1. Data Collection

In [261…

```
import pandas as pd
import numpy as np
df = pd.read_csv("clean_feature(1).csv")
df
```

Out[261…

	Day	DoWeek	hour	count	passed	period	data	ServiceID
0	12/29/2021	2	17	2974	2974	63.952589	612.081036	2
1	12/29/2021	2	17	4528	4528	63.000000	960.307862	11
2	12/29/2021	2	17	19	19	62.000000	566.315790	5
3	12/29/2021	2	17	1207	1207	63.000000	601.025684	7
4	12/29/2021	2	18	8450	8450	63.942130	611.024024	2
...	...	...	...	...	...	...	...	...
450666	7/24/2023	0	18	1	1	62.000000	624.000000	1
450667	7/24/2023	0	18	108	108	62.000000	1075.972222	5
450668	7/24/2023	0	18	5279	5279	63.000000	599.900170	7
450669	7/24/2023	0	18	2	2	63.000000	1437.500000	0
450670	7/24/2023	0	18	18	18	0.000000	378255.166700	10

450671 rows × 8 columns

In [261…

```
# Chọn ra những đơn hàng bị lỗi với điều kiện đơn hàng bị lỗi khi count != passed
df = df[df["count"] != df["passed"]]
df
```

Out[261…

	Day	DoWeek	hour	count	passed	period	data	ServiceID
49	12/30/2021	3	2	1	0	60.0	1468.000000	3
287	12/31/2021	4	14	3713	3712	63.0	611.059790	7
321	12/31/2021	4	17	1	0	60.0	1468.000000	3
541	1/2/2022	6	9	1510	1509	63.0	599.986093	7
860	1/4/2022	1	13	1	0	60.0	1468.000000	3
...	...	...	...	...	...	...	...	...
450207	3/28/2023	1	7	1	0	62.0	0.000000	6
450364	4/23/2023	6	17	1	0	60.0	1468.000000	3
450375	4/27/2023	3	17	2	0	60.0	1468.000000	3
450394	5/8/2023	0	17	3	0	0.0	1164.000000	4
450401	5/8/2023	0	18	1	0	0.0	1164.000000	4

8086 rows × 8 columns

In [261…

```
#Xem trong cột ServiceID có bao nhiêu giá trị
#ServiceID_2 không gây ra lỗi trong bộ dữ liệu nên trong xuất hiện trong chuỗi dưới
df["ServiceID"].unique()
```

Out[261…

```
array([ 3,  7,  0,  1,  8, 11,  6,  4,  9, 10,  5])
```

In [261…

```
#Gắn nhãn và phân loại đơn hàng bị lỗi do dịch dịch vụ nào
service_dummies = pd.get_dummies(df['ServiceID'], prefix='ServiceID')
df = pd.concat([df, service_dummies], axis=1)
df = df.drop(['ServiceID'], axis=1)
label_columns = [col for col in df.columns if col.startswith('ServiceID')]
df[label_columns] = df[label_columns].replace([True, False], [1, 0])
df
```

```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_12256\2147077583.py:6: FutureWarning: Do
wncasting behavior in `replace` is deprecated and will be removed in a future versio
n. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. T
o opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting',
True)`
  df[label_columns] = df[label_columns].replace([True, False], [1, 0])
```

Out[261…

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	Serv
49	12/30/2021	3	2	1	0	60.0	1468.000000	0	
287	12/31/2021	4	14	3713	3712	63.0	611.059790	0	
321	12/31/2021	4	17	1	0	60.0	1468.000000	0	
541	1/2/2022	6	9	1510	1509	63.0	599.986093	0	
860	1/4/2022	1	13	1	0	60.0	1468.000000	0	
...	...	...	...	...	...	...	...	...	
450207	3/28/2023	1	7	1	0	62.0	0.000000	0	
450364	4/23/2023	6	17	1	0	60.0	1468.000000	0	
450375	4/27/2023	3	17	2	0	60.0	1468.000000	0	
450394	5/8/2023	0	17	3	0	0.0	1164.000000	0	
450401	5/8/2023	0	18	1	0	0.0	1164.000000	0	

8086 rows × 18 columns

2. Statistics

In [261…

```
# Tên của các cột
df.columns
```

Out[261…

```
Index(['Day', 'DoWeek', 'hour', 'count', 'passed', 'period', 'data',
       'ServiceID_0', 'ServiceID_1', 'ServiceID_3', 'ServiceID_4',
       'ServiceID_5', 'ServiceID_6', 'ServiceID_7', 'ServiceID_8',
       'ServiceID_9', 'ServiceID_10', 'ServiceID_11'],
      dtype='object')
```

In [261…

```
# Ta thấy độ lệch chuẩn của Service_3 và Service_7 khác cao so với các dịch vụ còn
df.describe()
```

Out[261…

	DoWeek	hour	count	passed	period	data
count	8086.000000	8086.000000	8086.000000	8086.000000	8086.000000	8086.000000
mean	2.757853	13.785061	1487.818699	1473.146673	54.601135	992.745548
std	1.978233	5.399448	2195.336593	2179.958109	19.618023	3646.222300
min	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000
25%	1.000000	10.000000	1.000000	0.000000	60.000000	600.379858
50%	3.000000	14.000000	2.000000	0.000000	62.000000	820.500000
75%	4.000000	18.000000	2959.000000	2952.500000	63.000000	1468.000000
max	6.000000	23.000000	17354.000000	16077.000000	63.000000	326030.500000

In [261…

```
label_counts = df[label_columns].sum()
label_counts
```

Out[261…

```
ServiceID_0      150
ServiceID_1      211
ServiceID_3     3010
ServiceID_4      919
ServiceID_5        1
ServiceID_6       64
ServiceID_7     3210
ServiceID_8      509
ServiceID_9        3
ServiceID_10       1
ServiceID_11       8
dtype: int64
```

In [262…

```
#Đa số cột đều có kiểu dữ liệu là int, chỉ riêng cột Day có kiểu string
df.dtypes
```

Out[262…

```
Day             object
DoWeek           int64
hour             int64
count            int64
passed           int64
period         float64
data           float64
ServiceID_0      int64
ServiceID_1      int64
ServiceID_3      int64
ServiceID_4      int64
ServiceID_5      int64
ServiceID_6      int64
ServiceID_7      int64
ServiceID_8      int64
ServiceID_9      int64
ServiceID_10     int64
ServiceID_11     int64
dtype: object
```

In [262…

```
df.head(10)
```

Out[262…

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceI
49	12/30/2021	3	2	1	0	60.0	1468.000000	0	
287	12/31/2021	4	14	3713	3712	63.0	611.059790	0	
321	12/31/2021	4	17	1	0	60.0	1468.000000	0	
541	1/2/2022	6	9	1510	1509	63.0	599.986093	0	
860	1/4/2022	1	13	1	0	60.0	1468.000000	0	
861	1/4/2022	1	13	5	4	63.0	1940.000000	1	
872	1/4/2022	1	14	8	7	63.0	2010.000000	1	
878	1/4/2022	1	15	5	3	63.0	1575.000000	1	
884	1/4/2022	1	15	1	0	63.0	554.000000	1	
889	1/4/2022	1	16	5274	5247	63.0	609.899128	0	

