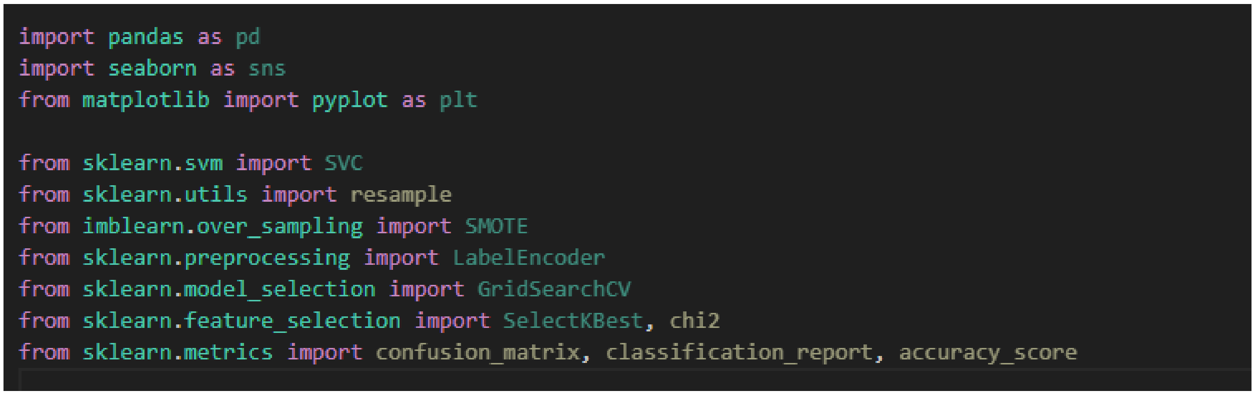
Support Vector Machine Algorithm (SVM)-Part2

1. Import necessary libraries (you should be able to install any libraries that have not been installed in your environment so far)



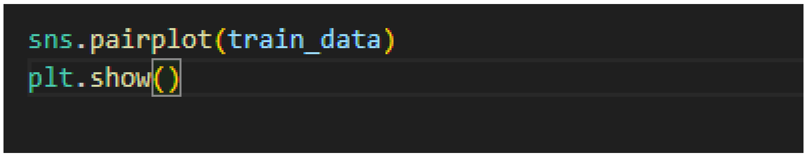
```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt

from sklearn.svm import SVC
from sklearn.utils import resample
from imblearn.over_sampling import SMOTE
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import GridSearchCV
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

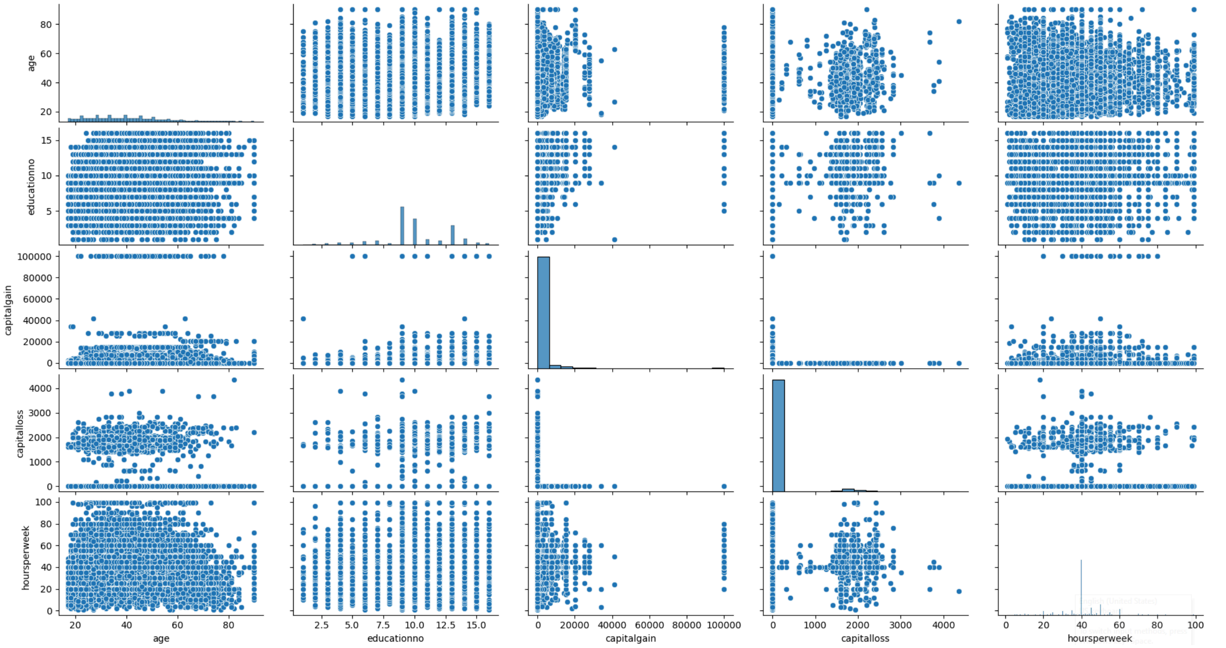
2. Load Train and Test Dataset, and print out and screenshot the first 10 data records from the test data. (see the train and test data files in the data folder)

3. Data understanding (use dataframe attributes, methods, and other python methods to investigate the dataset. Answer the below questions)
    a. What is the dataset size (train? And test?)
    b. Give the data description of the below column with the the column meaning and the value range for the train data (hint: try the pandas describe() and info())
    Examples:
        - age: age of a person: 17 to 90

4. Data visualization
   Example 1: scatter plot each two columns of the train data



