คำอธิบาย Lab1.2

```
✓ Import tools
[ ] from google.colab import files uploaded = files.upload()
Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable. Saving Credit-Card-Defaulter-Prediction.csv to Credit-Card-Defaulter-Prediction.csv
• import numpy as np import pandas as pd import seaborn as sns from sklearn import preprocessing from sklearn import preprocessing from sklearn.feature_selection import chi2
```

- Import library ที่จะใช้ในการเขียนโปรแกรม

- Import csv file ที่จะใช้ในการเขียนโปรแกรม
- Print df ออกมา

```
Remove column ID.

Hints

[] ### START CODE HERE ###

df.drop('ID', axis='columns', inplace=True)

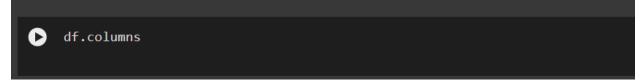
df

### END CODE HERE ###
```

- ลบ column 'ID'

```
df.info()
  df.describe()
  # df.median()
```

- แสดงข้อมูล datatype ของแต่ละ column ใน df
- แสดงค่าที่คำนวณออกมาได้จาก column ทั้งหมดของ df เช่น count, mean, std, min, 25% 50%, 75%, max



- แสดง index ชื่อ column ทั้งหมด

```
### START CODE HERE ###

df = df.fillna(df.median())
df

### END CODE HERE ###
```

- Fill ข้อมูลที่เป็น N/A ด้วย median เนื่องจากค่า max ดูโดดจากข้อมูลอื่นเกินไปเกินไปจึงอาจจะทำให้ ค่า mean เพี้ยนไปได้

```
df.info()
```

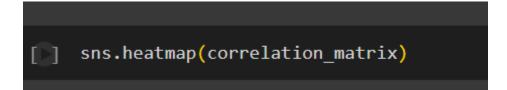
- แสดงข้อมูล datatype ของแต่ละ column ใน df

```
### START CODE HERE ###

correlation_matrix = df.corr()
correlation_matrix

### END CODE HERE ###
```

- คำนวณค่า correlation ของข้อมูลทั้งหมดและใส่ไว้ในตัวแปร correlation_matrix
- Print correlation matrix



- สร้าง heatmap จากค่า correlation

```
[ ] ### START CODE HERE ###
  lower = np.tril(correlation_matrix,-1)
  ### END CODE HERE ###

> Expected output

[ ] sns.heatmap(lower)
```

- แสดงค่าเฉพาะ Lower triangle ของ heatmap

```
### START CODE HERE ###

dfCopy = df.copy()
lower = correlation_matrix.where(np.tril(np.ones(correlation_matrix.shape), k=-1).astype(bool))
to_drop = [column for column in lower.columns if any(lower[column] > 0.6)]
dfCopy.drop(to_drop, axis=1, inplace=True)
dfCopy
### END CODE HERE ###
```

- สร้างตัวแปร dfCopy ขึ้นมาเพื่อคัดลอก df

- ให้ drop column ที่มีค่า correlation มากกว่า 0.6 เพราะข้อมูลมีความเหมือนกันมาก

► Expected output ► dfCopy.describe()

- แสดงค่าที่คำนวณออกมาได้จาก column ทั้งหมดของ dfCopy เช่น count, mean, std, min, 25% 50%, 75%, max

```
### START CODE HERE ###

# non_numeric_columns = df['SEX', 'EDUCATION', 'MARRIAGE','default'].unique()

# non_numeric_columns = np.unique(df[['SEX', 'EDUCATION', 'MARRIAGE']].values)
non_numeric_columns = df.select_dtypes(exclude=['number'])
non_numeric_columns
# df[non_numeric_columns.columns].unique()

### END CODE HERE ###
```

- สร้างตัวแปร non_numeric_columns ขึ้นมาเพื่อสร้างตารางใหม่ที่มีเฉพาะ columns ที่ข้อมูลที่ไม่ใช้ ตัวเลข(ตัวอักษร)

```
### START CODE HERE ###

label_encode = preprocessing.LabelEncoder()
df_encode = non_numeric_columns.apply(label_encode.fit_transform)

df_encode = df_encode.astype('int64')
# df_encode.info()
dfCopy.update(df_encode.astype('int64'))
# dfCopy
### END CODE HERE ###
```

- แปลงข้อมูลที่เป็นตัวอักษรให้กลายเป็นตัวเลข

```
### START CODE HERE ###
output = df_encode['default ']
variable = df_encode[['SEX', 'EDUCATION', 'MARRIAGE']]

chi2_stat, p_values = chi2(variable,output)

chi2_table = pd.DataFrame({
    "Feature": variable.columns,
    "Chi2 Statistic": chi2_stat,
    "P-Value": p_values
})

### END CODE HERE ###
chi2_table
```

- คำนวณค่า chi-Square และ P-value ของ column 'SEX', 'EDUCATION', 'MARRIAGE' ว่าส่งผลต่อ 'default ' มากแค่ไหน

```
[ ] ### START CODE HERE ###
    to_drop = chi2_table[chi2_table['P-Value'] > 0.05]
    print("to_drop :",to_drop.Feature)
    dfCopy.drop(to_drop.Feature, axis=1, inplace=True)
    ### END CODE HERE ###
```

- Drop column ที่มีค่า P-value มากกว่า 0.05 คือ 'MARRIAGE' เนื่องจากไม่มีผลต่อ 'default '



- แสดงตารางข้อมูลล่าสุดที่ทำการ clean เรียบร้อยแล้ว