

PORTFOLIO

แฟ้มสะสมผลงานประกอบการสัมภาษณ์



นายภูวดล วรบุตร

วิทยาศาสตร์บัณฑิต

มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

ประวัติส่วนตัว

จุดมุ่งหมายในการทำงาน

ผมเป็นนักศึกษาจบใหม่มีความสนใจในสายงาน Data เนื่องจาก ตอนทำโปรเจกต์มีโอกาสได้ทำโปรเจกต์เกี่ยวกับ Machine Learning ได้ออกแบบโมเดลทำยังงี้ถึงจะได้โมเดลที่ดีที่สุดและพอได้ศึกษาจริงๆเลยมีความชอบใน Machine Learning เพราะมีการทำอะไรสนุกๆ ได้หลายอย่างและตอนฝึกงานได้เข้าไปอยู่ในทีม Data และมีโอกาสได้ทำงานไม่ว่าจะเป็นการเขียนสคริปต์สำหรับแยกไฟล์รูปแบบ JSON ด้วยภาษา Python การแกะหาที่มาของข้อมูลจากไฟล์Macro และการทำ ETL โดยใช้ภาษา C# แต่ตอนนั้นไม่มีความรู้และไม่เคยเรียนมาก่อนและมีเวลาจำกัด มันเป็นเรื่องที่ท้าทายดีถึงแม้จะปวดหัวไปบ้าง แต่ก็ชอบในสายงานนี้รู้สึกทุกอย่างที่ทำในสายงานนี้เป็นอะไรที่ต้องหาความรู้อยู่ตลอดเวลาและดูสนุกไปหมด

ข้อมูลส่วนตัว

ชื่อ นายกวดล วรรณบุตร
ชื่อเล่น อีต อายุ 24 ปี
วัน/เดือน/ปีเกิด 22 ตุลาคม 2542
กรุ๊ปเลือด A ศาสนา พุทธ
เชื้อชาติ ไทย สัญชาติ ไทย

ประวัติการศึกษา

2563 - 2566

มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

คณะ วิทยาศาสตร์และเทคโนโลยี

สาขา วิทยาการการจัดการข้อมูล

GPA 3.26

2559 - 2562

โรงเรียนพระแม่สกลสงเคราะห์

แผนการเรียนสาย วิทยาศาสตร์ - คณิต

Skills

Programming Language

- Python (Pandas , Numpy , Scikit-learn , TensorFlow , Matplotlib) (Good)
- C# (Good)
- Golang (Fair)

Database

- SQL (Good)
- PL/SQL (Fair)

Web Developer

- HTML & CSS (Fair)
- JavaScript (Fair)

Tool

- Microsoft Office
- Rapidminer
- Visual Studio Code
- Visual Studio 2022
- Google Colab
- Jupyter Notebook

Database

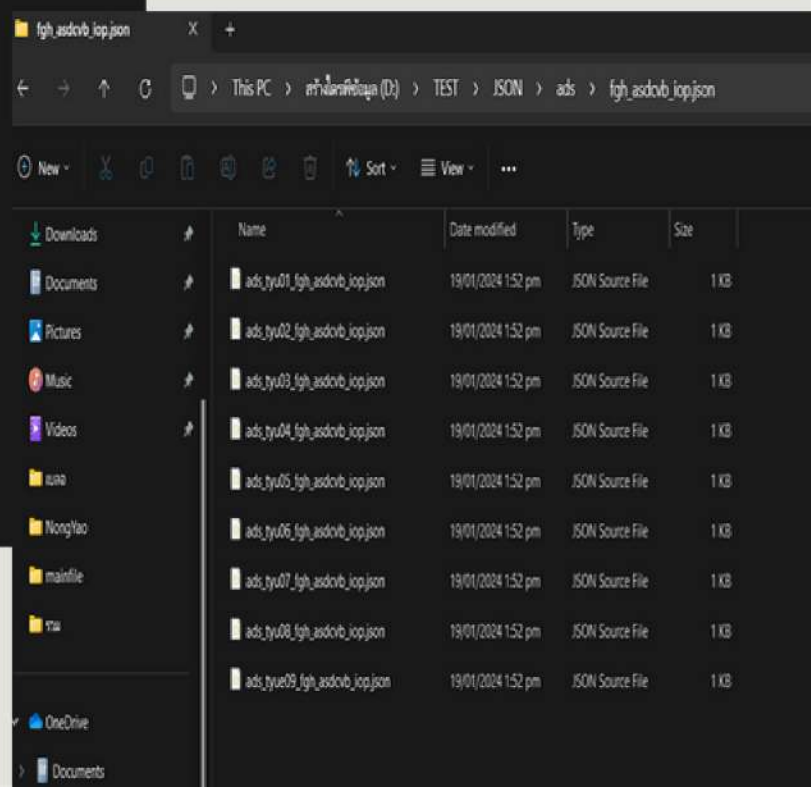
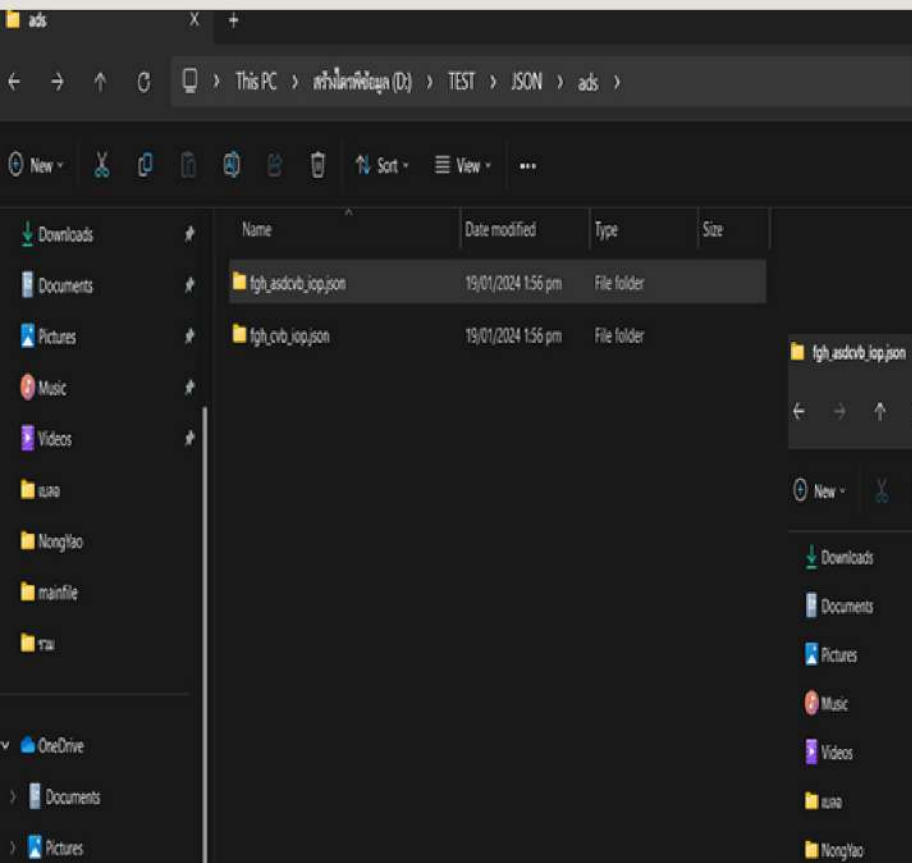
- Microsoft SQL Server
- MySQL
- SQL Developer

