SENTIFI JUNIOR DATA SCIENCE ASSIGNMENT

1. **Model discussion**

a. Variables

I fetched the user information and their tweets (excluding tweets that retweet others' tweets). They are the most basic and accessible pieces of data yet very helpful. Other information such as the retweets of their tweets, the tweets they retweeted, or other tweets that used the same hashtag they used might be helpful as well but I would like to start simple first.

User-level variables:

- favorite count (count of 'following')
- description
- listed count
- statuses count
- followers count
- friends count

Tweet-level variables

- average retweet count
- max retweet count
- average favorite count
- max favorite count
- up to 600 latest tweets

Many times twitter user's occupation can be inferred from their description:

```
description
Politician
['councillor', 'senat', 'member', 'labour', 'conserv', 'minist', 'chair']
Trader
['trader', 'forex', 'stock', 'trade', 'fx', 'financi', 'option']
Journalist
['journalist', 'report', 'com', 'produc', 'editor', 'writer', 'photojournalis
t']
```

And they often mention technical terms in their tweets:

```
tweets
Politician
['wisen', 'wvlegi', 'wvchemleak', 'feingold', 'amp', 'teamronjon', 'council']
Trader
['eurusd', 'esf', 'forex', 'gbpusd', 'zulutradepnl', 'tgt', 'daytrad']
Journalist
['avlnew', 'wordwatch', 'newscentermain', 'toread', 'wkbtnew', 'wjtv', 'senza
glutin']
```

b. Evaluation metrics

I used accuracy as the performance metrics since this dataset doesn't suffer from imbalance and I don't know the priority of the task. However, I will also present precision, recall, and f1 score since different priority would use different metric (for example, if capturing traders is much more important, then the model with high recall for traders should be preferred).

c. Feature preprocessing

There are 41 user ids in the file that can no longer be found. I eliminate these users from the dataset since I can't collect any information about them. Therefore, there are 515 samples left.

There are also users who protects their account or do not have any self-post tweets. We can only extract user information in these cases.

For user's description and tweets:

- Remove links
- Remove mentions
- Keep only space and alphabetical characters
- Use Snowball stemmer
- Use Tfidf on training set to calculate score of words in each category (score of a word depends proportionally on its frequency in that category and inversely on its frequency in all categories). Keep top terms in each category to build vocabulary for vectorizers. Use count vectorizer for Naïve Bayes and tf-idf vectorizer for others (because using count vectorizer improve Naïve Bayes' performance by at least 40% in this problem)

For numerical variables:

- Take the logarithm: lambda x: log(1+x) since they are highly skewed. Taking the logarithm significantly increase performance, especially Naïve Bayes and SVM.

d. Model building

I employed four models: Logistic Regression, Naïve Bayes, Random Forest, and SVM with linear kernel. These models are simple, interpretable (easy to extract the most significant features).

To evaluate each model, I repeat the following process for 30 times:

- Split into train and test set (20%) with different seed
- Take the test accuracy as the score

By obtaining 30 independent results, I can obtain the mean and standard deviation of cross validation performance. Note that with each run, training samples and the vocabulary built from training text is different.

For the first phase, I trained the above four models with different vocabulary sizes (20% top terms and 30% top terms):

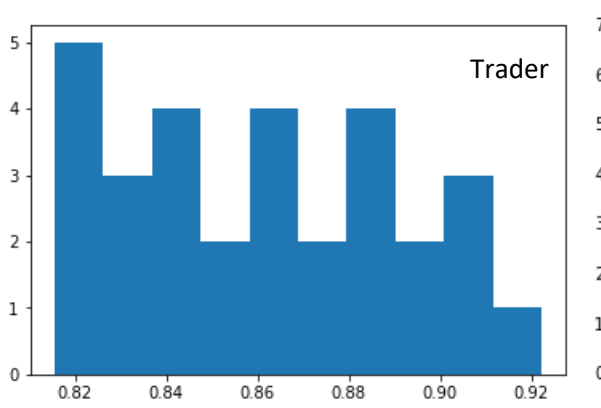
	Journalist		Politician		Trader	
Model	**Average**	**StdDev**	**Average**	**StdDev**	**Average**	**StdDev**
LG	**0.7652**	**0.0435**	**0.8913**	**0.0188**	**0.8744**	**0.0277**
0.2	0.7647	0.0421	0.8909	0.0189	0.8741	0.0269
0.3	0.7657	0.0456	0.8916	0.0191	0.8748	0.0290

