User Manual : CIS\_7030\_20285863

Task 4

Code path: <https://github.com/ritzrepo/CIS7030/>

Data path: <https://icbt-my.sharepoint.com/:f:/g/personal/roshenr_icbtcampus_edu_lk/EnMD8edZyUBEmHGxoWh1ucoBSecQALZ9PxYkG_T4_lGWpQ?e=L1Yoqb>

|  |  |  |  |
| --- | --- | --- | --- |
|  | Folder structure |  |  |
| ID | Project Folder | Subfolder | File description |
| 01 | D:\Projects\TwitterAnalysisDashboards\ | Python project | the main.py for the Dashboard project |
|  |  | \assets | All the css and other assets |
|  |  | \data | tweeter\_data.csv |
|  |  | \environment | Holing the requirements |
|  |  | \myvenv | Virtual environment folder |
|  |  | \src | Holding layout.py file |
|  |  | \ src\components | All the python components py files |
|  |  | \src\data | Loder file |
|  |  |  |  |

Folder structure

A screenshot of a computer

Description automatically generated

Fig 1 : folder structure

|  |  |  |
| --- | --- | --- |
| Steps to run the project |  |  |
| Project | Steps to execute |  |
| TwitterAnalysisDashboards |  |  |
|  | Open the folder in Visual studio code |  |
|  | Open terminal |  |
|  | PS D:\Projects\ TwitterAnalysisDashboards> myvenv/scripts/activate |  |
|  | (myvenv) PS D:\Projects\ TwitterAnalysisDashboards> py main.py |  |
|  | Dash is running on http://127.0.0.1:8050/    \* Serving Flask app 'main'  \* Debug mode: on |  |
|  | Copy the <http://127.0.0.1:8050/> and paste it on the browser window |  |
|  |  |  |
| Requirements.txt | Babel  dash  dash-bootstrap-components  pandas  pandas-stubs  plotly  python-i18n[YAML] | install requirements from an existing file 1  steps   pip install -r .\environment\requirements.txt      python.exe -m pip install --upgrade pip      pip install -r .\environment\requirements-dev.txt |
| Requirements-dev.txt | altair==5.1.2  ansi2html==1.8.0  anyio==4.0.0  argon2-cffi==23.1.0  argon2-cffi-bindings==21.2.0  arrow==1.3.0  astroid==3.0.0  asttokens==2.4.0  async-lru==2.0.4  attrs==23.1.0  Babel==2.13.0  backcall==0.2.0  beautifulsoup4==4.12.2  black==23.9.1  bleach==6.0.0  certifi==2023.7.22  cffi==1.16.0  charset-normalizer==3.3.0  click==8.1.7  colorama==0.4.6  comm==0.1.4  contourpy==1.2.0  cycler==0.12.1  dash==2.13.0  dash-bootstrap-components==1.5.0  dash-core-components==2.0.0  dash-html-components==2.0.0  dash-table==5.0.0  debugpy==1.8.0  decorator==5.1.1  defusedxml==0.7.1  dill==0.3.7  executing==2.0.0  fastjsonschema==2.18.1  Flask==2.2.5  fonttools==4.45.1  fqdn==1.5.1  idna==3.4  ipykernel==6.25.2  ipython==8.16.1  ipython-genutils==0.2.0  ipywidgets==8.1.1  isoduration==20.11.0  isort==5.12.0  itsdangerous==2.1.2  jedi==0.19.1  Jinja2==3.1.2  joblib==1.3.2  json5==0.9.14  jsonpointer==2.4  jsonschema==4.19.1  jsonschema-specifications==2023.7.1  jupyter==1.0.0  jupyter-console==6.6.3  jupyter-events==0.7.0  jupyter-lsp==2.2.0  jupyter\_client==8.3.1  jupyter\_core==5.3.2  jupyter\_server==2.7.3  jupyter\_server\_terminals==0.4.4  jupyterlab==4.0.6  jupyterlab-pygments==0.2.2  jupyterlab-widgets==3.0.9  jupyterlab\_server==2.25.0  kiwisolver==1.4.5  MarkupSafe==2.1.3  matplotlib==3.8.2  matplotlib-inline==0.1.6  mccabe==0.7.0  mistune==3.0.2  mypy-extensions==1.0.0  nbclient==0.8.0  nbconvert==7.9.2  nbformat==5.9.2  nest-asyncio==1.5.8  nltk==3.8.1  notebook==7.0.4  notebook\_shim==0.2.3  numpy==1.26.0  overrides==7.4.0  packaging==23.2  pandas==2.1.1  pandas-stubs==2.1.1.230928  pandocfilters==1.5.0  parso==0.8.3  pathspec==0.11.2  pickleshare==0.7.5  Pillow==10.1.0  platformdirs==3.11.0  plotly==5.17.0  prometheus-client==0.17.1  prompt-toolkit==3.0.39  psutil==5.9.5  pure-eval==0.2.2  pycparser==2.21  Pygments==2.16.1  pylint==3.0.1  pyparsing==3.1.1  python-dateutil==2.8.2  python-i18n==0.3.9  python-json-logger==2.0.7  pytz==2023.3.post1  pywin32==306  pywinpty==2.0.11  PyYAML==6.0.1  pyzmq==25.1.1  qtconsole==5.4.4  QtPy==2.4.0  referencing==0.30.2  regex==2023.10.3  requests==2.31.0  retrying==1.3.4  rfc3339-validator==0.1.4  rfc3986-validator==0.1.1  rpds-py==0.10.4  Send2Trash==1.8.2  six==1.16.0  sniffio==1.3.0  soupsieve==2.5  stack-data==0.6.3  tenacity==8.2.3  terminado==0.17.1  tinycss2==1.2.1  tomlkit==0.12.1  toolz==0.12.0  tornado==6.3.3  tqdm==4.66.1  traitlets==5.11.2  types-python-dateutil==2.8.19.14  types-pytz==2023.3.1.1  typing\_extensions==4.8.0  tzdata==2023.3  uri-template==1.3.0  urllib3==2.0.6  wcwidth==0.2.8  webcolors==1.13  webencodings==0.5.1  websocket-client==1.6.3  Werkzeug==2.2.3  widgetsnbextension==4.0.9  wordcloud==1.9.2 | install requirements from an existing file 2  pip install -r .\environment\requirements-dev.txt |

