A brief on the approach: Use Python’s Pandas library to solve the problem. First I solve it in Jupyter Notebook then creates the pipeline using Visual Studio. The data was bit messy, so I clean it first then impute the NULLs and at last used different aggregation techniques to have the result.

Problem Faced: -

* UserID fields has many missing values which can’t be imputed so I dropped the rows
* The time field has both unix and normal time, so I used a lambda to find strings which 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.
* When we do aggregation the result will be less in numbers than actual user id. So for every new field I created a dictionary and then map it with all user id taken from sample table. In this way the performance is increased and time complexity reduced because of dictionary mapping is faster.

**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 done on separate function to increase reusability and functionality of the program.

**mypipeline.py** The main function is written here, so, you 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.**