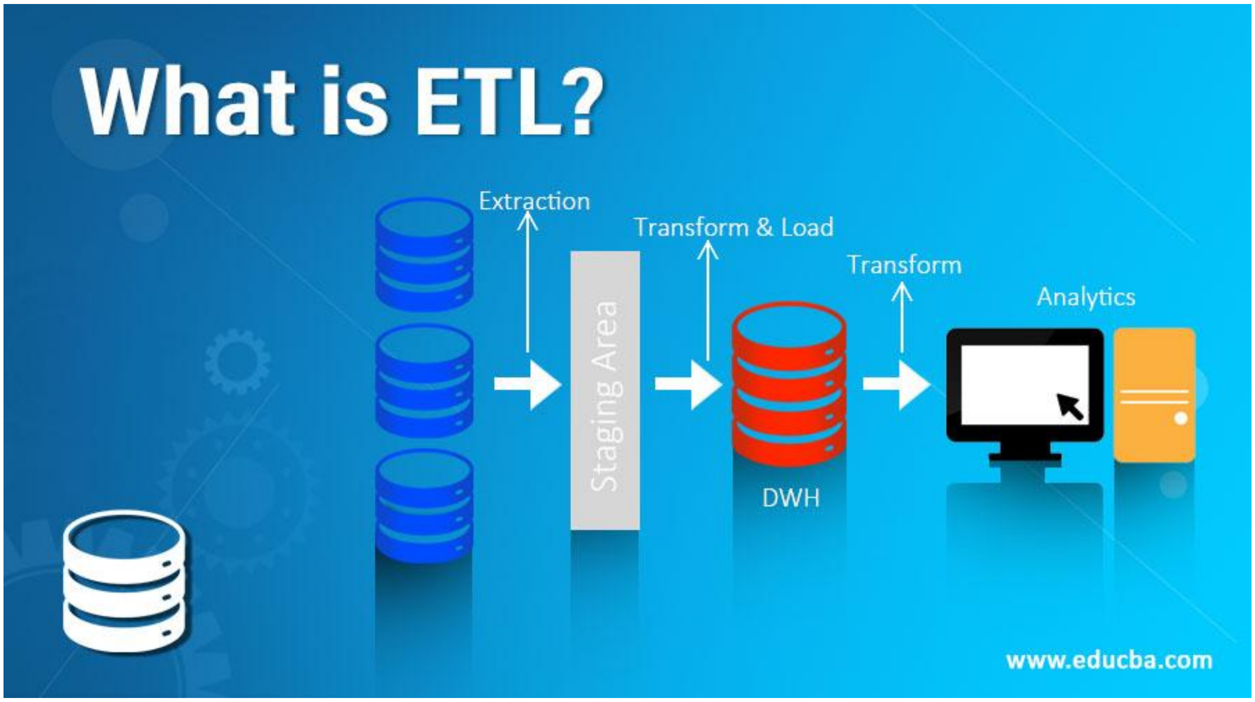
NAME

# SplitColumn Assignment

Using Python Language



## Introduction

We have a data file in CSV format containing 1000 rows and 5 columns in CSV file, we need to filter the file depending on some Terms,

Requirement:

1- Ignore empty lines.

2-Replace empty values with "NA"

3-Duplicate rows should be ignored.

4- The column number used to split is an input from the user, preferred to be a command line the argument, The new filenames should be distinguished by the column values.

## Library Used:

1-pandas

pandas is a software library written for the Python programming language for data manipulation and analysis.

- We Used Pandas to Access & Manipulation Excel file such Replance values / access row /access column / drop / delete duplicate

2-numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices

- We used Numpy to access an empty value and replace it with NA,

3-os

The OS module in Python provides functions for interacting with the operating system

- We used OS Module to access and validate the existing  file and remove them to prevent overwriting

4-argparse

The *argparse module* makes it easy to write user-friendly command-line interfaces.

- We used it to apply a command-line argument

5-glob

The glob module is a useful part of the Python standard library. glob (short for global) is used to return all file paths that match a specific pattern

- We used it to return all files with specific pattern

# For a standard install

-pip install pandas

-pip install numpy

### How to run this script:

We applied the concept of static function to build a code to have better performance and save money

- python main.py -column 3

**Github Link:https://github.com/mohammadfaidi/SplitByColumn-**

## LifeCycle  of this script:

I separate files to show the result of each process.