Data Visualization

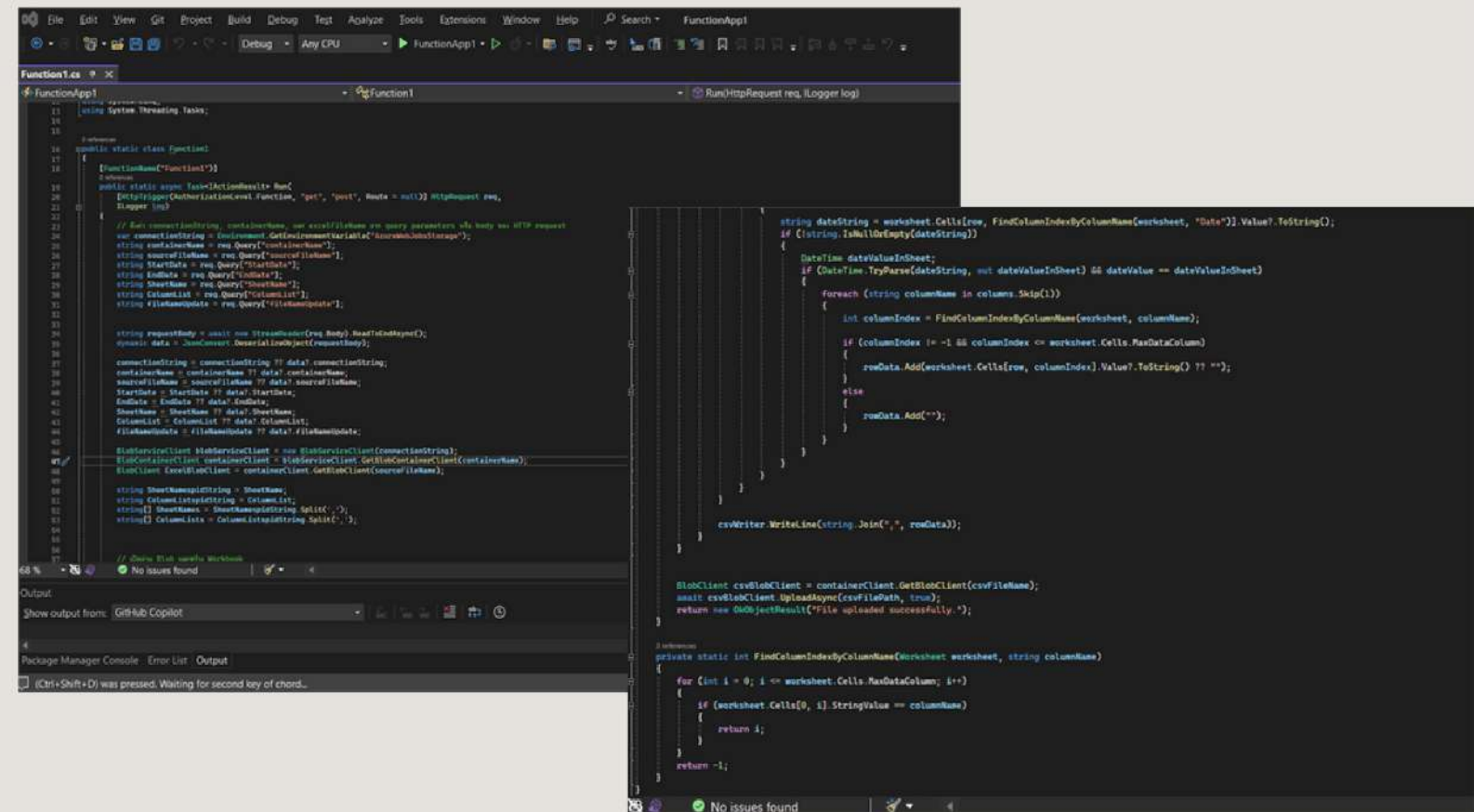
- Power BI (DAX)
- Looker Studio

CREATE SCRIPT FOR EXTRACT JSON FROMAT FILES WITH PYTHON

```
File Edit Selection View Go Run Terminal Help
Python_Batch > .py > get_destination_folder
1 import os
2 import shutil
3
4 def get_destination_folder(prefix, subfolder=None):
5     if subfolder:
6         return os.path.join(r'D:\TEST\JSON', prefix, subfolder)
7     else:
8         return os.path.join(r'D:\TEST\JSON', prefix)
9
10 def move_file(source_path, destination_path):
11     shutil.move(source_path, destination_path)
12
13 def move_files_to_folders(folder_path, prefixes):
14     for filename in os.listdir(folder_path):
15         file_path = os.path.join(folder_path, filename)
16         moved = False
17         for prefix in prefixes:
18             if filename.startswith(prefix):
19                 parts = filename.split('.')
20                 destination_folder = get_destination_folder(prefix, "-".join(parts[3:6]))
21                 destination_path = os.path.join(destination_folder, filename)
22                 if not os.path.exists(destination_folder):
23                     os.makedirs(destination_folder)
24                 move_file(file_path, destination_path)
25                 moved = True
26                 break
27         if not moved:
28             parts = filename.split('.')
29             destination_folder = get_destination_folder("others", "-".join(parts[3:]))
30             destination_path = os.path.join(destination_folder, filename)
31             if not os.path.exists(destination_folder):
32                 os.makedirs(destination_folder)
33             move_file(file_path, destination_path)
34
35 folder_path = r'D:\TEST\JSON\others'
36 # Prefixes to move
37 prefixes_to_move = ['ads', 'ert', 'nty']
38
39 move_files_to_folders(folder_path, prefixes_to_move)
```



Create C# Script on Azure Functions for transform data to csv format files



```
using System.Threading.Tasks;

public static class Function1
{
    [FunctionName("Function1")]
    public static async Task<ActionResult> Run(
        [Trigger(AuthorizationLevel.Function, "get", "post", Route = null)] HttpRequest req,
        ILogger log)
    {
        // Run connection string, containerName, use excelFileNames as query parameters via body via HTTP request
        var connectionString = Environment.GetEnvironmentVariable("AzureWebJobsStorage");
        string containerName = req.Query["containerName"];
        string sourceFileNames = req.Query["sourceFileNames"];
        string startDate = req.Query["startDate"];
        string endDate = req.Query["endDate"];
        string worksheet = req.Query["worksheet"];
        string columnList = req.Query["columnList"];
        string fileNameUpdate = req.Query["fileNameUpdate"];

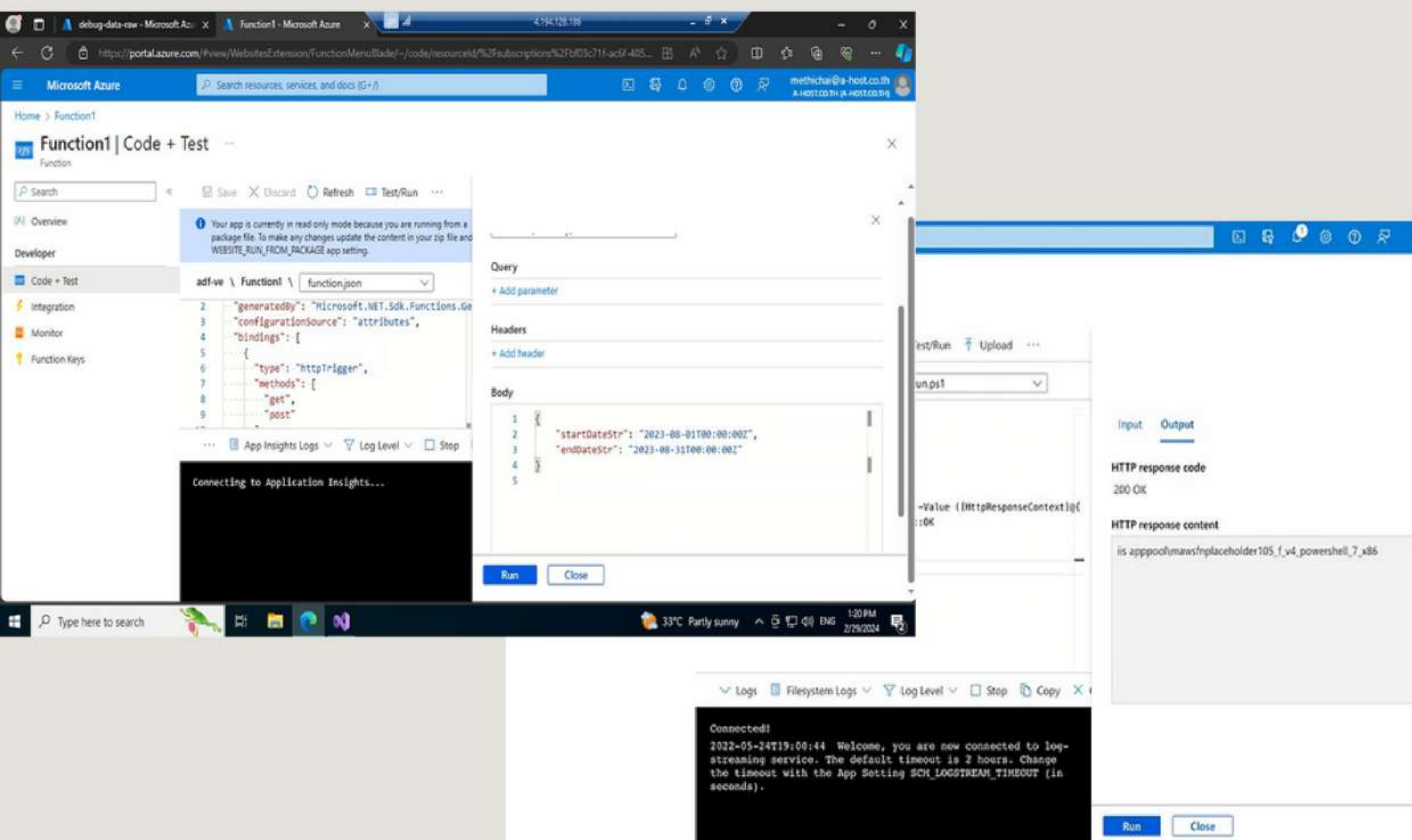
        string requestBody = await new StreamReader(req.Body).ReadToEndAsync();
        dynamic data = JsonConvert.DeserializeObject(requestBody);
        connectionString = connectionString ?? data?.connectionString;
        containerName = containerName ?? data?.containerName;
        sourceFileNames = sourceFileNames ?? data?.sourceFileNames;
        startDate = startDate ?? data?.startDate;
        endDate = endDate ?? data?.endDate;
        worksheet = worksheet ?? data?.worksheet;
        columnList = columnList ?? data?.columnList;
        fileNameUpdate = fileNameUpdate ?? data?.fileNameUpdate;

        BlobServiceClient blobServiceClient = new BlobServiceClient(connectionString);
        BlobContainerClient containerClient = blobServiceClient.GetBlobContainerClient(containerName);
        BlobClient excelBlobClient = containerClient.GetBlobClient(sourceFileNames);

        string SheetNameSplitting = worksheet;
        string[] SheetNames = SheetNameSplitting.Split(',');
        string[] ColumnList = ColumnList.Split(',');

        // Query Blob storage Worksheet
        BlobClient csvBlobClient = containerClient.GetBlobClient(csvFileName);
        await csvBlobClient.UploadAsync(csvFilePath, true);
        return new OkObjectResult("File uploaded successfully.");
    }

    private static int FindColumnIndexByColumnName(Worksheet worksheet, string columnName)
    {
        for (int i = 0; i < worksheet.Cells.MaxDataColumn; i++)
        {
            if (worksheet.Cells[0, i].StringValue == columnName)
            {
                return i;
            }
        }
        return -1;
    }
}
```