3. Data Preprocessing

3.1 Data Normalization

In [262…

```
#Chuẩn hóa dữ liệu: df["Day"] (string) -> df["Day"] (datetime)
df["Day"] = pd.to_datetime(df['Day'])
```

```
print(df['Day'].dtypes)
df
```

```
datetime64[ns]
```

Out[262…

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID
49	2021-12-30	3	2	1	0	60.0	1468.000000	0	
287	2021-12-31	4	14	3713	3712	63.0	611.059790	0	
321	2021-12-31	4	17	1	0	60.0	1468.000000	0	
541	2022-01-02	6	9	1510	1509	63.0	599.986093	0	
860	2022-01-04	1	13	1	0	60.0	1468.000000	0	
...	...	...	...	...	...	...	...	...	
450207	2023-03-28	1	7	1	0	62.0	0.000000	0	
450364	2023-04-23	6	17	1	0	60.0	1468.000000	0	
450375	2023-04-27	3	17	2	0	60.0	1468.000000	0	
450394	2023-05-08	0	17	3	0	0.0	1164.000000	0	
450401	2023-05-08	0	18	1	0	0.0	1164.000000	0	

8086 rows × 18 columns

3.2 Handle Missing or Invalid Values

In [262…

```
#Check missing datas
#Không có cột nào bị miss data
df.isna().sum()
```

Out[262…

```
Day            0
DoWeek         0
hour           0
count          0
passed         0
period         0
data           0
ServiceID_0    0
ServiceID_1    0
ServiceID_3    0
ServiceID_4    0
ServiceID_5    0
ServiceID_6    0
ServiceID_7    0
ServiceID_8    0
ServiceID_9    0
ServiceID_10   0
ServiceID_11   0
dtype: int64
```

In [262…

```
#Check duplicated datas
df.loc[df.duplicated()]
```

Out[262...

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID_1	S
7029	2022-02-13	6	12	1	0	60.0	1468.0	0	0	
9886	2022-03-03	3	16	1	0	60.0	1468.0	0	0	
13914	2022-03-27	6	15	1	0	60.0	1468.0	0	0	
49817	2022-10-13	3	11	1	0	60.0	1468.0	0	0	
52382	2022-10-26	2	19	1	0	60.0	1468.0	0	0	
...	...	...	...	...	...	...	...	...	...	
449925	2023-03-26	6	18	1	0	60.0	1468.0	0	0	
450207	2023-03-28	1	7	1	0	62.0	0.0	0	0	
450364	2023-04-23	6	17	1	0	60.0	1468.0	0	0	
450394	2023-05-08	0	17	3	0	0.0	1164.0	0	0	
450401	2023-05-08	0	18	1	0	0.0	1164.0	0	0	

925 rows × 18 columns

In [262...

```
#Drop duplicated datas
df= df.loc[~df.duplicated()].reset_index(drop=True).copy()
df
```

Out[262…

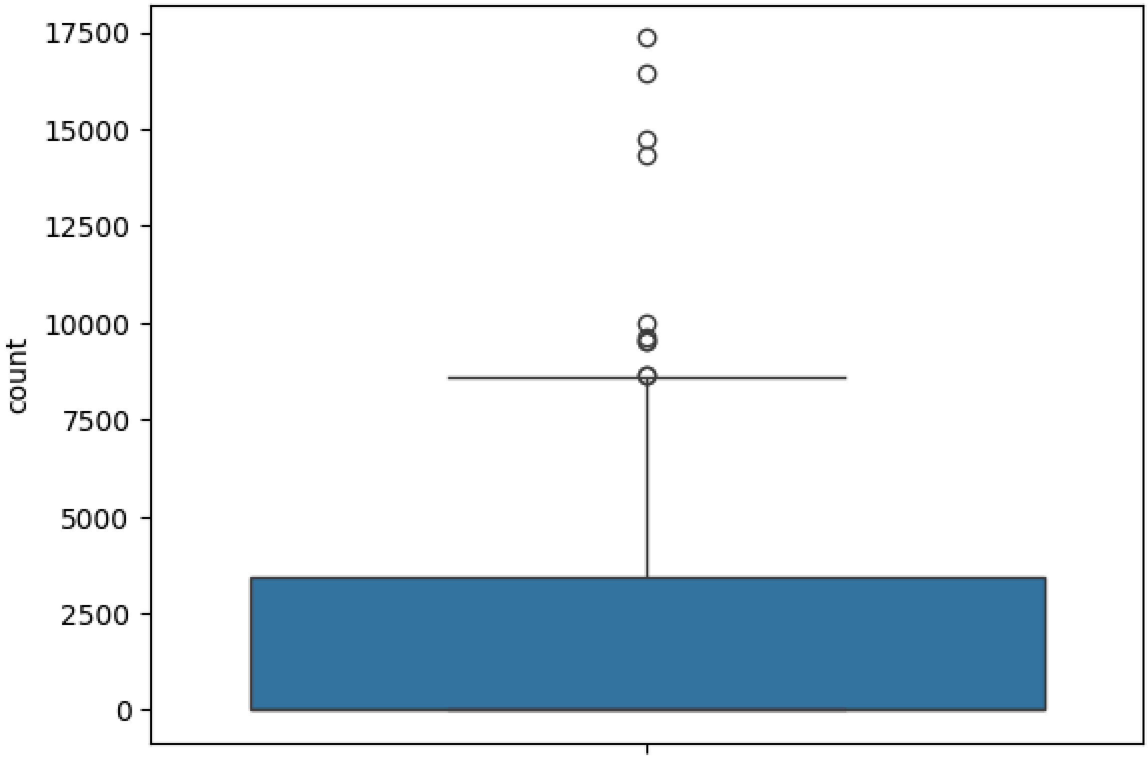
	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID_1
0	2021-12-30	3	2	1	0	60.0	1468.000000	0	0
1	2021-12-31	4	14	3713	3712	63.0	611.059790	0	0
2	2021-12-31	4	17	1	0	60.0	1468.000000	0	0
3	2022-01-02	6	9	1510	1509	63.0	599.986093	0	0
4	2022-01-04	1	13	1	0	60.0	1468.000000	0	0
...	...	...	...	...	...	...	...	...	...
7156	2022-11-22	1	17	8	6	62.0	238.500000	0	0
7157	2023-02-02	3	17	1	0	60.0	1468.000000	0	0
7158	2023-03-25	5	1	1	0	60.0	1468.000000	0	0
7159	2023-03-27	0	18	1	0	60.0	1468.000000	0	0
7160	2023-04-27	3	17	2	0	60.0	1468.000000	0	0

