFrame(Offline_data)

# Convert 'timestamp' to datetime
df['timestamp'] = pd.to_datetime(df['timestamp'])

# Set 'timestamp' as the index
df.set_index('timestamp', inplace=True)
print(df.head())
```

Offline ML

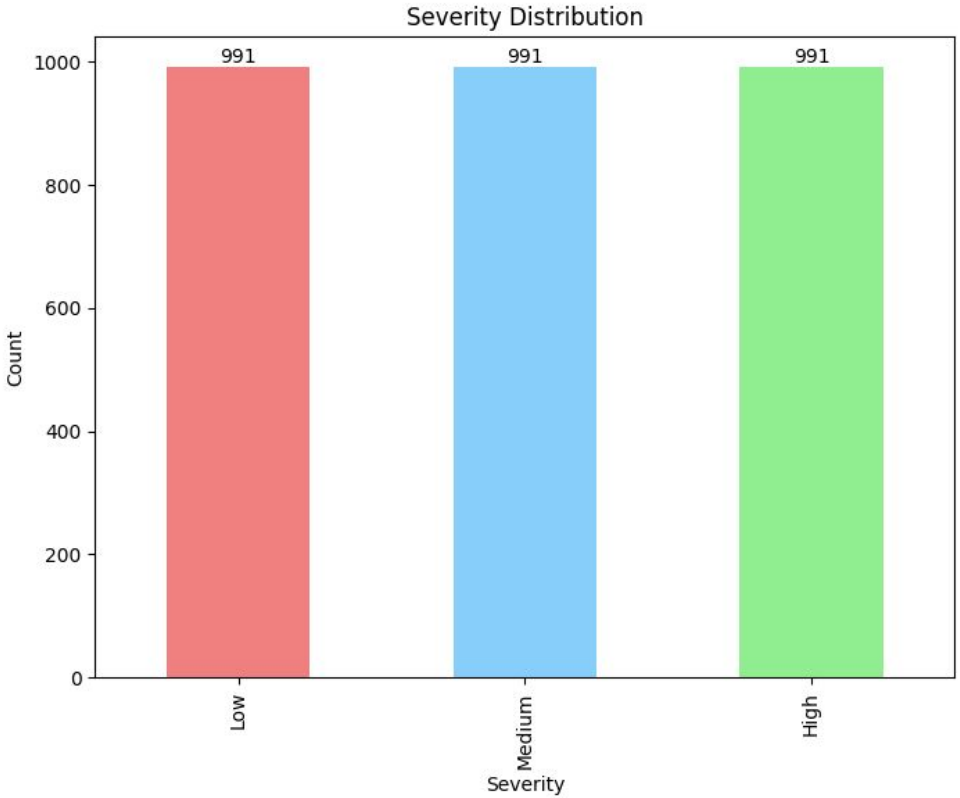
Resampling:

- Undersampling was applied to balance the data, cutting out the over-represented classes

```
rus = RandomUnderSampler(sampling_strategy='auto', random_state=42)

X_resampled, y_resampled = rus.fit_resample(df.drop(columns=['Severity']), df['Severity'])

df_resampled = pd.DataFrame(X_resampled, columns=df.columns.drop('Severity'))
df_resampled['Severity'] = y_resampled

print(df_resampled['Severity'].value_counts())
```

```
Severity
high      991
low       991
medium    991
Name: count, dtype: int64
```

Offline ML


Severity Distribution
991
991
991
1000
800
600
400
200
0
Count
Low
Medium
High
Severity

Offline ML

Model Training:

- The resampled data was used for training offline ML models
- PyCaret library was utilized to implement classification models
- The three models with the highest accuracy scores were selected: Gradient Boosting Classifier, Light Gradient Boosting Machine, Extra Trees Classifier