Microsoft Azure

Function1 | Code + Test

ad:\ve \ Function1 \ function.json

```
{
  "generatedBy": "Microsoft.NET.Sdk.Functions.Generator",
  "configurationSource": "attributes",
  "bindings": [
    {
      "type": "httpTrigger",
      "direction": "in",
      "methods": [
        "get",
        "post"
      ]
    }
  ]
}
```

Query

Headers

Body

```
{
  "startDate": "2023-08-01T00:00:00Z",
  "endDate": "2023-08-31T00:00:00Z"
}
```

Test/Run Upload

Input Output

HTTP response code: 200 OK

HTTP response content: is apppoolmanwfrplaceholder105_f_v4_powershell_7_x86

Connected! 2022-05-24T19:08:44 Welcome, you are now connected to log-streaming service. The default timeout is 2 hours. Change the timeout with the App Setting SCM_LOGSTREAM_TIMEOUT (in seconds).

Project

การทำนายราคาทองคำด้วยการเรียนรู้ของเครื่องบนอนุกรมเวลาแบบแบน

วัตถุประสงค์

เพื่อออกแบบ พัฒนา และทดสอบแบบจำลองราคาทองคำรายวัน และวิเคราะห์ประเมินประสิทธิภาพของแบบจำลอง ตลอดจนเปรียบเทียบประสิทธิภาพของแบบจำลองการทำนาย ราคาทองคำแบบต่างๆ โดยใช้ข้อมูลย้อนหลัง 10 ปี จากเว็บทองคำ.com หรือ สมาคมค้าทองคำ และจะแบ่งข้อมูลเป็น 3 ส่วน เป็น 7 วัน 14 วัน และ 30 วันย้อนหลัง

อัลกอริทึมที่ใช้

- Decision Tree Regression
- Random Forest Regression
- Artificial Neural Network
- Linear Regression
- Lasso Regression

เครื่องมือในการดำเนินโครงการ

Microsoft Excel , Python , Google Colab
มีการใช้ไลบรารีจาก Scikit - Learn และ TensorFlow

ทดสอบโมเดลด้วย

MAE , MSE , RMSE และโมเดล Baseline และสร้างแบบจำลองการซื้อขายเพื่อดูกำไรว่าโมเดลที่ได้ค่า MAE , MSE , RMSE น้อยที่สุดหรือดีที่สุดจะได้กำไรหรือขาดทุน


```

df = pd.read_excel('/content/drive/MyDrive/Project/plt.xlsx')
x123 = df.iloc[:, 1]
x123_normalized = (x123 - x123.min()) / (x123.max() - x123.min())
USD = df.iloc[:, 2]
USD_normalized = (USD - USD.min()) / (USD.max() - USD.min())

plt.scatter(x123_normalized, USD_normalized, c=USD_normalized, label='USD')
plt.xlabel("Goldspot Data (Normalized)")
plt.ylabel("USD Data (Normalized)")
plt.title("Scatter Plot of Normalized Goldspot vs Normalized USD")
plt.colorbar(label="USD (Normalized)")
plt.legend()
plt.show()

GOLDSPOT = df.iloc[:, 1]
DATE = df.iloc[:, 0]
plt.figure(figsize=(20, 6))
plt.plot(DATE, GOLDSPOT)
plt.grid(True)
plt.xlabel("year")
plt.ylabel("gold spot")
plt.show()

# sns.pairplot(df)

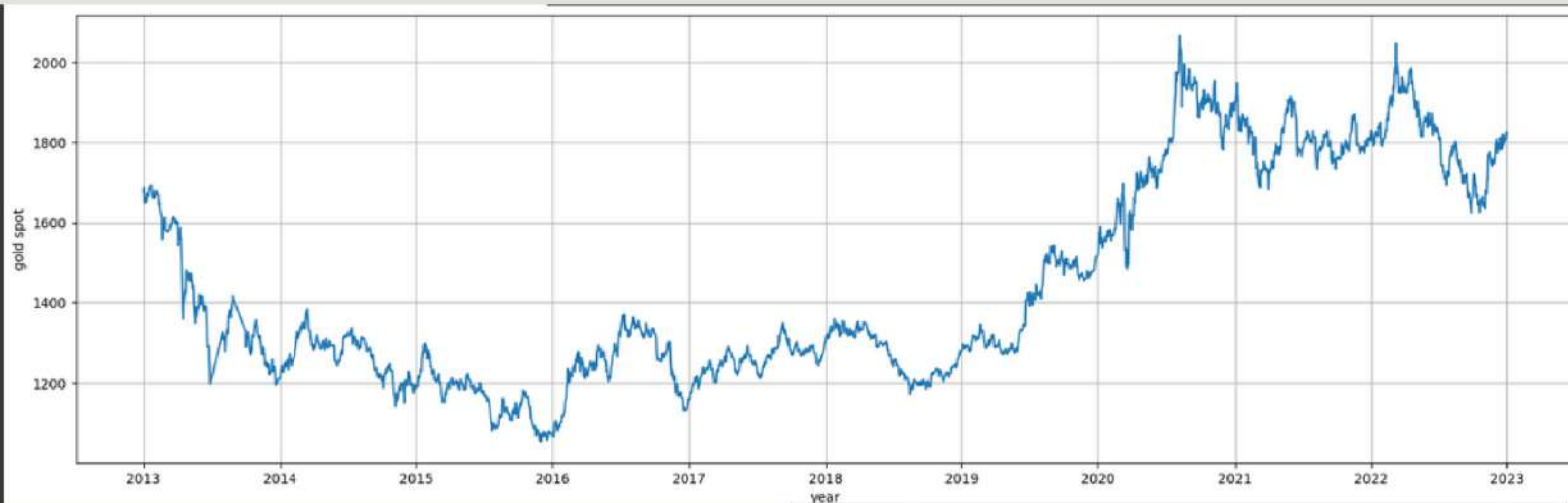
# train
data_train = pd.read_excel('/content/drive/MyDrive/Project/PDATA_Flannten_Time Series.xlsx', '7DAY')

# test
data_test = pd.read_excel('/content/drive/MyDrive/Project/PDATA_Flannten_Time Series.xlsx', 'DATA TEST 7DAY')

# train
train_target = data_train.iloc[:, 1] # Y
train_factor = data_train.iloc[:, 4:] # x

# test
test_target = data_test.iloc[:, 1] # Y
test_factor = data_test.iloc[:, 4:] # x

