Automatic Prediction of Car Brand Ownership Based on Tweets

Wei Xu	**Weizi Liu**	**Renhan Zhang**	**Bohan Zhao**
weixu@umich.edu	weizliu@umich.edu	rhzhang@umich.edu	zhaoboha@umich.edu

README

Summary for packages to be installed

To be able to run our project code, you need to install a list of packages which are described as follows:

- Python 2.x
- Numpy
- Scipy
- Scikit Learn
- Tweepy
- twitter-python
- Ipython
- pydot

Please make sure you install all these required packages, and we surely will give details and instructions about how to install them. If you still have any problems about installing or running our code, feel free to contact us.

How to run our code

1. Crawl general tweets from Twitter API given car brands

To run the program crawling valid twitter users for training, first you need to install tweepy, a python package for twitter crawling. To do this, type `pip install tweepy` in the terminal. Second, cd to the directory "*Grab Tweets*" and execute the program by typing the following in the terminal: *python Grab_tweets.py*. If done correctly, the program should collect tweets and append them to the file *Grab_tweets.csv* under the same directory.

2. Crawl tweets from Twitter API for specific Twitter USER ID

In order to crawl tweets with a specified user id, you need to download **twitter-python** from https://github.com/bear/python-twitter.git(you can input the following in your terminal: `git`

`clone git://github.com/bear/python-twitter.git`), and then there is a directory named "*grab tweets with a specified user id*", which contains a directory named "*twitter*", that is what we want to run pdata_user.py. Put pdata_user.py, twitter directory and user_id.csv into a directory, type *python pdata_user.py* in your terminal, you will get a directory named "*Complete Datasets*" in the current directory. In this "*Complete Datasets*" directory, there are csv files named as <user id>.<car branch>.csv, which contains crawled tweets of that user, those are our annotate data.

Note: We have already finished retrieving tweets for each user, and put the complete dataset in the parent directory. This complete dataset is what we use for prediction model.

3. Run Naive Bayes Prediction Model

In order to run Model_NaiveBayes.py, firstly you need to install **Scikit-Learn** and **IPython** in your computer. To install the sklearn in you computer, please refer to internet resource to see how to install in your type of computer.

Then, you can input *pip install IPython* in your terminal to install IPython.

Once set up, please go to the folder "Naive Bayes", and run the file Model_NaiveBayes.py by typing *python Model_NaieBayes.py* in the terminal. The code will output system accuracy, classification report and confusion matrix in your terminal.

4. Run Decision Tree Prediction Model

In order to run Model_decisionTree.py, please firstly install **pydot**, input the following commands in your terminal:

```
pip uninstall pyparsing
pip install -Iv
https://pypi.python.org/packages/source/p/pyparsing/pyparsing-1.5.7.tar.gz#md5=9be0fcdc
c595199c646ab317c1d9a709
pip install pydot
```

The first two are to ensure you have the right version of pyparsing to use pydot. We need to modify the originally installed scikit learn package to get visualized decision tree and map from vocabulary index to vocabulary (*vocabulary will be stored in a list, and in visualized decision tree, there will be an index instead of that vocabulary, e.g.X[1000], we need a map from this index to the real vocabulary*). But do not worry about the modification, we have

included the modified sklearn package in our directory of “Decision Tree”(which also contains Model_DecisionTree.py). You can directly use it without any other modification, provided you have already installed scikit learn package.

So, just run *python Model_DecisionTree.py* in your terminal, and the code will output system accuracy, classification report, confusion matrix in your terminal. Besides, the code will also write the visualized decision trees and maps into the local directory.

In case that you have trouble in running our code, we also provide our running result in the folder “*our results*”, please refer to that to see the results of Decision Tree method.

5. Run SVM Prediction Model

To run the SVM prediction model, you should make sure that you have installed the scikit-learn package in your computer. Then go to the folder “*SVM*”, and run the program by typing:

`python Model_SVM.py` in your terminal, and the code will the system accuracy, classification report and confusion matrix in the terminal.

Note: To speed up the classification process, we have stored the trained model into the folder “svm_model”, and the Model_SVM.py will only read the saved model and do the test process here.

Still, if you want to run the whole training process and test process, there are two tips needed to be noticed:

1) Because training model needs long time (about 20 minutes), we have save our trained models in the folder ‘svm_model’, hence make sure ‘svm_model’ in the same directory with ‘Model_SVM.py’. And by default, the ‘Model_SVM.py’ applies the trained models in the ‘svm_model’ folder. If you want to training the model by yourself, you should change the code in ‘Model_SVM.py’ (there is instruction to tell you how to change the code in ‘Model_SVM.py’ to training the model).
2) The ‘twokenize.py’ is the tokenizer from CMU. By default, we apply the tokenizer from sklearn rather than CMU Tweets tokenizer. If you want to apply the CMU Tweets tokenizer, you should change the code in ‘Model_SVM.py’ (there is instruction to tell you how to change the code in ‘Model_SVM.py’ to apply the CMU tokenizer).

6. Analyze Most Frequent Words

To see the most frequent words for each of the car brand, go to the directory "*Most Frequent words*" and run the program by typing *python Most_Frequency_Word.py* in the terminal. If the code is running correctly, the program should print the 30 most frequent words for each of the car brand in your terminal.