```
# Use PyCaret for classification
clf = setup(df_resampled, target='Severity', session_id = 123)
best = clf.compare_models()
evaluate_model(best)
predict_model(best)
# Save  model
save_model(best, 'Model_gbc')

lightgbm = create_model('lightgbm')
save_model(lightgbm, 'Model_lightgbm')

rf = create_model('rf')
save_model(rf, 'Model_rf')
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
gbc	Gradient Boosting Classifier	0.9889	0.0000	0.9889	0.9891	0.9890	0.9834	0.9835	0.1890
rf	Random Forest Classifier	0.9885	0.9976	0.9885	0.9886	0.9885	0.9827	0.9828	0.0470
lightgbm	Light Gradient Boosting Machine	0.9885	0.9990	0.9885	0.9886	0.9885	0.9827	0.9828	0.4830
et	Extra Trees Classifier	0.9875	0.9974	0.9875	0.9876	0.9875	0.9813	0.9813	0.0440
lr	Logistic Regression	0.9856	0.0000	0.9856	0.9858	0.9856	0.9784	0.9785	0.6530
dt	Decision Tree Classifier	0.9837	0.9884	0.9837	0.9839	0.9836	0.9755	0.9756	0.0090
knn	K Neighbors Classifier	0.9832	0.9949	0.9832	0.9833	0.9832	0.9748	0.9748	0.5020
ridge	Ridge Classifier	0.9731	0.0000	0.9731	0.9737	0.9731	0.9596	0.9599	0.0090
lda	Linear Discriminant Analysis	0.9731	0.0000	0.9731	0.9737	0.9731	0.9596	0.9599	0.0110
svm	SVM - Linear Kernel	0.9693	0.0000	0.9693	0.9707	0.9694	0.9539	0.9545	0.0140
nb	Naive Bayes	0.9486	0.9890	0.9486	0.9505	0.9485	0.9229	0.9239	0.0090
ada	Ada Boost Classifier	0.8535	0.0000	0.8535	0.8290	0.8345	0.7803	0.7972	0.0310
qda	Quadratic Discriminant Analysis	0.3335	0.0000	0.3335	0.1112	0.1668	0.0000	0.0000	0.0150
dummy	Dummy Classifier	0.3321	0.5000	0.3321	0.1103	0.1655	0.0000	0.0000	0.0080

Online ML

Online ML

Data Streaming Simulation

- The remaining 40% of the data was used for streaming simulation and online ML training and prediction
- The River ML was utilized to implement classification models
- The two models with the classification methods were selected: Hoeffding Tree Classifier and Extremely Fast Decision Tree Classifier

```python
from sklearn.metrics import accuracy_score
from river import metrics, tree, anomaly, compose
import random

# Define online models
model_hoeff = tree.HoeffdingTreeClassifier(grace_period=100)
model_ex = tree.ExtremelyFastDecisionTreeClassifier(grace_period=100)

# Define accuracy metrics for online models
accuracy_hoeff = metrics.Accuracy()
accuracy_ex = metrics.Accuracy()
accuracy_sgt = metrics.Accuracy()
```

```python
# Online model 1 (HoeffdingTreeClassifier)
y_pred_hoeff = model_hoeff.predict_one(data)
model_hoeff.learn_one(data, user.Severity)
print("\nOnline Prediction (Hoeffding) = ", y_pred_hoeff)

accuracy_hoeff.update(user.Severity, y_pred_hoeff)
print("Accuracy (Online Model Hoeffding):", accuracy_hoeff.get())

# Online model 2 (ExtremelyFastDecisionTreeClassifier)
y_pred_ex = model_ex.predict_one(data)
model_ex.learn_one(data, user.Severity)
print("\nOnline Prediction (ExtremelyFastDecision) = ", y_pred_ex)

accuracy_ex.update(user.Severity, y_pred_ex)
print("Accuracy (Online Model ExtremelyFastDecision):", accuracy_ex.get())
```

System Output

Producer Output

```
timestamp            2020-08-08 02:44:43
TP2                                -0.014
TP3                                  8.4
H1                                 8.388
DV_pressure                        -0.022
Reservoirs                           8.4
Oil_temperature                     59.3
Motor_current                     0.0425
COMP                                 1.0
DV_eletric                           0.0
Towers                               1.0
MPG                                  1.0
LPS                                  0.0
Pressure_switch                      1.0
Oil_level                            0.0
Caudal_impulses                      1.0
Severity                             low
Name: 316, dtype: object
```