```



```

plt.figure(figsize=(15, 6))
plt.plot(test_target.values.flatten(), label='test_target')
plt.plot(predictions.flatten(), label='predictions')
plt.legend()
plt.show()

```

```

RES=test_target - predictions
sns.displot(RES,bins=30, kde=True)

```

```

lasso_model = Lasso(positive=True)
lasso_model.fit(train_factor, train_target)
predictions = lasso_model.predict(test_factor)
predictions = np.round(predictions, 2)

date = data_test.iloc[:, 0]

for idx, (d, target, pred) in enumerate(zip(date, test_target, predictions), start=1):
    date_str = d.strftime("%Y-%m-%d")
    # print(f"Row {idx}: Date:{date_str}, Actual: {target}, Predicted: {pred}")

mse = mean_squared_error(test_target, predictions)
print(f"MSE: {mse}")

mae = mean_absolute_error(test_target, predictions)
print(f"MAE: {mae}")

rmse = mean_squared_error(test_target, predictions, squared=False)
print(f"RMSE: {rmse}")

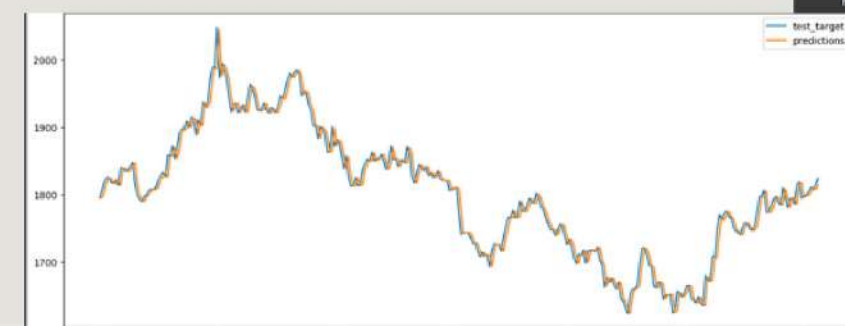
r_squared = r2_score(test_target, predictions)
print(f"R-squared: {r_squared}")

```

```

MSE: 223.2899517915311
MAE: 11.005667752443001
RMSE: 14.942889673404242

```



แบบจำลองการซื้อขาย

```

missed_opportunities_count = 0
no_trade_count = 0
buy_count = 0
profitable_buy_count = 0
loss_buy_count = 0
zero_profit_buy_count = 0
buy_sell_indicator_zero_count = 0

predictions = pd.DataFrame({'Predictions': predictions})
predictions = predictions.shift(periods=1)
predictions = np.array(predictions)
predictions
Actual1 = test_target
Actual2 = test_target.shift(-1)

buy_sell = []
profits = []

for idx, (d, actual1, actual2, predictions) in enumerate(zip(date, Actual1, Actual2, predictions), start=1):
    date_str = d.strftime('%Y-%m-%d')

    if actual1 < predictions:
        buy_sell_indicator = 1
        profit = actual2 - actual1
        action_description = "ซื้อ"
    else:
        if actual1 < actual2:
            buy_sell_indicator = 0
            profit = 0
            action_description = "เสียโอกาส"
        else:
            buy_sell_indicator = 0
            profit = 0
            action_description = "ไม่ทำการซื้อขาย"

    buy_sell.append(buy_sell_indicator)
    profits.append(profit)

    if action_description == "เสียโอกาส":
        missed_opportunities_count += 1
    elif action_description == "ไม่ทำการซื้อขาย":
        no_trade_count += 1
    elif action_description == "ซื้อ":
        buy_count += 1
        if profit > 0:
            profitable_buy_count += 1
        elif profit < 0:
            loss_buy_count += 1
        if profit == 0:
            zero_profit_buy_count += 1

```

Row 290: Date: 2022-12-12,	Actual 1: 1789.0,	Actual 2: 1784.5,	Predictions tomorrow: [1789.16],	Buy/Sell: 1,	Profit: -4.5,	Action: ซื้อ
Row 291: Date: 2022-12-13,	Actual 1: 1784.5,	Actual 2: 1810.5,	Predictions tomorrow: [1784.31],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 292: Date: 2022-12-14,	Actual 1: 1810.5,	Actual 2: 1797.0,	Predictions tomorrow: [1808.72],	Buy/Sell: 0,	Profit: 0,	Action: ไม่ทำการซื้อขาย
Row 293: Date: 2022-12-15,	Actual 1: 1797.0,	Actual 2: 1780.5,	Predictions tomorrow: [1797.19],	Buy/Sell: 1,	Profit: -16.5,	Action: ซื้อ
Row 294: Date: 2022-12-16,	Actual 1: 1780.5,	Actual 2: 1793.5,	Predictions tomorrow: [1781.],	Buy/Sell: 1,	Profit: 13.0,	Action: ซื้อ
Row 295: Date: 2022-12-17,	Actual 1: 1793.5,	Actual 2: 1795.0,	Predictions tomorrow: [1792.59],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 296: Date: 2022-12-19,	Actual 1: 1795.0,	Actual 2: 1786.0,	Predictions tomorrow: [1794.55],	Buy/Sell: 0,	Profit: 0,	Action: ไม่ทำการซื้อขาย
Row 297: Date: 2022-12-20,	Actual 1: 1786.0,	Actual 2: 1816.0,	Predictions tomorrow: [1786.07],	Buy/Sell: 1,	Profit: 30.0,	Action: ซื้อ
Row 298: Date: 2022-12-21,	Actual 1: 1816.0,	Actual 2: 1819.0,	Predictions tomorrow: [1814.22],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 299: Date: 2022-12-22,	Actual 1: 1819.0,	Actual 2: 1794.5,	Predictions tomorrow: [1818.36],	Buy/Sell: 0,	Profit: 0,	Action: ไม่ทำการซื้อขาย
Row 300: Date: 2022-12-23,	Actual 1: 1794.5,	Actual 2: 1799.0,	Predictions tomorrow: [1795.19],	Buy/Sell: 1,	Profit: 4.5,	Action: ซื้อ
Row 301: Date: 2022-12-24,	Actual 1: 1799.0,	Actual 2: 1799.0,	Predictions tomorrow: [1798.41],	Buy/Sell: 0,	Profit: 0,	Action: ไม่ทำการซื้อขาย
Row 302: Date: 2022-12-26,	Actual 1: 1799.0,	Actual 2: 1805.0,	Predictions tomorrow: [1798.63],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 303: Date: 2022-12-27,	Actual 1: 1805.0,	Actual 2: 1811.0,	Predictions tomorrow: [1804.25],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 304: Date: 2022-12-28,	Actual 1: 1811.0,	Actual 2: 1807.5,	Predictions tomorrow: [1810.39],	Buy/Sell: 0,	Profit: 0,	Action: ไม่ทำการซื้อขาย
Row 305: Date: 2022-12-29,	Actual 1: 1807.5,	Actual 2: 1816.0,	Predictions tomorrow: [1807.37],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 306: Date: 2022-12-30,	Actual 1: 1816.0,	Actual 2: 1824.0,	Predictions tomorrow: [1815.11],	Buy/Sell: 0,	Profit: 0,	Action: เสียโอกาส
Row 307: Date: 2022-12-31,	Actual 1: 1824.0,	Actual 2: nan,	Predictions tomorrow: [nan],	Buy/Sell: 0,	Profit: 0,	Action: ไม่ทำการซื้อขาย