Data file size

A screenshot of a computer

Description automatically generated

Data structure for the dashboard

**class** DataSchema:

    TWEETID = "Tweet\_ID"

    USERNAME = "Username"

    TEXT = "Text"

    RETWEETS = "Retweets"

    LIKES = "Likes"

    TIMESTAMP = "Timestamp"

    MONTH = "month"

    YEAR = "year"

    DATE = "date"

**def** load\_twitter\_data(path:str)-> pd.DataFrame:

*#load the data from TWEETER csv*

*# Unnamed: Tweet\_ID Username    Text    Retweets    Likes   Timestamp*

    data = pd.read\_csv(

        path,

        dtype={

            DataSchema.TWEETID:str,

            DataSchema.USERNAME:str,

            DataSchema.TEXT: str,

            DataSchema.RETWEETS:int,

            DataSchema.LIKES:int,

            DataSchema.TIMESTAMP:str,

        }, na\_values=["NA", "N/A"],

        parse\_dates=[DataSchema.TIMESTAMP],

    )

    data[DataSchema.MONTH] = data[DataSchema.TIMESTAMP].dt.month.astype(str)

    data[DataSchema.YEAR]= data[DataSchema.TIMESTAMP].dt.year.astype(str)

    data[DataSchema.DATE]= data[DataSchema.TIMESTAMP].dt.date.astype(str)

    return data

Screen shots

1. Year and Month not selected.

A screenshot of a computer

Description automatically generated

1. No month selected.

A screenshot of a computer

Description automatically generated

1. Keywork selected as Party

A screen shot of a graph

Description automatically generated

1. Keyword selected as book

A screenshot of a computer

Description automatically generated

1. Likes and Retweets count

A screenshot of a graph

Description automatically generated

1. Daily tweet count

A graph showing a graph

Description automatically generated

1. Time chart: likes based on date

A colorful lines and dots

Description automatically generated

1. Scatter chart : Likes count based on username

A colorful dots on a white background

Description automatically generated

1. Scatter chart: Relationship between Retweets and Likes

A colorful dots on a white background

Description automatically generated

1. Sentiment Analysis

A close-up of words

Description automatically generated

1. Word cloud from the tweet

A close-up of words

Description automatically generated

The system environment heavy memory and hard disk usage while loading channel, video and comments data into the dashboard.

A screenshot of a computer

Description automatically generated