7161 rows × 18 columns

3.3 Remove Outliers

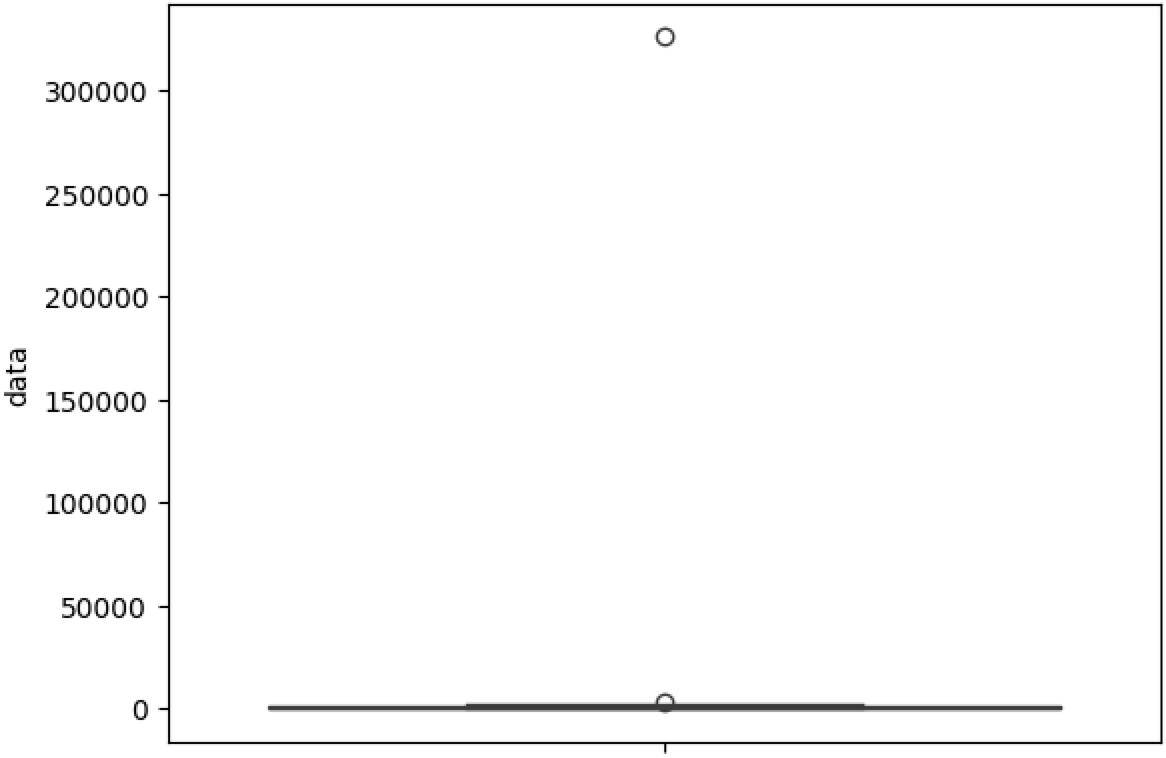
In [262…

```
import numpy as np
import seaborn as sns
sns.boxplot(df["count"])
old_lenght_df = len(df)
```



```
In [262…  sns.boxplot(df["data"])
```

```
Out[262…  <Axes: ylabel='data'>
```



Ban đầu nhóm đã lọc đi các giá trị outlier do count và data nhưng sau khi xem qua mối tương quan giữa chúng thì nhóm quyết định sẽ giữ lại các giá trị outliers

In [262…

```
df
```

Out[262…

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID_1
0	2021-12-30	3	2	1	0	60.0	1468.000000	0	0
1	2021-12-31	4	14	3713	3712	63.0	611.059790	0	0
2	2021-12-31	4	17	1	0	60.0	1468.000000	0	0
3	2022-01-02	6	9	1510	1509	63.0	599.986093	0	0
4	2022-01-04	1	13	1	0	60.0	1468.000000	0	0
...	...	...	...	...	...	...	...	...	...
7156	2022-11-22	1	17	8	6	62.0	238.500000	0	0
7157	2023-02-02	3	17	1	0	60.0	1468.000000	0	0
7158	2023-03-25	5	1	1	0	60.0	1468.000000	0	0
7159	2023-03-27	0	18	1	0	60.0	1468.000000	0	0
7160	2023-04-27	3	17	2	0	60.0	1468.000000	0	0

7161 rows × 18 columns

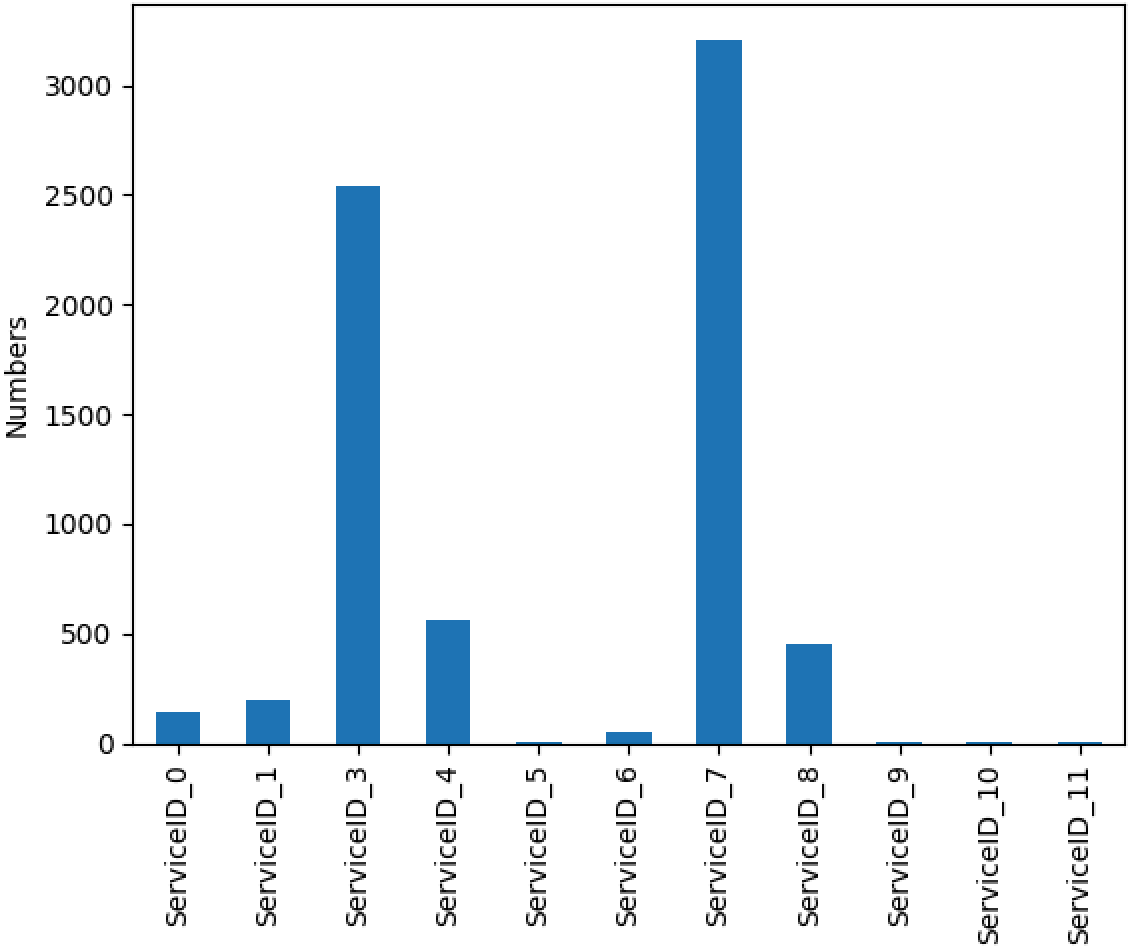
3.4 Balance for Data

In [262…

```python
import matplotlib.pyplot as plt
import seaborn as sns
#Pie chart for class distribution
def Column_chart(y):
    y.sum().plot(kind='bar')
    plt.ylabel('Numbers')
    plt.show()
```