ผลรวมของกำไร: 73.0

จำนวนการซื้อ: 96

จำนวนการซื้อกำไร: 47

จำนวนการซื้อขาดทุน: 46

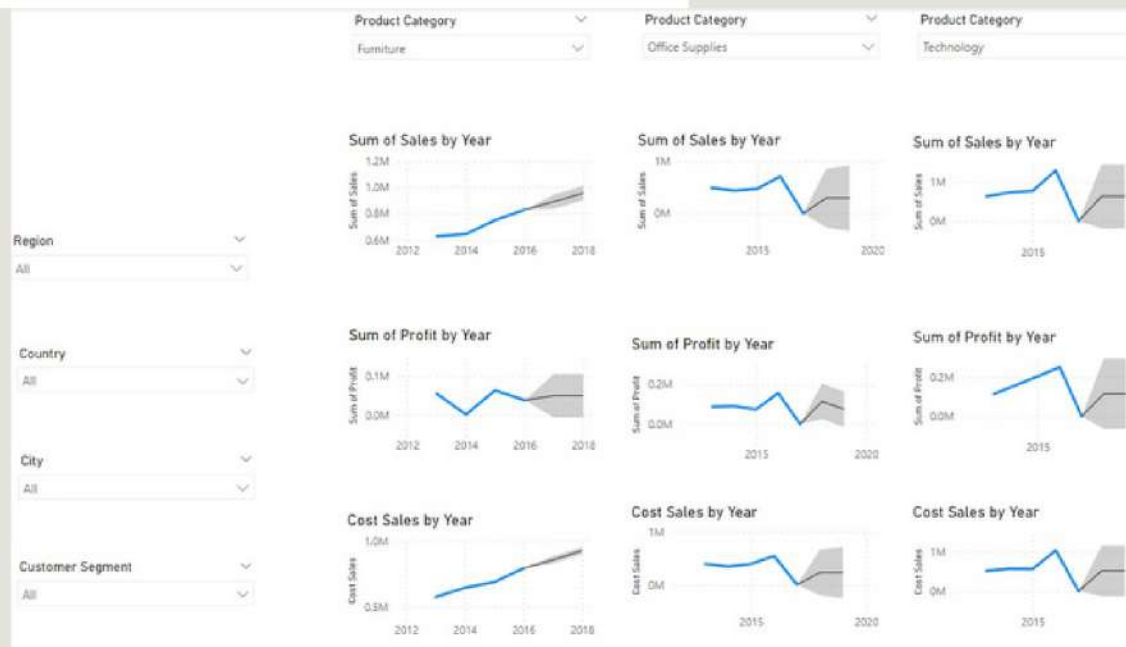
จำนวนการซื้อแต่ไม่ได้กำไร 3

จำนวนครั้งที่ไม่ซื้อ: 211

จำนวนครั้งที่ เสียโอกาส: 100

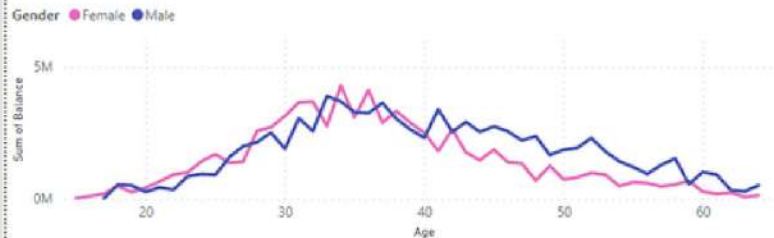
จำนวนครั้งที่ ไม่ทำการซื้อขาย: 111

Data Visualization

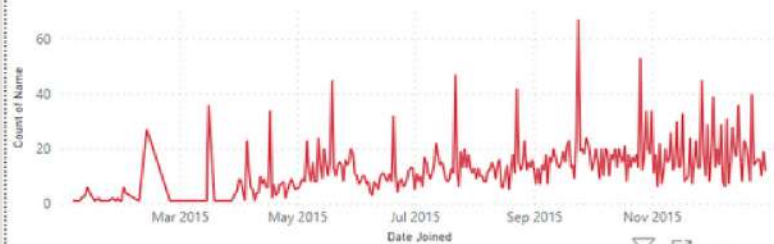


Data Visualization

Balance by Age and Gender



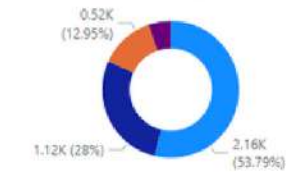
Total Customer



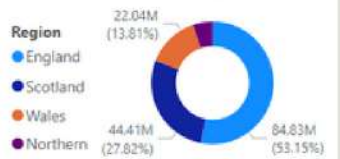
Number of Customer by Age and Gender



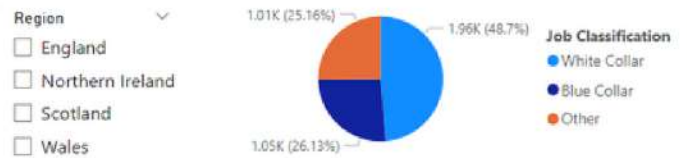
Count of Name by Region



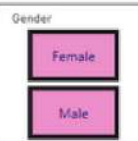
Balance by Region



Number Customer by Job

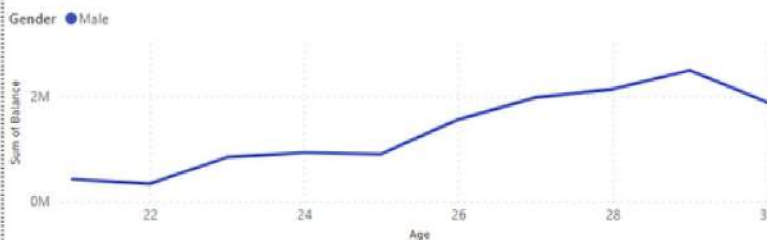


4014
Count of Name



Bank Data Analytics

Balance by Age and Gender



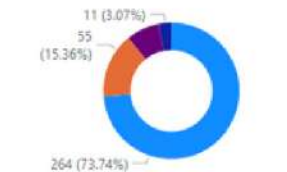
Total Customer



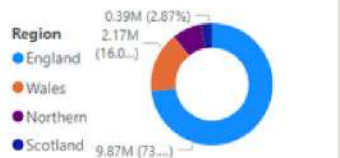
Number of Customer by Age and Gender



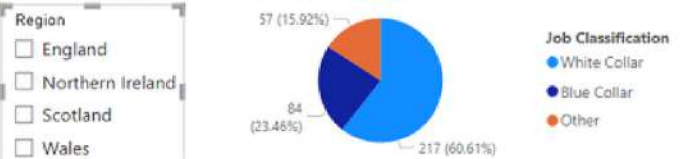
Count of Name by Region



Balance by Region



Number Customer by Job



358
Count of Name



Bank Data Analytics

SQL - ETL

The image displays a SQL Server environment with the Object Explorer on the left and SQL Query windows on the right. The Object Explorer shows the database structure for 'MIND\SQLXPRESS (SQL Server 16)' with tables like 'dbo.D_EMPLOYEE_ORA', 'dbo.D_DEPARTMENT_ORA', and 'dbo.F_SALARY_ORA'. The SQL Query windows show the following code:

```

USE Data_Phuvadol;
GO

CREATE TABLE D_EMPLOYEE_ORA(
    EMPLOYEE_CD INT,
    EMPLOYEE_NAME NVARCHAR(200),
    EMPLOYEE_EMAIL NVARCHAR(50),
    EMPLOYEE_PHONE_NUMBER NVARCHAR(200),
    HIRE_DATE DATE,
    LAST_DATE NVARCHAR(50),
    JOB_TITLE NVARCHAR(35),
    EMPLOYEE_COMMISSION DECIMAL(5,2),
    MANAGER_CD INT,
    MANAGER_NAME NVARCHAR(200),
    ETL_DATE DATETIME2(6),
    ETL_LAST_UPDATE DATETIME2(6)
);

USE Data_Phuvadol;
GO

CREATE TABLE D_DEPARTMENT_ORA(
    DEPARTMENT_ID INT,
    DEPARTMENT_NAME NVARCHAR(100),
    STREET_ADDRESS NVARCHAR(200),
    POSTAL_CODE NVARCHAR(10),
    CITY NVARCHAR(50),
    STATE_PROVINCE NVARCHAR(50),
    COUNTRY_NAME NVARCHAR(50),
    REGION_NAME NVARCHAR(50),
    ETL_DATE DATETIME2(6),
    ETL_LAST_UPDATE DATETIME2(6)
);

USE Data_Phuvadol;
GO

CREATE OR ALTER VIEW F_SALARY_ORA_V AS
SELECT
    YEAR(GETDATE()) AS YEAR_CD,
    CONVERT(VARCHAR(6), GETDATE(), 112) AS MONTH_CD,
    EMP.EMPLOYEE_ID AS EMPLOYEE_CD,
    J.JOB_ID AS JOB_CD,
    ISNULL(D.DEPARTMENT_ID, 0) AS DEPARTMENT_CD,
    ISNULL(CONVERT(VARCHAR, EMP.SALARY), '0') AS EMPLOYEE_SALARY,
    ISNULL(CONVERT(VARCHAR, MAX_SALARY), '0') AS MAX_JOB_SALARY,
    ISNULL(CONVERT(VARCHAR, MIN_SALARY), '0') AS MIN_JOB_SALARY,
    (
        SELECT SALARY
        FROM EMPLOYEES
        WHERE EMP.MANAGER_ID = EMPLOYEE_ID
    ) AS MANAGER_SALARY,
    DENSE_RANK() OVER (PARTITION BY EMP.JOB_ID ORDER BY EMP.SALARY DESC) AS JOB_SALARY_RANKING,
