راهنمای ماژول طبقه‌بندی بات‌های توییتر

* **هدف**: طبقه‌بندی و شناسایی بات‌های توییتر
* **زبان** برنامه‌نویسی: این ماژول به زبان پایتون نوشته شده است.

**فایل‌های پایتون:**

**الف) فایل های مربوط به بخش پیاده‌سازی:**

1. **ماژول‌های پیش‌پردازش:**

* **readExcel.py:** این فایل برای عملیات پیش‌پردازش اولیه روی فایل‌های اکسل ورودی هست که شامل دو فایل AllTweet2.xlsx و users.xlsx می باشد که در پوشه data قرار گرفتند و به ترتیب شامل اطلاعات مربوط به توییت دوماهه (12 دسامبر 2019 تا 20 فوریه 2020) و اطلاعات حساب کاربران منتشرکننده توییت می‌باشد.
* **Utility.**py: شامل تعدادی تابع می‌باشد که بیشتر مربوط به بخش پیش‌پردازش متن توییت‌ها می‌باشد.
  + **تابع** findSpecialWords: که دو مقدار آستانه و حداکثر تعداد کلمه را گرفته و کلمات پرتکرار مخصوص بات و انسان را از کل مجموعه توییت استخراج می‌کند.
  + **تابع** preprocessingandSpecial: کلمات پرتکرار مرحله قبل را گرفته و در هر رشته توییت کاربر تعداد آنها را می‌شمارد. همچنین مقدار TF-IDF و بردار تعبیه کلمات را نیز برای هر رشته توییت بدست‌ می‌آورد که این مجموعه بردارها به عنوان خروجی برگردانده می‌شوند.
  + **تابع** classifierModel که طبقه‌بند، داده‌های آموزشی و تست را گرفته و مدل طبقه‌بندی را در فایل موردنظر ایجاد می‌کند.
* **loadWord2Vec.py**: شامل توابعی برای بارگیری اطلاعات موجود در امبدینگ و ایجاد بردار تعبیه کلمات می‌باشد.
  + تابع loadModel: برای بارگیری اطلاعات امبدینگ fasttext می‌باشد.
  + تابع loadModel\_glove: برای بارگیری اطلاعات امبدینگ Glove
  + تابع make\_agg\_vec: برای ایجاد بردار ویژگی برای یک رشته توییت می‌باشد که میانگین بردار تمام کلمات توییت را در یک بردار نشان می‌دهد.

1. **ماژول‌های طبقه‌بندی:**

* **ClassifiyAccount.py:** ماژولی برای طبقه‌بندی بات بر اساس تنها اطلاعات حساب کاربر. مدل این طبقه‌بند در پوشه model با نام account\_classification\_model.pkl ذخیره شده است.
* **ClassifyTweets.py:** ماژولی برای طبقه‌بندی بات تنها بر اساس متن توییت‌های یک کاربر. مدل این طبقه‌بند در پوشه model با نام tweet\_classification\_model.pkl ذخیره شده است. همچنین لیست کلمات خاص بات و انسان به ترتیب در دو فایل botword.pkl و humanword.pkl ذخیره شده‌اند.
* **ClassifyTweetinfo.py**: ماژولی برای طبقه‌بندی بات تنها بر اساس اطلاعات آماری توییت‌های یک کاربر. مدل این طبقه‌بند در پوشه model با نام tweetinfo\_classification\_model.pkl ذخیره شده است.
* **Alltweetfeature.py:**  ماژولی برای طبقه‌بندی بات بر اساس متن توییت‌ها و اطلاعات آماری توییت‌ها. مدل این طبقه‌بند در پوشه model با نام alltweetfeature\_classification\_model.pkl ذخیره شده است.
* **classifyTweetInfoAndAccount.py:** ماژولی برای طبقه‌بندی بات بر اساس اطلاعات آماری توییت‌ها و اطلاعات حساب کاربر. مدل این طبقه‌بند در پوشه model با نام AccountTweetinfo\_classification\_model.pkl ذخیره شده است.
* **BotDetectionModel.py:** ماژول نهایی که از تمام ویژگی‌ها برای طبقه‌بندی بات استفاده می‌کند و بالاترین صحت را دارد. مدل این طبقه‌بند در پوشه model با نام allfeature\_classification\_model.pkl ذخیره شده است.

**ب) فایل‌های مربوط به بخش عملیاتی سازی:**

در بخش عملیاتی سازی هدف آن است که اطلاعات کاربر و توییت‌های یک بازه زمانی وی مستقیم از الاستیک سرچ خوانده شود و سپس با توجه به مدل طبقه‌بندی، برای هر کاربر برچسب بات بودن یا نبودن در MySQL ذخیره شود.