In [263…

```
#Ta thấy bộ dữ liệu toàn dịch vụ 3 và 7, một số dịch vụ khác rất ít -> imbalance da
Column_chart(df[label_columns])
```



In [263…

```
df
```

Out[263…

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID_1
0	2021-12-30	3	2	1	0	60.0	1468.000000	0	0
1	2021-12-31	4	14	3713	3712	63.0	611.059790	0	0
2	2021-12-31	4	17	1	0	60.0	1468.000000	0	0
3	2022-01-02	6	9	1510	1509	63.0	599.986093	0	0
4	2022-01-04	1	13	1	0	60.0	1468.000000	0	0
...	...	...	...	...	...	...	...	...	...
7156	2022-11-22	1	17	8	6	62.0	238.500000	0	0
7157	2023-02-02	3	17	1	0	60.0	1468.000000	0	0
7158	2023-03-25	5	1	1	0	60.0	1468.000000	0	0
7159	2023-03-27	0	18	1	0	60.0	1468.000000	0	0
7160	2023-04-27	3	17	2	0	60.0	1468.000000	0	0

7161 rows × 18 columns

In [263…

```
df[label_columns].sum()
```

Out[263…

```
ServiceID_0      140
ServiceID_1      200
ServiceID_3     2541
ServiceID_4      560
ServiceID_5        1
ServiceID_6       48
ServiceID_7     3210
ServiceID_8      449
ServiceID_9        3
ServiceID_10       1
ServiceID_11       8
dtype: int64
```

In [263…

```
features = df.drop(label_columns, axis=1)
labels = df[label_columns]

def balance_data_undersampling(df, labels):
    # labels là dataframe gồm các cột Outcome
    # Đếm số lượng mẫu của mỗi lớp
    label_counts = labels.sum(axis=0)

    # Đặt ngưỡng mục tiêu để cân bằng cho tất cả các nhãn với giá trị hợp lý
    target_count = min(label_counts.max(), 700)  # Đặt ngưỡng cân bằng là 300 để gi

    # Tạo danh sách các DataFrames đã được xử lý cho từng nhãn (undersampling)
    undersampled_dfs = []

    for col in labels.columns:
        # Lấy tất cả các mẫu có nhãn hiện tại là 1 và thực hiện undersampling nếu c
        positive_samples = df[df[col] == 1]
        if len(positive_samples) > target_count:
            positive_samples = positive_samples.sample(n=target_count, random_state

        # Thêm vào danh sách các DataFrames đã xử lý
        undersampled_dfs.append(positive_samples)

    # Kết hợp tất cả các DataFrames lại với nhau và loại bỏ các bản sao trùng lặp
    final_undersampled_df = pd.concat(undersampled_dfs).drop_duplicates().reset_ind
    return  final_undersampled_df
df = balance_data_undersampling(df, labels)
```

In [263…

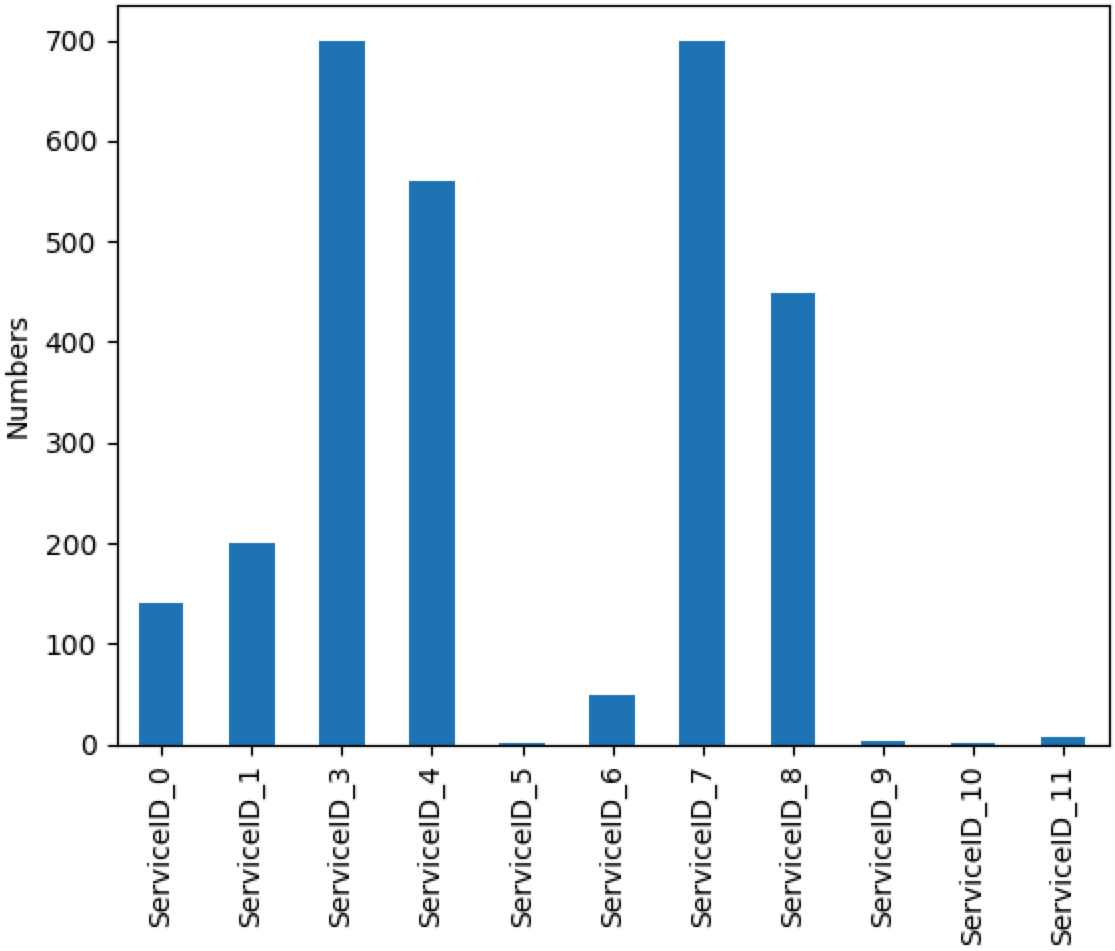
```
#Dữ liệu sau khi được undersampling
df[label_columns].sum()
```