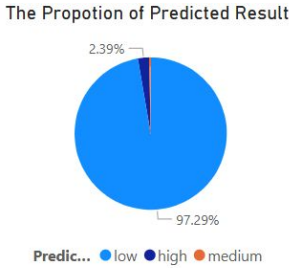
Consumer Output

```
Received data: {'timestamp': '2020-04-29 11:43:08', 'TP2': -0.0139999999999993, 'TP3': 8.388, 'H1': 8.38, 'DV_pressure': -0.021999999
9999984, 'Reservoirs': 8.392, 'Oil_temperature': 52.625000000000014, 'Motor_current': 0.0424999999999995, 'COMP': 1, 'DV_eletric': 0,
 'Towers': 1, 'MPG': 1, 'LPS': 0, 'Pressure_switch': 1, 'Oil_level': 1, 'Caudal_impulses': 1, 'Severity': 'low'}
Transformation Pipeline and Model Successfully Loaded

Predicted_Model_rf low  VS Actual= low
Accuracy (Offline_Model_rf): 1.0
Transformation Pipeline and Model Successfully Loaded

Predicted_Model_lightgbm low  VS Actual= low
Accuracy (Offline_Model_lightgbm): 0.9904761904761905
Transformation Pipeline and Model Successfully Loaded

Predicted_Model_gbc low  VS Actual= low
Accuracy (Offline_Model_gbc): 0.9904761904761905

Online Prediction (Hoeffding) =  low
Accuracy (Online Model Hoeffding): 0.9666666666666667

Online Prediction (ExtremelyFastDecision) =  low
Accuracy (Online Model ExtremelyFastDecision): 0.9666666666666667
Best Model: Offline Model (Random Forest Classifier) with Accuracy: 1.0


Data successfully sent to Power BI.
```

Database Output

```
mysql> select * from MetroPT;
+---------------------+--------+--------+--------+-------------+-----------+-----------------+---------------+------+-----------+--------+------+------+-----------------+-----------+-------------
----+----------+
| timestamp           | TP2    | TP3    | H1     | DV_pressure | Reservoirs | Oil_temperature | Motor_current | COMP | DV_eletric | Towers | MPG  | LPS  | Pressure_switch | Oil_level | Caudal_impul
ses | Severity |
+---------------------+--------+--------+--------+-------------+-----------+-----------------+---------------+------+-----------+--------+------+------+-----------------+-----------+-------------
----+----------+
| 2020-07-20 11:41:56 | -0.008 |  9.118 |  9.104 |      -0.014 |     9.122 |          68.525 |          0.04 |    1 |         0 |      1 |    1 |    0 |               1 |         1 |
  1 | low      |
| 2020-05-03 03:55:49 | -0.012 |  8.932 |   8.92 |      -0.022 |     8.934 |           57.15 |        0.0425 |    1 |         0 |      1 |    1 |    0 |               1 |         1 |
  1 | low      |
| 2020-08-11 14:27:36 |  9.728 |  9.334 | -0.012 |      -0.016 |      9.33 |          65.175 |        6.0125 |    0 |         1 |      0 |    0 |    0 |               1 |         0 |
  1 | low      |
| 2020-08-18 16:01:16 |  -0.01 |  8.944 |  8.932 |      -0.016 |     8.948 |          65.825 |          0.04 |    1 |         0 |      1 |    1 |    0 |               1 |         0 |
  1 | low      |
| 2020-03-03 04:37:52 | -0.016 |  8.278 |  8.268 |      -0.028 |     8.276 |          57.825 |          0.04 |    1 |         0 |      1 |    1 |    0 |               1 |         1 |
  0 | low      |
| 2020-04-07 00:38:59 | -0.012 |   9.07 |  9.058 |      -0.022 |      9.07 |           66.45 |        3.8675 |    1 |         0 |      1 |    1 |    0 |               1 |         1 |
  1 | low      |
| 2020-04-28 17:55:24 | -0.014 |   8.33 |  8.318 |      -0.022 |      8.33 |          55.075 |        0.0425 |    1 |         0 |      1 |    1 |    0 |               1 |         1 |
  1 | low      |
| 2020-06-16 17:40:25 | 10.406 | 10.034 | -0.012 |       -0.02 |     10.03 |           70.55 |        6.0825 |    0 |         1 |      1 |    0 |    0 |               1 |         1 |
  1 | low      |
```