Datafile.csv  (read Data)===ignore empty lines====>(write to ) req1.csv

req1.csv  (read Data)===replace with NA====>(write to ) req2.csv

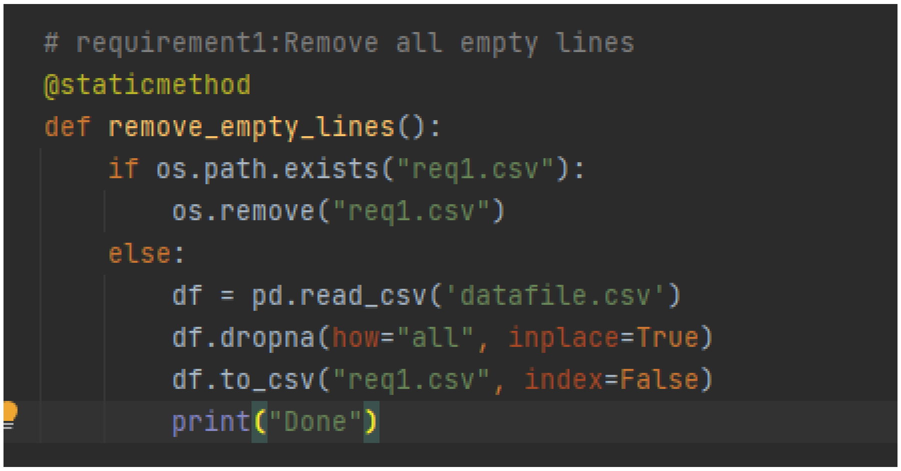
req1.csv  (read Data)===replace with NA====>(write to ) req2.csv

req2.csv  (read Data)===remove duplicate ====>(write to ) req3.csv

req3.csv  (read Data)===split depend on User Input ====>(write to ) multi files_

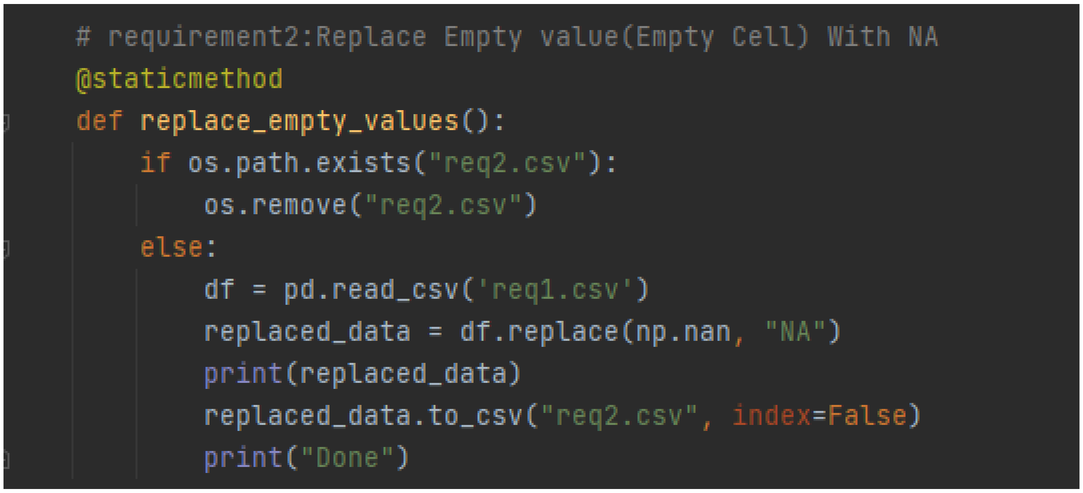
## Screenshot of each part of the requirements

Req1:



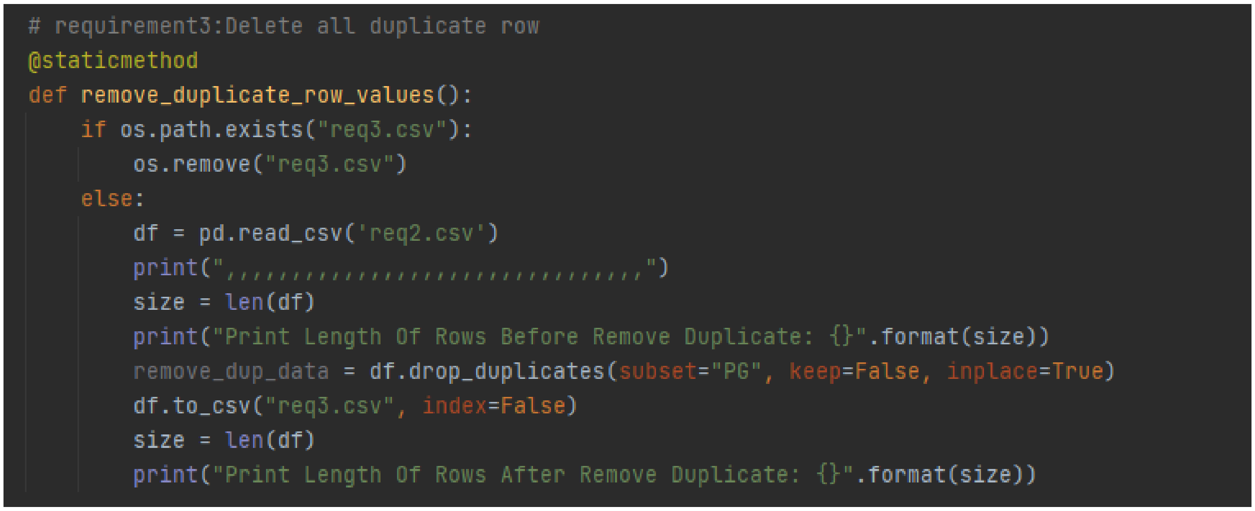
```
# requirement1:Remove all empty lines
@staticmethod
def remove_empty_lines():
    if os.path.exists("req1.csv"):
        os.remove("req1.csv")
    else:
        df = pd.read_csv('datafile.csv')
        df.dropna(how="all", inplace=True)
        df.to_csv("req1.csv", index=False)
        print("Done")
```