Out[263…

```
ServiceID_0     140
ServiceID_1     200
ServiceID_3     700
ServiceID_4     560
ServiceID_5       1
ServiceID_6      48
ServiceID_7     700
ServiceID_8     449
ServiceID_9       3
ServiceID_10      1
ServiceID_11      8
dtype: int64
```

In [263…

```
Column_chart(df[label_columns])
```

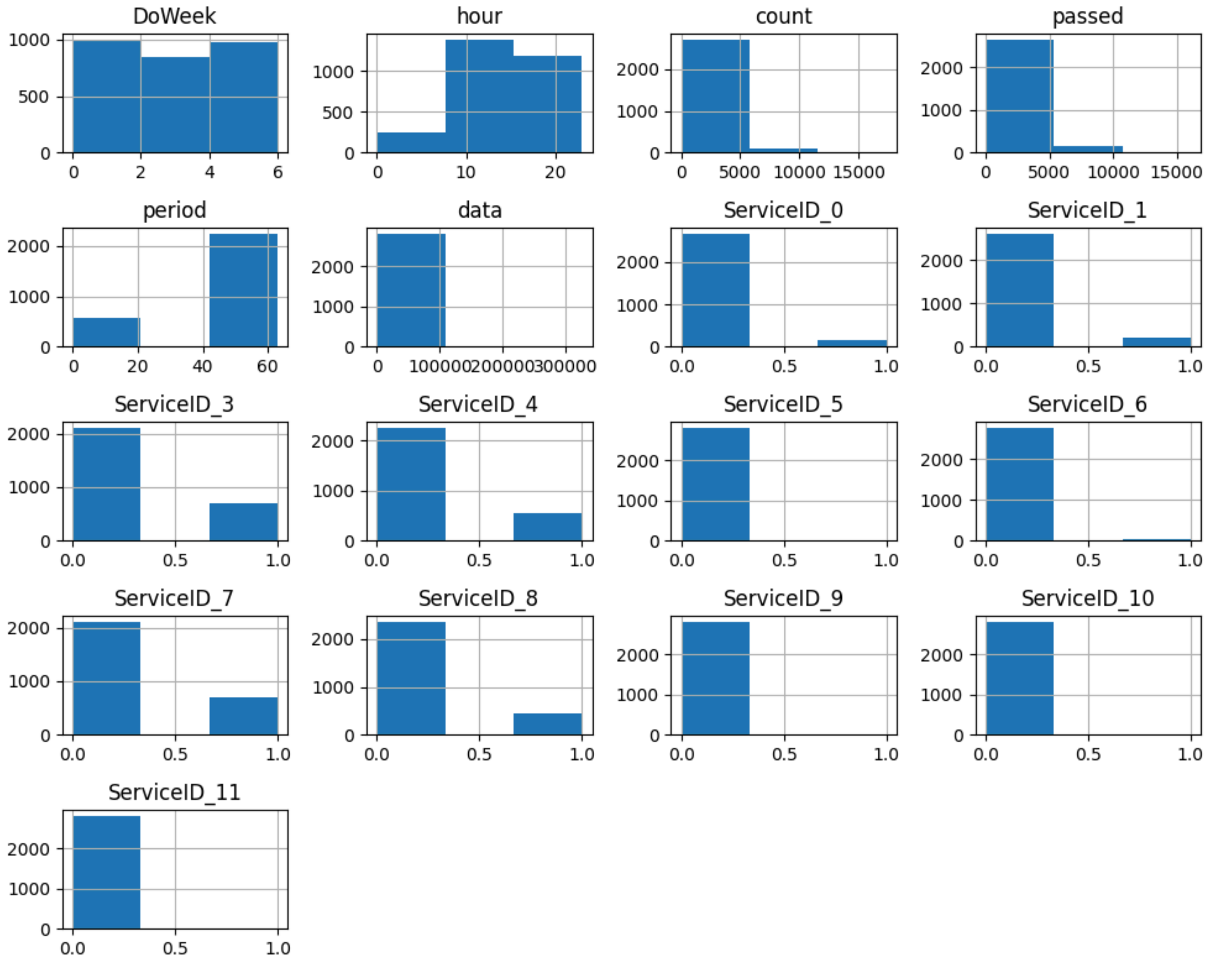


4. Data Visualization

Histogram

In [263…

```
df.drop("Day", axis=1).hist(bins=3, figsize=(10, 8))
plt.tight_layout()
plt.show()
```



Correlation Matrix Plot

In [263…

```
df.corr(method= "spearman")
```

Out[263...

