**READ ME**

(AUTOMATIC SECTIONIZER FLASK CODE)

JUPYTER NOTEBOOK

FILE NAME = "FLASK\_CODE\_WITH\_COMMENTS.IPYNB"DEVELOPER END EXPERIENCE -This is a read me file for the flask code. This is a basic flask web application which uses python, WSGI (web server gateway interface) libraries. The ‘\_\_main\_\_’ function enables the access to the route that the application is default set for i.e. the home page (http://127.0.0.1:5000/). Every app route has different set of functions which are pieces from the generic python code for the complete process of sectionizer. This is done to avail services for end-user interaction from any location through any server. There are 6 routes to this application some return webpages with a set of specific jobs and some avail downloadable files. *The default location of this flask code file is captured. All the pickled models or any uploaded documents from the server are saved, by default, at the same location. All html files are kept into a folder named “templates” which makes these webpages readily available. Also, any output generated is saved here and is then forwarded for download service to end-user.*a) App route (‘/’) Here the function returns the template named “upload-docx.html” i.e. the homepage (http://127.0.0.1:5000/) where the end-user is asked to upload a file into the HTML form. The data is then sent to next app route (2) using POST method in this HTML form.b) App route (‘/select\_model’) This is where the backend comes into action. The file provided by the POST method from HTML form is acquired by the function named “nlp\_algos”. This function nlp\_algos contains python code pieces which perform data preprocessing and generate vector spaces; thereby, returns the template named “select-model.html” i.e. the webpage (http://127.0.0.1:5000/select\_model/) where the end-user is asked to select a model of their choice using the drop down feature code in the html file. The local variable ‘f’ is where the WERKZEUG toolkit is used to save the file into the default location. The local variable ‘loc’ is used to read path of the uploaded file into the generic python code pieces (This variable can be made “global”, based on the scope of the variable needed between app routes). NOTE: The globally generated variables “result” comes from “run\_all” function which is used later for making predictions. The scope of python libraries is limited to an app route only. The python libraries as needed are to be mentioned again in every app route. c) App route (‘/output’) This is where the end-user provided model option is acquired (which was captured in the html file above using HTML form). The ‘ml\_models’ function uses this particular model file from all the pickled machine learning model files. All pickled model files are readily available since they are saved to the default location. These files are generated from a separate python code which were saved using “from joblib import load, dump” library. The ‘ml\_models’ function, then generates prediction by calling the predict function over the result variable (vector spaces named result that were made globally available in the run\_all function previously generated) into the model i.e. “model\_name” using ‘joblib.load’ feature of the ‘import joblib’ library. While executing other python code pieces, this function (ml\_models) returns the template named “third.html” i.e. the webpage (http://127.0.0.1:5000/output/) along with the generated python dataframe ‘df’ as a table. This DataFrame is saved into default location and made available for download through app route routed “/download\_output”, when HTML form calls for this route using the form action as “/download\_output”. NOTE: This dataframe named as ‘df’ is made globally available for other app routes. d) App route (‘/section\_view’) This provides the end-user with another view of the previously generated Dataframe to view the content in a sectioned format. The dataframe ‘df’ is captured here into a list named ‘l’ which is inside the function named ‘para\_view’. Now, a separating value/string (as "- - - - !!! - - - -") between the paragraphs in this list is appended i.e. when the prediction column for paragraphs (named “Section”) provides input as 1. This is list ‘l’ is then turned into a dataframe named as ‘df2’. This function (para\_view) returns the template named “final.html” i.e. the webpage (http://127.0.0.1:5000/section\_view/) along with the generated python dataframe ‘df2’ as a table. This DataFrame is saved into default location and made available for download through app route routed “/download\_section”, when HTML form calls for this route using the form action as “/download\_section”. NOTE: App route (‘/download\_output’) and App route (‘/download\_section’) actions are mentioned inside c) and d) above respectively.END USER EXPERIENCE -http://127.0.0.1:5000/ is the webpage where the user uploads a document. And submits the document by clicking UPLOAD DOCUMENT button on the webpage.http://127.0.0.1:5000/select\_model/ is the webpage where the user selects the ML model that they want to be executed by clicking RUN MODEL button on the webpage. http://127.0.0.1:5000/output/ is the webpage where the user can redirect to next view by clicking SECTIONIZED VIEW button or can view the table displayed on the webpage, also they can download this table displayed by clicking DOWNLOAD CSV button from the webpage.http://127.0.0.1:5000/section\_view/ is the webpage where the user can view the table displayed on the webpage, also they can download this table displayed by clicking DOWNLOAD CSV button from the webpage.