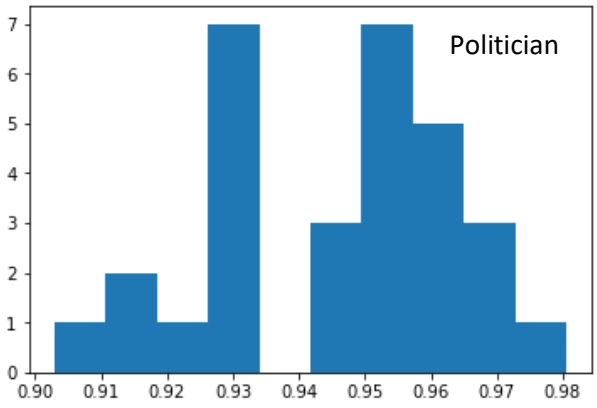
NB	**0.7244**	**0.0559**	**0.8872**	**0.0220**	**0.8225**	**0.0321**
0.2	0.7223	0.0565	0.8903	0.0214	0.8217	0.0330
0.3	0.7265	0.0563	0.8841	0.0225	0.8233	0.0318
RF	**0.7850**	**0.0280**	**0.8945**	**0.0226**	**0.8583**	**0.0361**
0.2	0.7858	0.0313	0.8955	0.0233	0.8618	0.0346
0.3	0.7841	0.0248	0.8935	0.0223	0.8547	0.0378
SVM	**0.8029**	**0.0392**	**0.9259**	**0.0231**	**0.8982**	**0.0231**
0.2	0.8013	0.0396	0.9249	0.0242	0.8981	0.0228
0.3	0.8045	0.0393	0.9269	0.0223	0.8984	0.0239

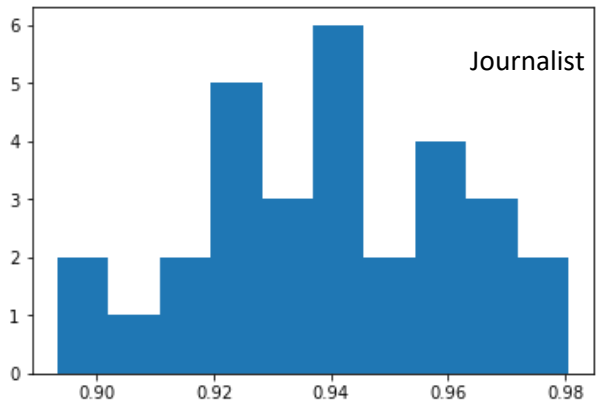
SVM is the winner for all categories and its performance is quite consistent. In most of the cases, using top 30% terms as vocabulary boosted performance of all models slightly.

Therefore, for phase 2, I decided to further tune the model with different penalty term for SVM, vocabulary size, and whether to balance the class weight for every category. This time, I still used test performance but didn't run for 30 times since the models look quite consistent to me. It seems that the winning configuration is the origin with C=1, linear kernel, and as-is class weights. However, the vocabulary size that seems best for politician is 0.1, trader 0.3, and journalist 0.5.

For phase 3, I tested the configuration for 30 times again. The model performance seems consistent and therefore I will use these models.







e. Result

Since my current models are one-vs-rest, I take the class with the highest probability as prediction for each sample. This way, the test accuracy, recall, precision, and f1-score are all above 86% and are 87.4%, 87.4%, 86.6% and 86.7% respectively.

2. Future Improvement

I would draw data from tweets that are retweets because the retweets also reflect what they care about. My model currently miss users with non-trivial number of followers but have no or few self-made tweets. Additionally, I would like to obtain the following data:

- Retweets of short and ambiguous tweets: they might be ambiguous to a newbie like our model. Looking at the retweets helps our model to understand better. In addition, there are terms that have different meanings when used in different industries. Reading more content about such terms might be helpful.
- Hashtags used in short and ambiguous tweets. Same reason as above, especially that hashtags are frequent used in tweets
- Description, tweets, or known labels of the targetted user's followers and people who they follow.
- Tweets that mention them.
- English translation for non-English text

In addition, we prioritize predictive performance, I would also like to use LSTM or other neural network models with word-to-vec features since they are often used in NLP tasks. I think word-to-vec will be very useful to analyze hashtags. However, it is quite difficult to extract significant features from these NN models while our easy-to-interpret models are doing quite good.

3. File description

Original file with label: user_ids.csv

Crawl: python download.py [consumerkey] [consumersecret] [access_token_key] [access_token_secret]

Process data: python process.py

Phase 1 cross validation (try out multiple models for 30 times):

Python build_model.py

Build_model.ipynb: phase 2 + phase 3 + final evaluation: examine different configurations for SVM and vocabulary sizes. Test for robustness of the model by running 30 times. Evaluate the model as a whole.