    SYSDATETIMEOFFSET(6) AS ETL_DATE,
    SYSDATETIMEOFFSET(6) AS ETL_LAST_UPDATE
FROM
    EMPLOYEES EMP
LEFT JOIN
    JOBS J ON EMP.JOB_ID = J.JOB_ID
LEFT JOIN
    DEPARTMENTS D ON EMP.DEPARTMENT_ID = D.DEPARTMENT_ID;

INSERT INTO d_employee_ora
SELECT * FROM D_EMPLOYEE_ORA_V;

INSERT INTO D_DEPARTMENT_ORA
SELECT * FROM D_DEPARTMENT_ORA_V;

INSERT INTO F_SALARY_ORA (
    YEAR_CD,
    MONTH_CD,
    EMPLOYEE_CD,
    JOB_CD,
    DEPARTMENT_CD,
    EMPLOYEE_SALARY,
    MAX_JOB_SALARY,
    MIN_JOB_SALARY,
    MANAGER_SALARY,
    JOB_SALARY_RANKING,
    ETL_DATE,
    ETL_LAST_UPDATE
)
SELECT
    YEAR_CD,
    MONTH_CD,
    EMPLOYEE_CD,
    JOB_CD,
    DEPARTMENT_CD,
    EMPLOYEE_SALARY,
    MAX_JOB_SALARY,
    MIN_JOB_SALARY,
    MANAGER_SALARY,
    JOB_SALARY_RANKING,
    ETL_DATE,
    ETL_LAST_UPDATE
FROM F_SALARY_ORA_V;
    
```

The SQL Query window on the right shows the creation of two views: 'D_EMPLOYEE_ORA_V' and 'D_DEPARTMENT_ORA_V'. The 'D_EMPLOYEE_ORA_V' view selects data from the 'EMPLOYEES' table, joining with 'JOBS' and 'DEPARTMENTS' tables. The 'D_DEPARTMENT_ORA_V' view selects data from the 'DEPARTMENTS' table, joining with 'CITIES' and 'STATES' tables. Both views include ETL metadata columns for tracking data extraction and updates.

ETL - Python

```
Create_Excel.py > ...
1 import pandas as pd
2 from datetime import datetime, timedelta
3 import random
4 import os
5
6 # สร้างโฟลเดอร์หากยังไม่มี
7 folder_path = r'D:\Phuvadol\skill\ETL\file_excel'
8 if not os.path.exists(folder_path):
9     os.makedirs(folder_path)
10
11 for i in range(1, 21):
12     try:
13         start_date = datetime(2023, 1, 1)
14         end_date = datetime(2024, 3, 30)
15         date_range = [start_date + timedelta(days=random.randint(0, (end_date - start_date).days)) for _ in range(30)]
16
17         # สุ่มค่า Value
18         values = [random.randint(9, 999999) for _ in range(30)]
19
20         # สร้าง DataFrame
21         data = {'Date': date_range, 'Value': values}
22         df = pd.DataFrame(data)
23
24         # บันทึกเป็นไฟล์ Excel
25         file_path = os.path.join(folder_path, f'data_{i}.xlsx')
26         df.to_excel(file_path, index=False)
27         print(f"ไฟล์ '{file_path}' ถูกสร้างเรียบร้อยแล้ว")
28     except Exception as e:
29         print(f"เกิดข้อผิดพลาดในการสร้างไฟล์ที่ '{file_path}': {e}")
30
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS TERMINAL POSTMAN CONSOLE

ไฟล์ 'D:\Phuvadol\skill\ETL\file_excel\data_15.xlsx' ถูกสร้างเรียบร้อยแล้ว
ไฟล์ 'D:\Phuvadol\skill\ETL\file_excel\data_16.xlsx' ถูกสร้างเรียบร้อยแล้ว
ไฟล์ 'D:\Phuvadol\skill\ETL\file_excel\data_17.xlsx' ถูกสร้างเรียบร้อยแล้ว
ไฟล์ 'D:\Phuvadol\skill\ETL\file_excel\data_18.xlsx' ถูกสร้างเรียบร้อยแล้ว
ไฟล์ 'D:\Phuvadol\skill\ETL\file_excel\data_19.xlsx' ถูกสร้างเรียบร้อยแล้ว
ไฟล์ 'D:\Phuvadol\skill\ETL\file_excel\data_20.xlsx' ถูกสร้างเรียบร้อยแล้ว