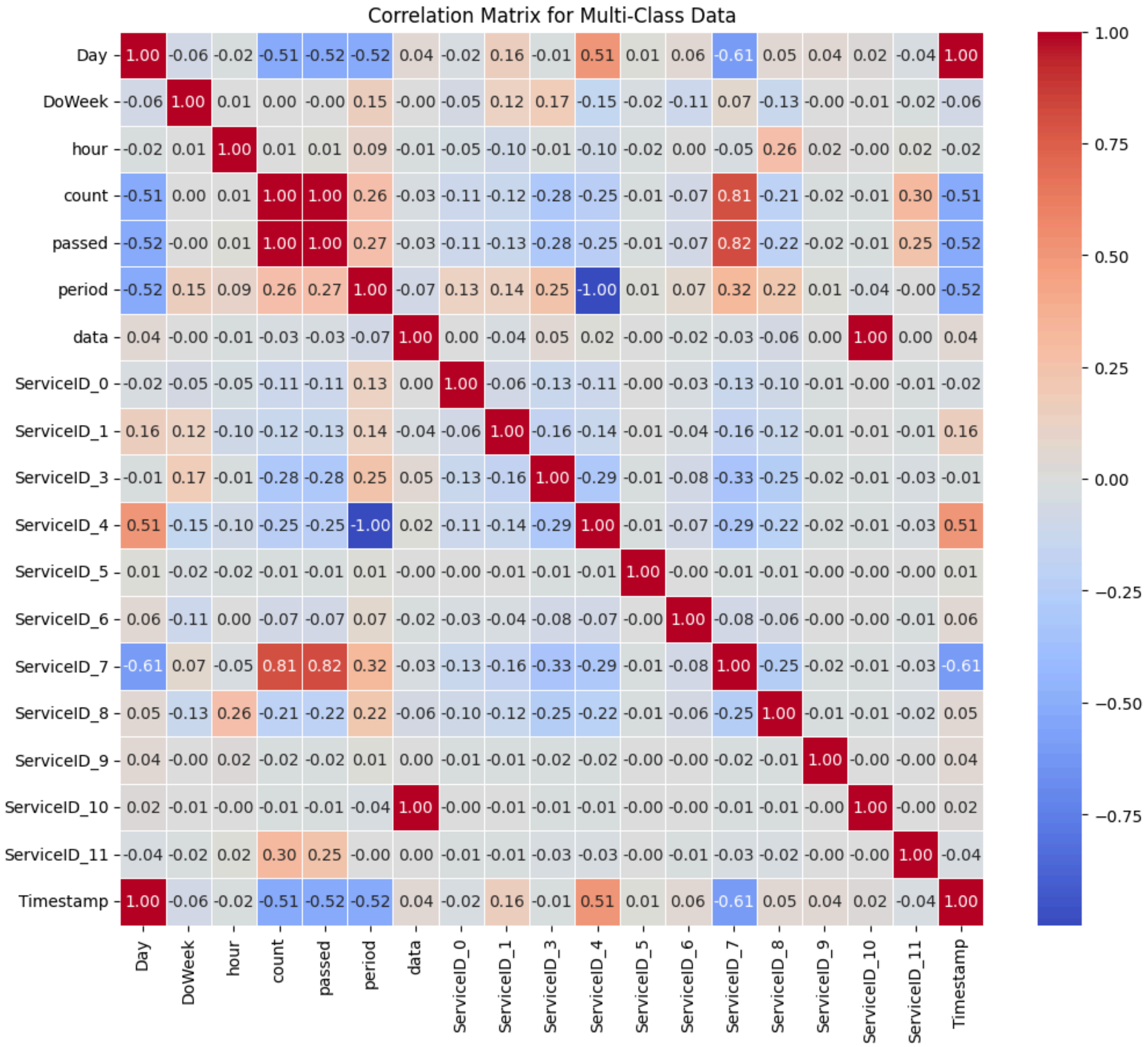
	Day	DoWeek	hour	count	passed	period	data	Se
Day	1.000000	-0.055161	-0.000033	-0.455692	-0.565052	-0.585600	0.069118	
DoWeek	-0.055161	1.000000	0.014367	-0.005846	-0.042123	0.051831	0.040467	
hour	-0.000033	0.014367	1.000000	-0.014637	0.005752	0.056762	-0.131743	
count	-0.455692	-0.005846	-0.014637	1.000000	0.877871	0.771694	-0.540147	
passed	-0.565052	-0.042123	0.005752	0.877871	1.000000	0.807532	-0.385657	
period	-0.585600	0.051831	0.056762	0.771694	0.807532	1.000000	-0.522451	
data	0.069118	0.040467	-0.131743	-0.540147	-0.385657	-0.522451	1.000000	
ServiceID_0	-0.040633	-0.051573	-0.067354	-0.019098	0.045584	0.289629	0.072030	
ServiceID_1	0.152765	0.093991	-0.093571	0.196433	-0.076060	0.068089	-0.365327	
ServiceID_3	-0.002660	0.168955	0.016934	-0.563856	-0.464987	-0.353826	0.734347	
ServiceID_4	0.470496	-0.142829	-0.135470	-0.348479	-0.402750	-0.714243	0.214303	
ServiceID_5	0.007304	-0.026645	-0.025248	0.013961	0.016998	-0.002871	-0.020722	
ServiceID_6	0.035594	-0.107312	-0.005332	-0.048838	-0.056155	0.037428	-0.186179	
ServiceID_7	-0.597485	0.066502	-0.036810	0.772096	0.827615	0.720718	-0.245716	
ServiceID_8	0.097713	-0.116091	0.276251	0.010952	0.042650	0.123812	-0.534632	
ServiceID_9	0.039605	-0.001693	0.021056	-0.013725	-0.013057	-0.015171	0.028610	
ServiceID_10	0.029669	-0.004272	-0.006082	0.010552	0.015985	-0.027013	0.033091	
ServiceID_11	-0.023056	-0.012098	0.014996	0.095575	0.096482	-0.057144	0.003211	

In [263...

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Tính ma trận tương quan cho DataFrame đã cân bằng
correlation_matrix = final_undersampled_df.corr()

# Vẽ heatmap cho ma trận tương quan
plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5, fmt=".
plt.title("Correlation Matrix for Multi-Class Data")
plt.show()
```



Tương quan thấp (cả âm và dương) giữa các nhãn dịch vụ -> dịch vụ này khá độc lập

Ta thấy hệ số tương quan giữa count và ServiceID7 rất cao 0.81 -> dịch vụ 7 thường có lỗi khi có count lớn

5. Model Building

Các dịch khá độc lập -> chon RandomForest để thực hiện dự án này

In [263…

```
df['Timestamp'] = df.apply(lambda row: row['Day'] + pd.Timedelta(hours=row['hour'])
df = df.sort_values(by='Timestamp')
df
```

Out[263...

	Day	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID_1
602	2021-12-29	2	21	1	0	60.0	1468.000000	0	0
1802	2021-12-30	3	11	4036	4035	63.0	605.483895	0	0
1911	2021-12-31	4	14	3713	3712	63.0	611.059790	0	0
569	2021-12-31	4	14	1	0	60.0	1468.000000	0	0
2323	2021-12-31	4	14	4118	4116	63.0	610.565809	0	0
...	...	...	...	...	...	...	...	...	...
2575	2023-08-02	2	21	2	1	62.0	280.500000	0	0
2796	2023-08-02	2	21	3	1	62.0	106.000000	0	0
2693	2023-08-02	2	21	4	0	62.0	0.000000	0	0
905	2023-08-02	2	22	1	0	60.0	1468.000000	0	0
2576	2023-08-04	4	6	3	2	62.0	374.000000	0	0

2810 rows × 19 columns

In [264...

```
df = df.drop("Day", axis=1)
df
```

Out[264...

	DoWeek	hour	count	passed	period	data	ServiceID_0	ServiceID_1	Servi
602	2	21	1	0	60.0	1468.000000	0	0	
1802	3	11	4036	4035	63.0	605.483895	0	0	
1911	4	14	3713	3712	63.0	611.059790	0	0	
569	4	14	1	0	60.0	1468.000000	0	0	
2323	4	14	4118	4116	63.0	610.565809	0	0	
...	...	...	...	...	...	...	...	...	
2575	2	21	2	1	62.0	280.500000	0	0	
2796	2	21	3	1	62.0	106.000000	0	0	
2693	2	21	4	0	62.0	0.000000	0	0	
905	2	22	1	0	60.0	1468.000000	0	0	
2576	4	6	3	2	62.0	374.000000	0	0	

2810 rows × 18 columns

In [264...

```
# Đưa cột cuối lên đầu
def change_col_position(df):
    cols = df.columns.tolist()
    new_order = [cols[-1]] + cols[:-1]
    df = df[new_order]
    return df
```

In [264...

