Kaggle team name: Dota2 is a free game
Team member: Yiting Wang(yw883); Siyao Huang(sh2435); Shengqi Wang(sw979); Hanyang Zhang(hz547); Mengceng He(mh2387)
Your score on the public leaderboard: 0.84166
Citations for any code you have used that isn't your own.

All the code are written by ourselves. We use a lot of packages, like keras, TensorFlow, sklearn, nltk, pandas, numpy.

The preprocessing techniques used

a. use pandas to read the csv file
b. use nltk to separate the text into words
c. use hash to map different words into certain index (the size of vocabulary is 5000)

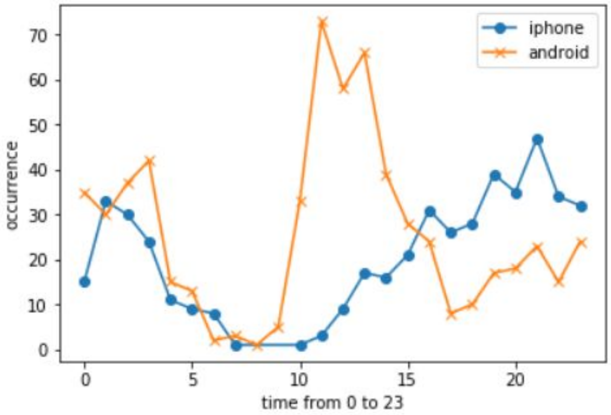
Previous attempts:

1. RNN (only text):
   a. using the same hash function, but keep the order of the mapped words
   b. shape output length to 150, fill the empty space as 0.
   c. import function from Tensorflow to use RNN model to process the input.
   d. get output, the accuracy of which is the same as accuracy of professor's output
2. RNN+CNN/SVM/RandomForest (text+favoriteCount+retweetCount):
   a. combine the output layer of the RNN with new input (favoriteCount+retweetCount) as the new input layer
   b. put the new input into CNN/SVM/RandomForest
   c. CNN and SVM model output lower output with lower accuracy
   d. Used the output from RNN as a feature in Randomforest. Also added time as another feature. Got the result like 79%
3. SVM (only text)
   a. Used sklearn text feature extraction to get training data.
   b. Trained a SVM to predict data. Got the score around 71%
4. Random forest (time + favoriteCount + retweetCount + whether there is a link in text):

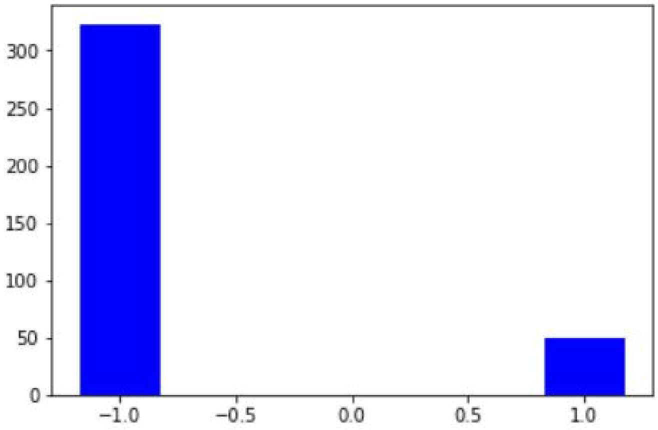
After analyzing data, I figured that post time is important. And if there is a link in the text, it's highly possible that that tweet is post by iphone. So I use these features to train random forest. The result I got was like 78%.

What you learned while exploring the data

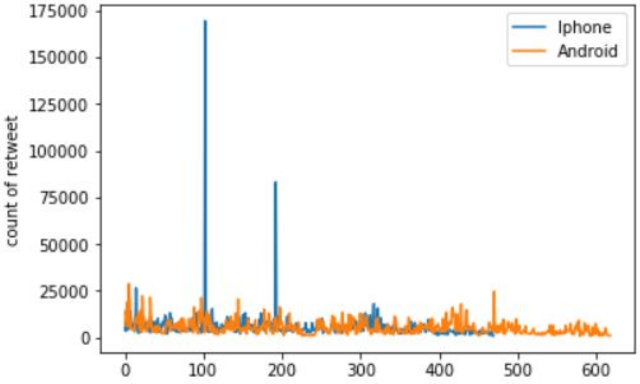
There are some patterns of time (hour):



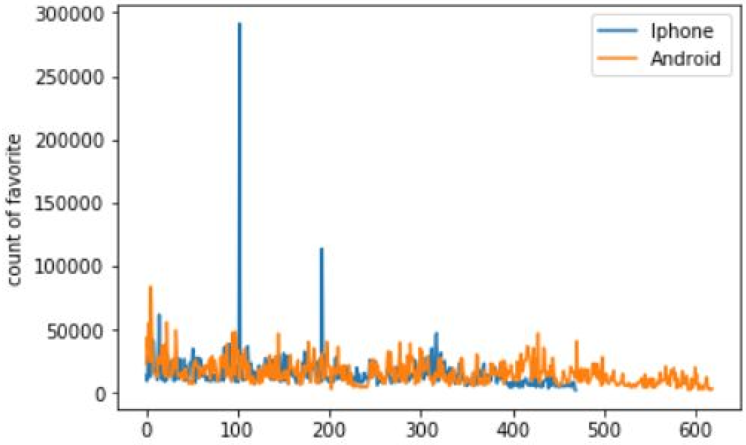
Whether the text contain link or not:



Count of retweet:



Count of favorite:



How you selected your model

We have tried RNN, SVM, Naive bayes, RNN+random forest and random forest. Finally we found random forest model performs better than others.

What features you extracted

We only use text feature of twitters. We use hash function to extract the word in twitter.

How you searched for hyper-parameters

We wrote a for loop to check the validation accuracy of different hyper-parameters, and choose the best hyper-parameters in our mind.