```
sns.pairplot(train_data)
plt.show()
```

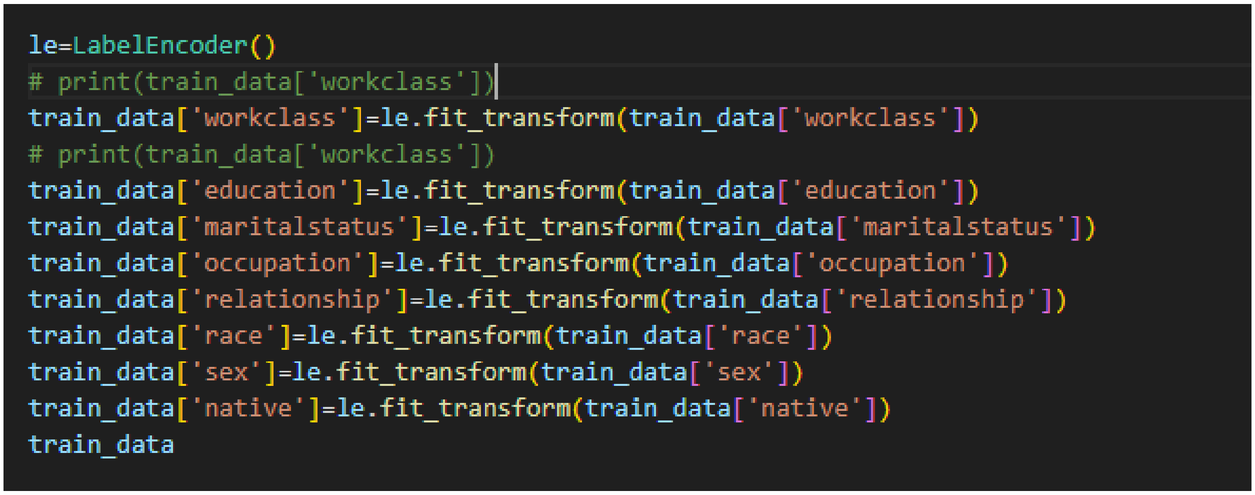


4.1. Take example 1, explore the bar plot and plot the Count .vs. Sex (code and figure)

4.2. explore the bar plot and plot the Count .vs. Salary (salary <=50k, salary>50k) (code and figure

4.3. explore the bar plot and plot the Count .vs. WorkClass for each category of salary in 4.2. (code and figure)

5. Data preprocessing



```
le=LabelEncoder()
# print(train_data['workclass'])
train_data['workclass']=le.fit_transform(train_data['workclass'])
# print(train_data['workclass'])
train_data['education']=le.fit_transform(train_data['education'])
train_data['maritalstatus']=le.fit_transform(train_data['maritalstatus'])
train_data['occupation']=le.fit_transform(train_data['occupation'])
train_data['relationship']=le.fit_transform(train_data['relationship'])
train_data['race']=le.fit_transform(train_data['race'])
train_data['sex']=le.fit_transform(train_data['sex'])
train_data['native']=le.fit_transform(train_data['native'])
train_data
```

Explain what the above codes do

Reset the value of Salary column: if salary <50k, set the salary value to 0;
Else set the salary value to 1;

6. Model Building

Prepare the X_train, y_trian, X_test, and the y_test, and print out the data shape for each of them

7. Model Training | Testing | Evaluation - SVM Model

Select two columns from the train data that you think they can affect the salary most (It doesn't matter which two columns you select, so don't worry about the right answer), then Train SVM models use different kernels by using these two columns, plot the figure of each model's boundary for the test data.
You can take reference from here:
https://scikit-learn.org/stable/auto_examples/exercises/plot_iris_exercise.html#sphx-glr-auto-examples-exercises-plot-iris-exercise-py

Give the testing accuracy for each SVM model on the test data.