```
Create_Script.py > ...
1 import os
2 import re
3 import pandas as pd
4
5 # รวบรวมข้อมูลของไฟล์ Excel ที่มีไฟล์ Excel
6 folder_path = r'D:\Phuvadol\skill\ETL\file_excel'
7 output_file_path = r'D:\Phuvadol\skill\ETL\file_excel\output.xlsx'
8
9 data = []
10
11 # วนลูปผ่านไฟล์ในโฟลเดอร์
12 for file_name in os.listdir(folder_path):
13     if file_name.endswith('.xlsx'):
14         file_path = os.path.join(folder_path, file_name)
15         df = pd.read_excel(file_path)
16         latest_date_row = df[df['Date'] == df['Date'].max()]
17         value = latest_date_row['Value'].values[0]
18         data.append({'File Name': file_name, 'Date': latest_date_row['Date'].values[0], 'Value': value})
19 result_df = pd.DataFrame(data)
20
21 # ฟังก์ชันสำหรับการตรวจสอบว่าเลขในชื่อไฟล์ถูกต้องหรือไม่
22 def is_number_in_filename(file_name):
23     match = re.search(r'\d+', file_name)
24     return match.group() if match else None
25 result_df['Number'] = result_df['File Name'].apply(is_number_in_filename)
26
27 # เลือกเฉพาะแถวที่ไม่มีข้อมูล
28 result_df_invalid = result_df[result_df['Number'].isna()]
29
30 # แสดงไฟล์ที่ไม่ผ่านการตรวจสอบ
31 print("Invalid files:", result_df_invalid)
32
33 # ตรวจสอบว่ามีไฟล์ที่ไม่ผ่านการตรวจสอบหรือไม่
34 if not result_df_invalid.empty:
35     print("Some files contain invalid numbers in their names. Please check and correct them.")
36 else:
37
38     def extract_number(file_name):
39         return int(re.search(r'\d+', file_name).group())
40
41 # เรียงลำดับตามเลขในชื่อไฟล์
42 result_df_sorted = result_df.sort_values(by='Number', key=lambda x: x.map(extract_number))
43 result_df_sorted.to_excel(output_file_path, index=False)
44 print("Data has been written to:", output_file_path)
45
```

	A	B	C	D
	File Name	Date	Value	Number
1				
2	data_1.xlsx	2024-03-25 12:0	42,542	1
3	data_2.xlsx	2024-02-25 12:0	167,569	2
4	data_3.xlsx	2024-03-26 12:0	559,429	3
5	data_4.xlsx	2024-03-22 12:0	410,521	4
6	data_5.xlsx	2024-03-24 12:0	133,807	5
7	data_6.xlsx	2024-03-13 12:0	540,340	6
8	data_7.xlsx	2024-02-28 12:0	808,734	7
9	data_8.xlsx	2024-03-27 12:0	301,296	8
10	data_9.xlsx	2024-03-22 12:0	511,233	9
11	data_10.xlsx	2024-03-02 12:0	616,982	10
12	data_11.xlsx	2024-03-26 12:0	194,369	11
13	data_12.xlsx	2024-03-25 12:0	915,112	12
14	data_13.xlsx	2024-03-21 12:0	71,794	13
15	data_14.xlsx	2024-03-25 12:0	160,240	14
16	data_15.xlsx	2024-03-26 12:0	498,812	15
17	data_16.xlsx	2024-03-04 12:0	772,618	16
18	data_17.xlsx	2024-03-28 12:0	40,168	17
19	data_18.xlsx	2024-03-27 12:0	459,792	18
20	data_19.xlsx	2024-03-26 12:0	907,617	19
21	data_20.xlsx	2024-03-13 12:0	998,538	20

Web Scraping Python

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# ดึงข้อมูลจากเว็บไซต์
response = requests.get("https://www.myhora.com/%E0%B8%AB%E0%B8%A7%E0%B8%A2/%E0%B8%AA%E0%B8%
soup = BeautifulSoup(response.text, 'html.parser')

# ดึงข้อมูลที่ต้องการจากเว็บไซต์
data = []
temp_row = []
a = soup.find_all('div', class_='colx')
del a[:28]

for i, div in enumerate(a, start=1):
    temp_row.append(div.text.strip())
    if i % 12 == 0:
        data.append(temp_row)
        temp_row = []

if temp_row:
    data.append(temp_row)

df = pd.DataFrame(data)

df.to_excel("หวย.xlsx", index=False, header=False)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1	เมษายน	เม.ย.	2567	67	803481	81	481	90	122 809 5 90		122 809 559 947	
2	16	มีนาคม	มี.ค.	2567	67	997626	26	626	78	509 571 3 78		509 571 329 794	
3	1	มีนาคม	มี.ค.	2567	67	253603	03	603	79	900 975 3 79		900 975 382 703	
4	16	กุมภาพันธ์	ก.พ.	2567	67	941395	95	395	43	056 330 3 43		056 330 375 587	
5	1	กุมภาพันธ์	ก.พ.	2567	67	607063	63	063	09	454 943 5 09		454 943 544 591	
6	17	มกราคม	ม.ค.	2567	67	105979	79	979	61	429 931 1 61		429 931 196 635	
7	30	ธันวาคม	ธ.ค.	2566	66	625544	44	544	89	600 648 4 89		600 648 456 882	
8	16	ธันวาคม	ธ.ค.	2566	66	356757	57	757	85	058 410 5 85		058 410 584 964	
9	1	ธันวาคม	ธ.ค.	2566	66	251097	97	097	91	055 265 0 91		055 265 092 280	
10	16	พฤศจิกายน	พ.ย.	2566	66	557990	90	990	14	346 412 7 14		346 412 778 961	
11	1	พฤศจิกายน	พ.ย.	2566	66	743951	51	951	63	335 913 0 63		335 913 019 349	
12	16	ตุลาคม	ต.ค.	2566	66	931446	46	446	44	167 398 2 44		167 398 272 970	
13	1	ตุลาคม	ต.ค.	2566	66	727202	02	202	66	324 355 4 66		324 355 426 615	
14	16	กันยายน	ก.ย.	2566	66	320812	12	812	46	037 699 0 46		037 699 057 344	
15	1	กันยายน	ก.ย.	2566	66	915478	78	478	91	521 596 2 91		521 596 291 692	
16	16	สิงหาคม	ส.ค.	2566	66	471782	82	782	67	431 739 7 67		431 739 737 742	
17	31	กรกฎาคม	ก.ค.	2566	66	260453	53	453	11	268 708 3 11		268 708 387 601	
18	16	กรกฎาคม	ก.ค.	2566	66	169530	30	530	62	261 384 0 62		261 384 066 780	
19	1	กรกฎาคม	ก.ค.	2566	66	922605	05	605	16	281 867 4 16		281 867 491 947	
20	16	มิถุนายน	มิ.ย.	2566	66	264872	72	872	30	519 628 2 30		519 628 202 874	
21	1	มิถุนายน	มิ.ย.	2566	66	125272	72	272	09	000 681 3 09		000 681 386 971	
22	16	พฤษภาคม	พ.ค.	2566	66	132903	03	903	99	678 739 6 99		678 739 693 731	
23	2	พฤษภาคม	พ.ค.	2566	66	843019	19	019	65	500 780 1 65		500 780 187 269	
24	16	เมษายน	เม.ย.	2566	66	984906	06	906	71	670 678 5 71		670 678 551 797	
25	1	เมษายน	เม.ย.	2566	66	087907	07	907	99	111 914 2 99		111 914 290 698	
26	16	มีนาคม	มี.ค.	2566	66	025873	73	873	73	420 800 3 73		420 800 355 544	
27	1	มีนาคม	มี.ค.	2566	66	417652	52	652	55	577 919 7 55		577 919 748 984	
28	16	กุมภาพันธ์	ก.พ.	2566	66	590417	17	417	80	195 664 3 80		195 664 377 523	
29	1	กุมภาพันธ์	ก.พ.	2566	66	297411	11	411	92	181 789 1 92		181 789 101 664	
30	17	มกราคม	ม.ค.	2566	66	812519	19	519	47	389 443 5 47		389 443 564 849	
31	30	ธันวาคม	ธ.ค.	2565	65	157196	96	196	58	007 522 2 58		007 522 250 425	
32	16	ธันวาคม	ธ.ค.	2565	65	845093	93	093	14	411 912 5 14		411 912 593 855	
33	1	ธันวาคม	ธ.ค.	2565	65	375805	05	805	08	170 786 4 08		170 786 409 421	
34	16	พฤศจิกายน	พ.ย.	2565	65	121789	89	789	64	532 722 1 64		532 722 157 973	
35	1	พฤศจิกายน	พ.ย.	2565	65	913106	06	106	70	722 839 3 70		722 839 343 922	
36	16	ตุลาคม	ต.ค.	2565	65	613106	06	106	15	037 158 6 15		037 158 606 799	

Association Rule - Python

```
import pandas as pd
import numpy as np
from mlxtend.frequent_patterns import apriori, association_rules
```

Python

```
df = pd.read_csv("D:\Phuvadol\Portfolio_ETC\Data Set\Market_Basket_Optimisation.csv")
df
```

Python

```
<>:1: DeprecationWarning: invalid escape sequence '\P'
<>:1: DeprecationWarning: invalid escape sequence '\P'
C:\Users\athit\AppData\Local\Temp\ipykernel_13800\1690963744.py:1: DeprecationWarning: invalid escape sequence '\P'
df = pd.read_csv("D:\Phuvadol\Portfolio_ETC\Data Set\Market_Basket_Optimisation.csv")
```

	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie	spinach	olive oil
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...
7495	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7496	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7497	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	eggs	frozen	yogurt	low fat	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In 2, Col 5 Go Live

```
df = df.apply(lambda row: ', '.join(row.dropna().astype(str)), axis=1)
df = pd.DataFrame(df, columns=["Products"])
df
```

Products

```
0      burgers,meatballs,eggs
1      chutney
2      turkey,avocado
3  mineral water,milk,energy bar,whole wheat rice...
4      low fat yogurt
...
7495  butter,light mayo,fresh bread
7496  burgers,frozen vegetables,eggs,french fries,ma...
7497      chicken
7498  escalope,green tea
7499  eggs,frozen smoothie,yogurt cake,low fat yogurt
```

7500 rows x 1 columns

```
data = list(df["Products"].str.split(","))
from mlxtend.preprocessing import TransactionEncoder
a = TransactionEncoder()
a_data = a.fit(data).transform(data)
df = pd.DataFrame(a_data, columns=a.columns_)
df = df.replace(False,0)
df
```

✓ 1.2s

Python

	asparagus	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	...	turkey	vegetables mix	water spray	white wine	whole weat flour	whole wheat pasta	whole wheat rice	yams	yogurt cake	zucchini
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	True	0	0	0	0	...	True	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	True	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
...
7495	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7496	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7497	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7498	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
7499	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	True	0

7500 rows x 120 columns


```

true_counts = df.sum()
sorted_counts = true_counts.sort_values(ascending=False)
print(sorted_counts)

```

```

mineral water    1787
eggs             1348
spaghetti        1306
french fries     1282
chocolate        1229
...
bramble          14
cream             7
napkins           5
water spray       3
asparagus        1
Length: 120, dtype: object

```

