Approach to the problem of solving Data Engineering Pipeline
By Analytics Vidhya

A brief on the approach: Used Python's Pandas library to solve the problem. First I analyze the data in Jupyter Notebook then creates the pipeline files (.py) using Visual Studio. The data was a bit messy, so I clean it first then impute the NULLs and string conversions were done and at last, used different aggregation techniques to have the result.

Problem Faced and approach to solve them: -

- UserID fields have many missing values which can't be imputed so I dropped the rows
- The VisitDateTime field has both Unix and normal time, so I used the lambda function to find strings that don't have '-' in between, they are Unix time, then I transform them to normal time using pandas built-in function.
- Other Null fields have been filled using forward fill or backward filled after sorting data.
- Many strings are in Upper or Lower case, so I changed them to one uniform Uppercase.
- I found that the sample file has all the UserID present in exact order along with all necessary fields that meant to be calculated. So, I convert it to a dataframe and filled it with my aggregation data, that way I ensure all the id are in the same order as requested.
- For the fields we need to prepare, I used mostly pandas groupby(). I filled the records to a dictionary to map the result with userid.
- When we do aggregation the result will be less in numbers than the actual user id. So for every new field, I created a dictionary and then map it with all user id taken from the sample table. In this way, the performance is increased and time complexity reduced because dictionary mapping is faster.
- At last the NULLs of the end result have been imputed as instructed.

Pipeline: I created three separate python file (.py) for this.

config,py: It is used to set file path and last date for calculation (in this case 28-05-2018)

transformer.py: All the functions and activities are handled here. Every data engineering is done on a separate function to increase the reusability and functionality of the program.

mypipeline.py: The main function is written here, so, we just need to run this file and total feature transformation will be done in a pipeline created and automated using all three files.

Tools used: Python, Pandas, Visual Basic editor, Jupyter Notebook.