

DataPreparation	DecisionTree	Features	GloVe	Helper	IbmAPI	LiwcAPI	MapsAPI	ModelApplication	ModelTraining
					apiKey url		gmaps : Client		
clean_text(textString)				printProgressBar(iteration, total, prefix, suffix, decimals, length, fill, printEnd)	get_profile(text, fillProfile, language)	import_liwc_result(fullPath, profileCol)	get_address(geoString)		

Profile
attributeNameList : list big5_agreeableness big5_conscientiousness big5_extraversion big5_neuroticism big5_openness facet_achievement_striving facet_activity_level facet_adventurousness facet_altruism facet_anger facet_anxiety facet_artistic_interests facet_assertiveness facet_cautiousness facet_cheerfulness facet_cooperation facet_depression facet_dutifulness facet_emotionality facet_excitement_seeking facet_friendliness facet_gregariousness facet_imagination facet_immoderation facet_intellect facet_liberalism facet_modesty facet_morality facet_orderliness facet_self_consciousness facet_self_discipline facet_self_efficacy facet_sympathy facet_trust facet_vulnerability language : NoneType liwc_category_list : list numberTweets : NoneType numberWords : NoneType text : NoneType userID : NoneType
load_ibm_json(ibmJson)

ProfileCollection
profileList : list
add_profile(profileObj) get_profile_by_user_id(userID) read_profile_list_file(full_path) write_profile_list_file(full_path)

RequestHandler

TweetCollection
addAttrBool : bool additionalAttributes : NoneType tweetList : list
add_tweet_collection(tweetCol) combine_tweet_text() funcAddTweet(tweetObj) get_distinct_user_id_list() get_id_list() get_tweets_of_userid(userID) read_tweet_list_file(full_path, ids_only, removeNewLineChar) write_tweet_list_file(full_path, ids_only)

TweetObj
created_at : NoneType id_str : NoneType isRetweet : NoneType, int text : NoneType user_id : NoneType

TwitterAPI
additionalAttributes : NoneType api : API ignoreRetweets : bool removeNewLineChar : bool streaming_enabled : bool
funcGetTweetListByUser(userID, limit) funcGetUserID(screen_name) getUsersByList(userIDList, maxFollowerCount, minStatusesCount, skipPrivateUsers) get_followers_of_user(userIDList, limit) get_tweets_by_list(idList, limit) stream_tweets_by_location(location, timeLimit, maxFollowerCount, minStatusesCount)

User
followers : NoneType id_str : NoneType location : NoneType screen_name : NoneType tweet_count : NoneType

UserCollection
userList : list
funcAddUser(userObj) get_id_list() read_user_list_file(full_path, ids_only) write_user_list_file(full_path, ids_only)

_MyStreamListener
additionalAttributes ignoreRetweets : bool limit : int maxFollowerCount : int minStatusesCount : int removeNewLineChar : bool skipFollowerCounter : int skipStatusesCounter : int start_time tweetCollect
on_error(status_code) on_status(tweet)