Req2:



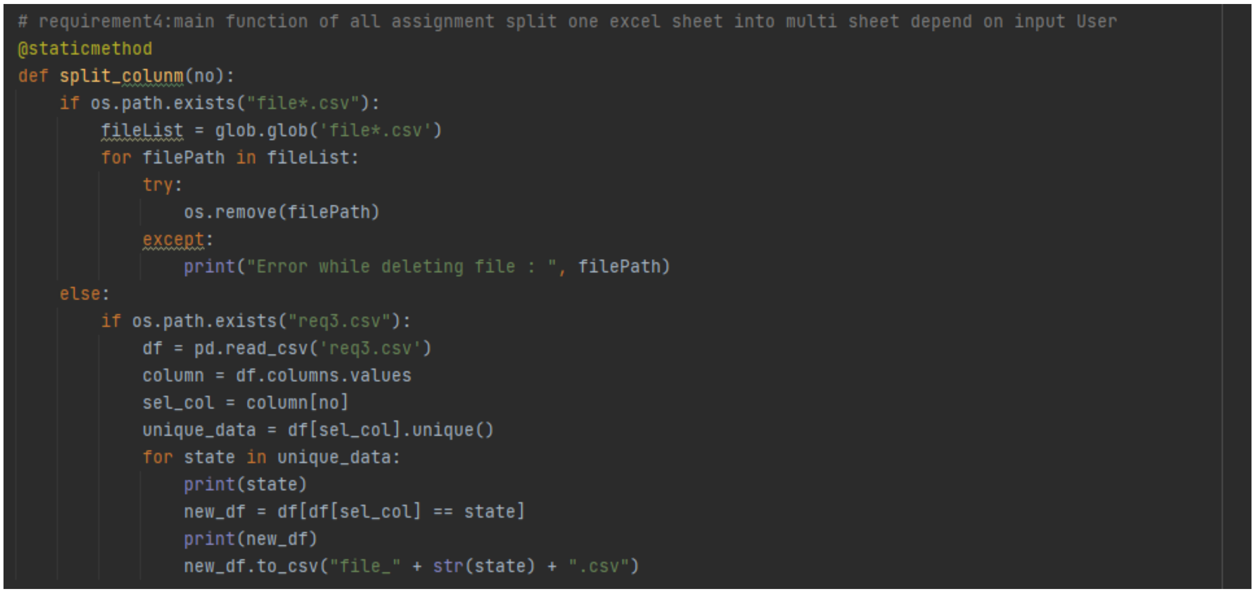
```
# requirement2:Replace Empty value(Empty Cell) With NA
@staticmethod
def replace_empty_values():
    if os.path.exists("req2.csv"):
        os.remove("req2.csv")
    else:
        df = pd.read_csv('req1.csv')
        replaced_data = df.replace(np.nan, "NA")
        print(replaced_data)
        replaced_data.to_csv("req2.csv", index=False)
        print("Done")
```

Req3:



```
# requirement3:Delete all duplicate row
@staticmethod
def remove_duplicate_row_values():
    if os.path.exists("req3.csv"):
        os.remove("req3.csv")
    else:
        df = pd.read_csv('req2.csv')
        print(",,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,")
        size = len(df)
        print("Print Length Of Rows Before Remove Duplicate: {}".format(size))
        remove_dup_data = df.drop_duplicates(subset="PG", keep=False, inplace=True)
        df.to_csv("req3.csv", index=False)
        size = len(df)
        print("Print Length Of Rows After Remove Duplicate: {}".format(size))
```

Req4:



```
# requirement4:main function of all assignment split one excel sheet into multi sheet depend on input User
@staticmethod
def split_colunm(no):
    if os.path.exists("file*.csv"):
        fileList = glob.glob('file*.csv')
        for filePath in fileList:
            try:
                os.remove(filePath)
            except:
                print("Error while deleting file : ", filePath)
    else:
        if os.path.exists("req3.csv"):
            df = pd.read_csv('req3.csv')
            column = df.columns.values
            sel_col = column[no]
            unique_data = df[sel_col].unique()
            for state in unique_data:
                print(state)
                new_df = df[df[sel_col] == state]
                print(new_df)
                new_df.to_csv("file_" + str(state) + ".csv")
```