* **Develop.py**: این فایل اصلی مربوط به این بخش است و 3 پارامتر ورودی دریافت می‌کند. تاریخ شروع، تاریخ پایان و نوع طبقه‌بندی.
  + **تابع** read\_tweets: توییت‌ها و اطلاعات مربوط به آن را در بازه زمانی مشخص شده توسط کوئری زیر از ایندکس socailmedia\_v5 می‌خواند:

query = "SELECT id,publisher,copies,comments,likes,content\_text,shdate,publish\_id,mentions FROM socialmedia\_v5 WHERE source='twitter-tweets' AND shdate>= '{}' AND shdate <='{}'".format(startDate , endDate)

* + **تابع** get\_result\_tweet: مسئول واکشی اطلاعات توییت از الاستیک سرچ می‌باشد. همچنین لیست کاربران منحصربه‌فرد توییت‌های استخراج شده را نیز بدست می آورد.
  + **تابع** read\_profile: اطلاعات حساب کاربران یافت شده را از جدول profiles استخراج می‌کند:

query = "SELECT id, username,follower,following,verified,no\_posts,no\_likes,join\_date FROM profiles WHERE profiles.source='twitter' AND username='{}'" .format(publisher)

* + **تابع** get\_result\_profile: مسدول واکشی اطلاعات حساب از الاستیک سرچ است.

اطلاعات استخراج شده از توییت‌ها و حساب کاربر، پس از پیش‌پردازش با توجه به نوع انتخاب شده مدل طبقه‌بندی، به یکی از مدل‌های بدست‌ آمده در مرحله پیاده‌سازی داده می‌شود و پیش‌بینی بات بودن یا نبودن برای آن بدست می‌آید. در پایان اطلاعات در جدول bot\_users درج می‌گردد.

* **Preprocessing.py:** شامل توابع موردنیاز برای پیش‌پردازش اولیه داده‌های استخراج شده از الاستیک سرچ می‌باشد. مانند استخراج طول توییت‌ها، محاسبه غنای لغوی و ...
* **Applymodels.py:** شامل توابع مختلفی می‌باشد که در هر یک از آنها با توجه به مدل طبقه‌بندی انتخابی، پیش‌بینی انجام شده و نتایج آن در یک فایل ریخته می‌شود.
* **writeToMySQL.py:** شامل توابع مختلفی برای جستجو، حذف و افزودن سطر جدید در جدول bot\_users می‌باشد. برای افزودن سطر جدید در جدول از دستور زیر استفاده شده‌است:

"INSERT INTO bot\_users (userid,user,startdate,enddate,detectdate,classification\_model,subject,isbot) VALUES (%s, %s,%s,%s,%s,%s,%s,%s)"

که شامل id کاربر، نام کاربری، تاریخ شروع، تاریخ پایان، تاریخ تشخیص، مدل طبقه‌بندی، موضوع غالب در بازه زمانی و برچسب بات بودن یا نبودن می‌باشد.

**ج) فایل مربوط به #اعدام\_نکنید**

**Tweet\_case5.py:** توییت‌های بدست آمده برای #اعدام\_نکنید در جدول tweet\_case5 در MySQL جمع آوری شده بود که برای شناسایی بات به ماژول داده شد. کد پایتون این بخش بسیار شبیه کد پایتون develop.py می‌باشد با این تفاوت که در این کد اطلاعات توییت‌ها به جای آنکه از الاستیک سرچ خوانده شود از MySQL خوانده می‌شود. همچنین این ماژول تنها یک ورودی به نام مدل طبقه‌بندی دارد.

**آدرس دسترسی به گیت**: <http://172.20.81.145/gitlab/namvar/botdetection/tree/master>

**Run**

Two run, write in terminal:

python3 develop.py startDate endDate classification\_model subject

For example: python3 develop.py 1398-12-24 1399-04-12 tweetinfo corona

**Parameters Description**

1. **startDate:** response to khorshidi date, like "1398-12-24"
2. **endDate:** response to khorshidi date, like "1398-12-29"
3. **classification\_model:** specify classification model
   1. **account:** bot detection based on user account
   2. **tweet:** bot detection based on the text of tweets
   3. **tweetinfo:** bot detection based on statistical information of tweets
   4. **alltweet:** bot detection based on statistical information of tweets and the text of tweets
   5. **accounttweetinfo:** bot detection based on user account and statistical information of tweets
   6. **all:** bot detection based on all features (default)
4. **subject**: specify the important topic of the time period