```

df = pd.DataFrame(a_data,columns=a.columns_)
df = df.replace(False,0)

df = apriori(df, min_support = 0.05, use_colnames = True, verbose = 1)
df = df.sort_values(by='support', ascending=False)
df['support_percent'] = df['support'] * 100
df

```

```

Processing 6 combinations | Sampling itemset size 3 2
c:\Users\athit\AppData\Local\Programs\Python\Python311\Lib\site-packages\mlxt
warnings.warn(

```

	support	itemsets	support_percent
16	0.238267	(mineral water)	23.826667
6	0.179733	(eggs)	17.973333
21	0.174133	(spaghetti)	17.413333
8	0.170933	(french fries)	17.093333
3	0.163867	(chocolate)	16.386667
12	0.132	(green tea)	13.2
15	0.1296	(milk)	12.96
13	0.098267	(ground beef)	9.826667
10	0.095333	(frozen vegetables)	9.533333
18	0.095067	(pancakes)	9.506667
0	0.0872	(burgers)	8.72
1	0.081067	(cake)	8.106667
4	0.0804	(cookies)	8.04
7	0.079333	(escalope)	7.933333
14	0.0764	(low fat yogurt)	7.64
19	0.071333	(shrimp)	7.133333
22	0.0684	(tomatoes)	6.84

```

df_ar = association_rules(df, metric = "confidence", min_threshold = 0.06)
df_ar_sorted = df_ar.sort_values(by='confidence', ascending=False)
df_ar_sorted = df_ar_sorted.round(2)
df_ar_sorted

```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(spaghetti)	(mineral water)	0.17	0.24	0.06	0.34	1.44	0.02	1.16	0.37
2	(chocolate)	(mineral water)	0.16	0.24	0.05	0.32	1.35	0.01	1.12	0.31
4	(eggs)	(mineral water)	0.18	0.24	0.05	0.28	1.19	0.01	1.06	0.19
1	(mineral water)	(spaghetti)	0.24	0.17	0.06	0.25	1.44	0.02	1.10	0.40
3	(mineral water)	(chocolate)	0.24	0.16	0.05	0.22	1.35	0.01	1.07	0.34
5	(mineral water)	(eggs)	0.24	0.18	0.05	0.21	1.19	0.01	1.04	0.21

Certificate



หลักสูตรวิทยาการการจัดการข้อมูล
มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ
ขอมอบเกียรติบัตรฉบับนี้ให้ไว้เพื่อแสดงว่า

นาย ภูวดล วรบุตร

ผ่านการอบรม หัวข้อ

Google Data Studio กับการวิเคราะห์ข้อมูลเชิงธุรกิจ
โครงการพัฒนาศมรรถนะวิชาชีพนักศึกษาประจำหลักสูตร

วันที่ ๑๐ กุมภาพันธ์ พ.ศ ๒๕๖๖

(ดร.รุจิรา กองนุ้ย)

ประธานหลักสูตรวิทยาการการจัดการข้อมูล

Made for free with Certify'em



Certificate

OF COMPLETION

ภูวดล วรบุตร

has successfully completed

Get started with Programming รวมถึงที่ต้องรู้ ก่อนเดินสู่เส้นทางโปรแกรมเมอร์

an online course offered by FutureSkill

03/04/2024



Futureskill has confirmed the identity of this individual and their participation in the course

Futureskill.co



Certificate

OF COMPLETION

ภูวดล วรรณกุล

has successfully completed
UpSkill Python Programming เส้นทางสู่ Developer
an online course offered by FutureSkill
11/01/2024



Futureskill has confirmed the identity of this individual and
their participation in the course



Futureskill.co



Certificate

OF COMPLETION

ภูวดล วรรณกุล

has successfully completed
จัดการฐานข้อมูลอย่างมีประสิทธิภาพ พร้อมวิเคราะห์ข้อมูลที่มีในไม่ที่อีกด้วย SQL Command
an online course offered by FutureSkill
19/12/2023



Futureskill has confirmed the identity of this individual and
their participation in the course



Futureskill.co

แพลตฟอร์มเพื่อการเรียนรู้ออนไลน์ตลอดชีวิต | กระทรวงการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรม

ภูวดล วรรณตร

ได้ผ่านการเรียนออนไลน์ตามเกณฑ์การวัดผลในรายวิชา

วิทยาการข้อมูลและการประยุกต์ใช้ (30 ชั่วโมงการเรียนรู้)

พัฒนารายวิชาโดย จุฬาลงกรณ์มหาวิทยาลัย

ผศ.ดร.ณัฐฉา นนุไพโรจน์

ผู้ช่วยศาสตราจารย์

ภาควิชาวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

แพลตฟอร์มเพื่อการเรียนรู้ออนไลน์ตลอดชีวิต | กระทรวงการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรม

ภูวดล วรรณตร

ได้ผ่านการเรียนออนไลน์ตามเกณฑ์การวัดผลในรายวิชา

การวิเคราะห์เชิงลึกกับมหาข้อมูล (10 ชั่วโมงการเรียนรู้)

พัฒนารายวิชาโดย มหาวิทยาลัยรังสิต

ผศ. ดร. เชษฐเนติ ศรีสอาน

คณบดี, ว. นวัตกรรมดิจิทัลและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยรังสิต

แพลตฟอร์มเพื่อการเรียนรู้ออนไลน์ตลอดชีวิต | กระทรวงการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรม

ภูวดล วรบุตร

ได้ผ่านการเรียนออนไลน์ตามเกณฑ์การวัดผลในรายวิชา

ความรู้พื้นฐานเกี่ยวกับบิกดาตา (10 ชั่วโมงการเรียนรู้)

พัฒนารายวิชาโดย มหาวิทยาลัยสุโขทัยธรรมาธิราช

ผศ.ดร. วุฒิชัย ร่มสายหยุด
สาขาวิชาวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยสุโขทัยธรรมาธิราช

แพลตฟอร์มเพื่อการเรียนรู้ออนไลน์ตลอดชีวิต | กระทรวงการอุดมศึกษา วิทยาศาสตร์ วิจัยและนวัตกรรม

ภูวดล วรบุตร

ได้ผ่านการเรียนออนไลน์ตามเกณฑ์การวัดผลในรายวิชา

ภาษาอังกฤษสำหรับเทคโนโลยีสารสนเทศ (10 ชั่วโมงการเรียนรู้)

พัฒนารายวิชาโดย มหาวิทยาลัยขอนแก่น

ผศ.ดร.กรรวิภา พูลผล
รองคณบดีฝ่ายแผนและสารสนเทศ
คณะมนุษยศาสตร์และสังคมศาสตร์ มหาวิทยาลัยขอนแก่น



สำนักวิทยบริการและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

ขอมอบวุฒิบัตรให้ไว้เพื่อแสดงว่า

นาย ภูวดล วรบุตร

เข้าร่วมอบรมแนะนำการใช้งาน
Google Workspace for Education Plus

ให้ไว้ ณ วันที่ 30 มกราคม 2566

ดร.เอกชัย เนาวนิช

ผู้อำนวยการสำนักวิทยบริการและเทคโนโลยีสารสนเทศ



หลักสูตรวิทยาการการจัดการข้อมูล
มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

ขอมอบเกียรติบัตรฉบับนี้ให้ไว้เพื่อแสดงว่า

ภูวดล วรบุตร

ผ่านการอบรม

กิจกรรมที่ ๑ หัวข้อ คุณธรรมการดำเนินชีวิตในยุคปัจจุบัน
โครงการพัฒนาสมรรถนะนักศึกษาตามกรอบศตวรรษที่ ๒๑

วันที่ ๒๖ กรกฎาคม พ.ศ. ๒๕๖๖

(ดร.รุจิรา คงนุ้ย)

ประธานหลักสูตรวิทยาการการจัดการข้อมูล



สำนักวิทยบริการและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีราชมงคลสุวรรณภูมิ

ขอมอบวุฒิบัตรให้ไว้เพื่อแสดงว่า

นาย ภูวดล วรบุตร

ได้เข้าร่วมเสริมสร้างความฉลาดรู้ด้านดิจิทัลในอาเซียน
: ASEAN DIGITAL LITERACY PROGRAMME (ADLP) จำนวน 3 ชม.
ให้ไว้ ณ วันที่ 25 มกราคม พ.ศ. 2566

นายเอกชัย เนาวนิช
ผู้อำนวยการสำนักวิทยบริการและเทคโนโลยีสารสนเทศ



thank you