Streaming Data Visualization

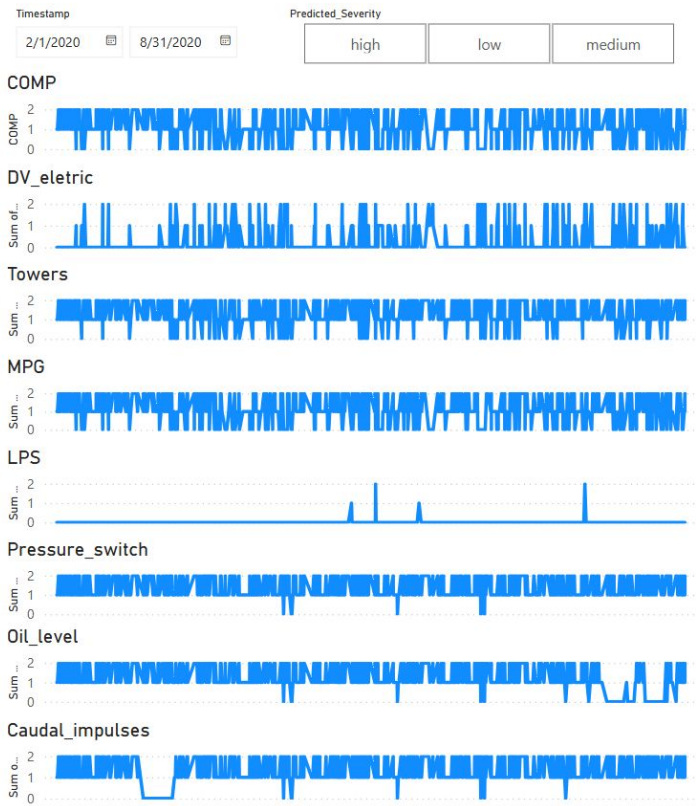


1253
Total Record

high
Actual Severity

high
Predicted Severity

Timestamp	TP2	TP3	H1	DV_pressure	Reservoirs	Oil_temperature	Motor_current
08/31/20 07:54:11 PM	-0.01	8.71	8.70	-0.02	8.71	63.45	0.04
08/31/20 07:53:32 PM	-0.01	8.77	8.76	-0.02	8.77	63.45	0.04
08/31/20 07:53:32 PM	-0.01	8.77	8.76	-0.02	8.77	63.45	0.04
08/31/20 02:46:45 PM	-0.01	9.38	9.37	-0.02	9.38	68.35	3.63
08/31/20 03:43:03 AM	10.31	9.93	-0.01	-0.02	9.93	63.17	6.12
08/31/20 03:41:23 AM	7.47	8.11	-0.03	-0.02	8.11	57.20	5.52
08/31/20 03:41:23 AM	7.47	8.11	-0.03	-0.02	8.11	57.20	5.52
08/30/20 09:38:34 PM	10.22	9.82	-0.01	-0.02	9.81	70.72	6.09
08/30/20 08:58:45 PM	-0.01	9.14	9.13	-0.02	9.14	69.03	3.67
08/30/20 02:48:59 PM	-0.01	9.50	9.48	-0.02	9.50	66.95	3.75
08/30/20 05:21:41 AM	-0.01	8.89	8.87	-0.02	8.89	59.78	0.04
08/30/20 04:37:34 AM	-0.01	8.99	8.98	-0.02	8.99	55.23	0.04
08/30/20 04:37:34 AM	-0.01	8.99	8.98	-0.02	8.99	55.22	0.04
08/29/20 08:25:16 PM	-0.01	9.52	9.51	-0.02	9.52	67.65	3.65
08/29/20 02:52:14 PM	-0.01	9.87	9.86	-0.02	9.88	71.57	3.67
08/29/20 09:17:03 AM	9.32	8.90	-0.01	-0.02	8.90	64.25	5.96
08/29/20 05:32:51 AM	-0.03	10.13	10.13	-0.02	10.13	65.88	3.79
08/28/20 08:04:52 PM	10.43	10.06	-0.01	-0.02	10.05	67.78	6.24
08/28/20 08:04:52 PM	10.43	10.06	-0.01	-0.02	10.05	67.78	6.24
08/27/20 01:52:49 PM	-0.01	8.19	8.18	-0.01	8.20	64.72	0.04
08/27/20 03:54:03 AM	-0.01	8.86	8.85	-0.02	8.86	61.10	0.04



System Simulation

Summary and Suggestion

Summary

- The anomaly detection system successfully integrated both offline and online machine learning models
- The streaming data successfully logged into the Database
- The use of Kafka for data streaming and Power BI for real-time visualization

Suggestion

- Data Augmentation
- Feature Engineering
- Model Algorithms and Optimization
- System Scalability
- Real-Time Alerts System

Thank You