**Social Capital Toolkit**

This document explains the working and background of the project.

The main.py file is the heart of the project. So, in this file there is a submit function which is getting the request from the user. The request that the submit function is getting is text submitted. In the back end this text data is getting converted into TF IDF vectors and later these vectors are further converted into two-dimensional components using principal component analysis. The files which belongs to transforming these text files are being called in the submit function in this file.

The social capital toolkit Python file will call a function which is social capital toolkit which will give us the connectivity to the SQL workbench. After getting connected to the SQL workbench the SQL data set will get converted into Python data frame. At the same time, we're also calling a function to clean this text data which is into preprocessing.py file. After getting the submission text in the data frame, we are getting the sentiment scores and the polarity scores of the submission text. After that, we are ready to apply the machine learning model on the different text data which are now converted into numbers. We have used K means clustering model for developing the different clusters based on these text data. After fitting the model, we are getting what are the top words in that clusters using get top keywords function within the same file. After that, we are connecting open-source API uclassify to get the clusters interpretation as in what is the cluster signifying. Using uclassify we can get predefined topics which gives the probability of those topics as per the clusters.

To call both files we're connecting the preprocessing files and the modeling files with app and main Python file. There are two more files which are having the HTML front end code to showcase what the user is entering. In the first HTML file we are getting the user input and in the second HTML file we are giving the output of the results interpreted by the model.

We must keep in mind that my SQL workbench is up and running with correct authentication tokens and usernames and passwords. Next step is to call the requirements.txt file which has the different libraries or methods to run our model.

Pip install -r requirements.txt command will help anyone to install all the libraries related to the model inside their own virtual environment. We just need to set up a virtual environment into our PyCharm directory and activate it using the CD command. After activating the virtual environment, we just must call the script main Python file 2 run all the necessary files within the structured folder. This will give us a local host URL which we have to copy into a browser and run our input accordingly.

How the files will be transferred from the GitHub? We must clone the GitHub repository into our virtual environment. Once the GitHub repository is cloned all the files will be copied from the repository to the local system and then from there, we can activate the virtual environment. The data is already in the data folders the static folder is containing the images for the plots and the images used behind the HTML code. The utilities folder contains the preprocessing files and the connectivity files for the data.

We must make sure we do not touch anything in the venv folder because that will harm the virtual environment configuration files. The templates are having the index and the result dot HTML files, the data preparation is having the queries to load into database which is the my SQL code and split into multiple CSV files which are used to get desired data set.

In this part I will paste some screenshots to show what needs to be done step by step.

1. First thing set up a virtual environment in your local system.
2. Clone the GitHub directory or the repository into the virtual environment using <https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository>
3. Once cloned we have to make sure my SQL workbench is up and running.
4. Change the passwords that were suggested earlier. The passwords will be changed in the social capital toolkit file.
5. The requirements.exe file needs to be run from the terminal in which the virtual environment is created. It will be better to use PyCharm like I used for deployment of the model.
6. Next is to run the command pip install -r requirements.txt
7. this will install all the necessary libraries inside the virtual environment and then we can start running the model using the main.py
8. the main board by file will call the necessary codes within the whole folder structure and we just need to give the input for the model.
9. The main file is actually running on flask API to give us the localhost server where and we're doing the get and post request.
10. The files are being run in the following manner which are represented as follows

Run the main.py file

A screen shot of a computer program

Description automatically generated with low confidence

Copy the URL inside a browser

A screenshot of a computer

Description automatically generated with medium confidence

Here you will see the main page where user can input their submission

A close-up of a logo

Description automatically generated with medium confidence

When you give your response into the text box and click submit, in background the model is being run and the clusters information and associated words with sentiment scores and the representation of the clusters will be given as an output

A picture containing text, font, line, screenshot

Description automatically generated

A picture containing text, screenshot, diagram, plot

Description automatically generated

A picture containing diagram, text, screenshot, design

Description automatically generated

After getting this sentiment scores and clusters information we can interpret that the particular submission text is having 0% negative score L 81% neutral score and 18% positive score and if you were to interpret the cluster related to this submission text we can see that it contributes to more than 40% in the business segment and around 35% in the social segment. The cluster information can be interpreted using our own topics here I have used open source API but we can use any API or build our own model on top of the K means model. The intuition behind using the open source API is that when we feed the clusters words inside uclassifier it will give us the probability of predefined topics and how much these topics are related to a particular cluster.