```
df = change_col_position(df)
df
```

```
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\IPython\cor
e\displayhook.py:281: UserWarning: Output cache limit (currently 1000 entries) hit.
Flushing oldest 200 entries.
  warn('Output cache limit (currently {sz} entries) hit.\n'
```

Out[264…

	Timestamp	DoWeek	hour	count	passed	period	data	ServiceID_0	Servic
602	2021-12-29 21:00:00	2	21	1	0	60.0	1468.000000	0	
1802	2021-12-30 11:00:00	3	11	4036	4035	63.0	605.483895	0	
1911	2021-12-31 14:00:00	4	14	3713	3712	63.0	611.059790	0	
569	2021-12-31 14:00:00	4	14	1	0	60.0	1468.000000	0	
2323	2021-12-31 14:00:00	4	14	4118	4116	63.0	610.565809	0	
...	...	...	...	...	...	...	...	...	
2575	2023-08-02 21:00:00	2	21	2	1	62.0	280.500000	0	
2796	2023-08-02 21:00:00	2	21	3	1	62.0	106.000000	0	
2693	2023-08-02 21:00:00	2	21	4	0	62.0	0.000000	0	
905	2023-08-02 22:00:00	2	22	1	0	60.0	1468.000000	0	
2576	2023-08-04 06:00:00	4	6	3	2	62.0	374.000000	0	

2810 rows × 18 columns

In [264…

```
def enter_time():
    while True:
        start_time = input("Nhập giá trị timestamp (định dạng: yyyy-mm-dd hh:mm:ss)
        try:
            start_time = pd.to_datetime(start_time)
            break
        except ValueError:
            print("Vui lòng nhập chính xác theo định dạng yêu cầu")
    return start_time
```

In [ ]:

```
features = df.drop(label_columns, axis=1)
labels = df[label_columns]

start_time = enter_time()
delta_time = int(input("Nhập khoảng thời gian cần dự đoán: "))
train_set = df[df["Timestamp"] < start_time]
test_set = df[(df["Timestamp"] >= start_time) & (df["Timestamp"] <= (start_time + p

x_train = train_set.drop(label_columns, axis=1)
x_train = x_train.drop("Timestamp", axis=1)
y_train = train_set[label_columns]

x_test = test_set.drop(label_columns, axis= 1)
x_test =  x_test.drop("Timestamp", axis= 1)
y_test = test_set[label_columns]
```

In [264…

```
x_test
```

Out[264…

	DoWeek	hour	count	passed	period	data
2164	0	19	7865	7374	63.000000	595.380292
2335	0	20	318	0	62.622642	722.553459
2304	0	21	1515	1510	63.000000	599.691749
1934	0	21	5232	5225	63.000000	599.780390
1023	1	7	1	0	60.000000	1468.000000
891	1	8	1	0	60.000000	1468.000000
1601	1	14	1	0	62.000000	0.000000
1616	1	15	2	0	62.000000	0.000000
1602	1	15	4	2	62.000000	283.000000
1641	1	15	3	1	62.000000	201.333333
2077	2	9	6887	6861	63.000000	603.176129
20	2	14	3	2	63.000000	1265.000000
96	2	14	3	2	63.000000	1641.666667
1016	2	19	1	0	60.000000	1468.000000

In [264…

```
y_test
```

Out[264…

	ServiceID_0	ServiceID_1	ServiceID_3	ServiceID_4	ServiceID_5	ServiceID_6	ServiceI
2164	0	0	0	0	0	0	
2335	0	0	0	0	0	0	
2304	0	0	0	0	0	0	
1934	0	0	0	0	0	0	
1023	0	0	1	0	0	0	
891	0	0	1	0	0	0	
1601	0	0	0	0	0	1	
1616	0	0	0	0	0	1	
1602	0	0	0	0	0	1	
1641	0	0	0	0	0	1	
2077	0	0	0	0	0	0	
20	1	0	0	0	0	0	
96	1	0	0	0	0	0	
1016	0	0	1	0	0	0	

In [264…

```
from sklearn.multioutput import MultiOutputClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Giả sử bạn đã chuẩn bị dữ liệu X (features) và y (multi-label targets)
# Chia dữ liệu thành tập huấn luyện và tập kiểm tra

# Sử dụng MultiOutputClassifier với Random Forest
model = MultiOutputClassifier(RandomForestClassifier(random_state=42))
model.fit(x_train, y_train)



# Dự đoán và đánh giá mô hình
y_pred = classifier.predict(x_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.7142857142857143
```

Lưu lại mô hình trước khi điều chỉnh tham số

In [264…

```
import joblib
joblib.dump(model, 'model.pkl')
```

Out[264…

```
['model.pkl']
```

In [264…

```python
# Do không có ServiceID_2 nên cột idex bị lệch
# Chuẩn hóa: 0 -> Ser_0, 1 -> Ser_1,       2 -> Ser_3, 3 -> Ser_4 ....      10 -> s
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       0.00      0.00      0.00         0
           2       1.00      1.00      1.00         3
           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00         0
           5       1.00      0.25      0.40         4
           6       1.00      0.80      0.89         5
           7       0.00      0.00      0.00         0
           8       0.00      0.00      0.00         0
           9       0.00      0.00      0.00         0
          10       0.00      0.00      0.00         0

   micro avg       0.83      0.71      0.77        14
   macro avg       0.36      0.28      0.30        14
weighted avg       1.00      0.71      0.79        14
 samples avg       0.71      0.71      0.71        14
```

```
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: Recall is ill-defined and bein
g set to 0.0 in labels with no true samples. Use `zero_division` parameter to contro
l this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: F-score is ill-defined and bei
ng set to 0.0 in labels with no true nor predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in samples with no predicted labels. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

In [265…

```python
# [ser_0, ser_1, ser_3, ser_4, ser_5, ser_6, ser_7, ser_8, ser_9, ser_10, ser_11]
for i,j in list(zip(y_pred, y_test.values)):
    print("Predict: {}. Label: {}".format(i, j))
```

```
Predict: [0 0 0 0 0 0 1 0 0 0 0]. Label: [0 0 0 0 0 0 1 0 0 0 0]
Predict: [0 0 0 0 0 0 0 0 0 0 0]. Label: [0 0 0 0 0 0 1 0 0 0 0]
Predict: [0 0 0 0 0 0 1 0 0 0 0]. Label: [0 0 0 0 0 0 1 0 0 0 0]
Predict: [0 0 0 0 0 0 1 0 0 0 0]. Label: [0 0 0 0 0 0 1 0 0 0 0]
Predict: [0 0 1 0 0 0 0 0 0 0 0]. Label: [0 0 1 0 0 0 0 0 0 0 0]
Predict: [0 0 1 0 0 0 0 0 0 0 0]. Label: [0 0 1 0 0 0 0 0 0 0 0]
Predict: [0 0 0 0 0 0 0 0 0 0 0]. Label: [0 0 0 0 0 1 0 0 0 0 0]
Predict: [0 0 0 0 0 1 0 0 0 0 0]. Label: [0 0 0 0 0 1 0 0 0 0 0]
Predict: [0 0 0 0 0 0 0 1 0 0 0]. Label: [0 0 0 0 0 1 0 0 0 0 0]
Predict: [0 0 0 0 0 0 0 1 0 0 0]. Label: [0 0 0 0 0 1 0 0 0 0 0]
Predict: [0 0 0 0 0 0 1 0 0 0 0]. Label: [0 0 0 0 0 0 1 0 0 0 0]
Predict: [1 0 0 0 0 0 0 0 0 0 0]. Label: [1 0 0 0 0 0 0 0 0 0 0]
Predict: [1 0 0 0 0 0 0 0 0 0 0]. Label: [1 0 0 0 0 0 0 0 0 0 0]
Predict: [0 0 1 0 0 0 0 0 0 0 0]. Label: [0 0 1 0 0 0 0 0 0 0 0]
```

In [265…

```
x_test
```

Out[265…

	DoWeek	hour	count	passed	period	data
2164	0	19	7865	7374	63.000000	595.380292
2335	0	20	318	0	62.622642	722.553459
2304	0	21	1515	1510	63.000000	599.691749
1934	0	21	5232	5225	63.000000	599.780390
1023	1	7	1	0	60.000000	1468.000000
891	1	8	1	0	60.000000	1468.000000
1601	1	14	1	0	62.000000	0.000000
1616	1	15	2	0	62.000000	0.000000
1602	1	15	4	2	62.000000	283.000000
1641	1	15	3	1	62.000000	201.333333
2077	2	9	6887	6861	63.000000	603.176129
20	2	14	3	2	63.000000	1265.000000
96	2	14	3	2	63.000000	1641.666667
1016	2	19	1	0	60.000000	1468.000000

6. Model Evaluation

average = 0.7142857142857143, F1 ở dịch vụ 7 và 3 khá cao -> dự đoán dịch vụ 7, 3 khá chính xác

F1 ở dịch vụ 6 thấp -> dự đoán dịch vụ 6 chưa chính xác lắm

7. Parameter Adjustment

In [265…

```
from sklearn.datasets import make_multilabel_classification
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.multioutput import MultiOutputClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score


# Sử dụng MultiOutputClassifier để mở rộng mô hình RandomForest cho multi-label

# Định nghĩa các giá trị tham số cần tìm kiếm với GridSearchCV
param_grid = {
    'estimator__n_estimators': [100, 150, 200],    # Số lượng cây trong RandomFores
    'estimator__min_samples_split': [1, 2],
    'estimator__min_samples_leaf': [1, 4, 6],
}

# Khởi tạo GridSearchCV
grid_search = GridSearchCV(model, param_grid, cv=3, verbose=1, n_jobs=-1)

# Huấn luyện mô hình sử dụng GridSearchCV
grid_search.fit(x_train, y_train)

# In ra tham số tốt nhất
print("Best Parameters:", grid_search.best_params_)

# Dự đoán và đánh giá trên tập kiểm tra
y_pred = grid_search.best_estimator_.predict(x_test)
```

```
Fitting 3 folds for each of 18 candidates, totalling 54 fits
```

```
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\mod
el_selection\_validation.py:540: FitFailedWarning: 
27 fits failed out of a total of 54.
The score on these train-test partitions for these parameters will be set to nan.
If these failures are not expected, you can try to debug them by setting error_score
='raise'.

Below are more details about the failures:
--------------------------------------------------------------------------------
27 fits failed with the following error:
Traceback (most recent call last):
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\model_selection\_validation.py", line 888, in _fit_and_score
    estimator.fit(X_train, y_train, **fit_params)
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\multioutput.py", line 543, in fit
    super().fit(X, Y, sample_weight=sample_weight, **fit_params)
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\base.py", line 1473, in wrapper
    return fit_method(estimator, *args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\multioutput.py", line 278, in fit
    self.estimators_ = Parallel(n_jobs=self.n_jobs)(
                       ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\utils\parallel.py", line 74, in __call__
    return super().__call__(iterable_with_config)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\job
lib\parallel.py", line 1918, in __call__
    return output if self.return_generator else list(output)
                                                ^^^^^^^^^^^^
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\job
lib\parallel.py", line 1847, in _get_sequential_output
    res = func(*args, **kwargs)
          ^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\utils\parallel.py", line 136, in __call__
    return self.function(*args, **kwargs)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\multioutput.py", line 67, in _fit_estimator
    estimator.fit(X, y, **fit_params)
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\base.py", line 1466, in wrapper
    estimator._validate_params()
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\base.py", line 666, in _validate_params
    validate_parameter_constraints(
  File "c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\skl
earn\utils\_param_validation.py", line 95, in validate_parameter_constraints
    raise InvalidParameterError(
sklearn.utils._param_validation.InvalidParameterError: The 'min_samples_split' param
eter of RandomForestClassifier must be an int in the range [2, inf) or a float in th
e range (0.0, 1.0]. Got 1 instead.
  warnings.warn(some_fits_failed_message, FitFailedWarning)
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\mod
el_selection\_search.py:1103: UserWarning: One or more of the test scores are non-fi
nite: [       nan        nan        nan 0.97111111 0.97185185 0.97185185
        nan        nan        nan 0.96666667 0.96666667 0.96740741
        nan        nan        nan 0.96592593 0.96444444 0.96444444]
  warnings.warn(
Best Parameters: {'estimator__min_samples_leaf': 1, 'estimator__min_samples_split':
2, 'estimator__n_estimators': 150}
```

In [265…

```
# Report ACC, Recall, F1
# Do không có ServiceID_2 nên cột idex bị lệch
# Chuẩn hóa: 0 -> Ser_0, 1 -> Ser_1,       2 -> Ser_3, 3 -> Ser_4 ....      10 -> s
print("Accuracy on Test Set:", accuracy_score(y_test, y_pred))
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
Accuracy on Test Set: 0.7142857142857143
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       0.00      0.00      0.00         0
           2       1.00      1.00      1.00         3
           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00         0
           5       1.00      0.25      0.40         4
           6       1.00      0.80      0.89         5
           7       0.00      0.00      0.00         0
           8       0.00      0.00      0.00         0
           9       0.00      0.00      0.00         0
          10       0.00      0.00      0.00         0

   micro avg       0.77      0.71      0.74        14
   macro avg       0.36      0.28      0.30        14
weighted avg       1.00      0.71      0.79        14
 samples avg       0.71      0.71      0.71        14
```

```
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in labels with no predicted samples. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: Recall is ill-defined and bein
g set to 0.0 in labels with no true samples. Use `zero_division` parameter to contro
l this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: F-score is ill-defined and bei
ng set to 0.0 in labels with no true nor predicted samples. Use `zero_division` para
meter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
c:\Users\ADMIN\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\met
rics\_classification.py:1531: UndefinedMetricWarning: Precision is ill-defined and b
eing set to 0.0 in samples with no predicted labels. Use `zero_division` parameter t
o control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is", len(result))
```

Lưu lại model đã điều chỉnh tham số

In [265…

```python
# Sau khi tìm kiếm với GridSearchCV
best_model = grid_search.best_estimator_

# Lưu lại mô hình tốt nhất
joblib.dump(best_model, 'best_model.pkl')
```

Out[265…

```
['best_model.pkl']
```

-> Thay đổi các parameter để tăng accuracy dường như KHÔNG ĐÁNG KỂ, các parameter mặc định cũng gần như là tối ưu nhất rồi -> Thay đổi model sau