Working with Twitter data to predict the number of reported cases of Influenza using a Support Vector Machine Learning algorithm

Written by Clayton Chapman (ICS 2018)
Advised by D. Mutchler, P. Graf

Hochschule Ulm

Table of Contents

- Problem Background
- Initial Research
- Revised Approach
- Applied Methods
- Support Vector Machines in Depth
- Results
- Further Research

Twitter: The Social Database

- A social media website with short, constantly updated statuses
    - 6,000 tweets posted every second worldwide
- Extremely diverse and immense database of information
- Robust API to allow for straightforward mining
    - Considerable amount of information is attached to a tweet
- I can build tools to use Twitter and the data associated with tweets to produce a solution to a Data Science problem

Initial Research

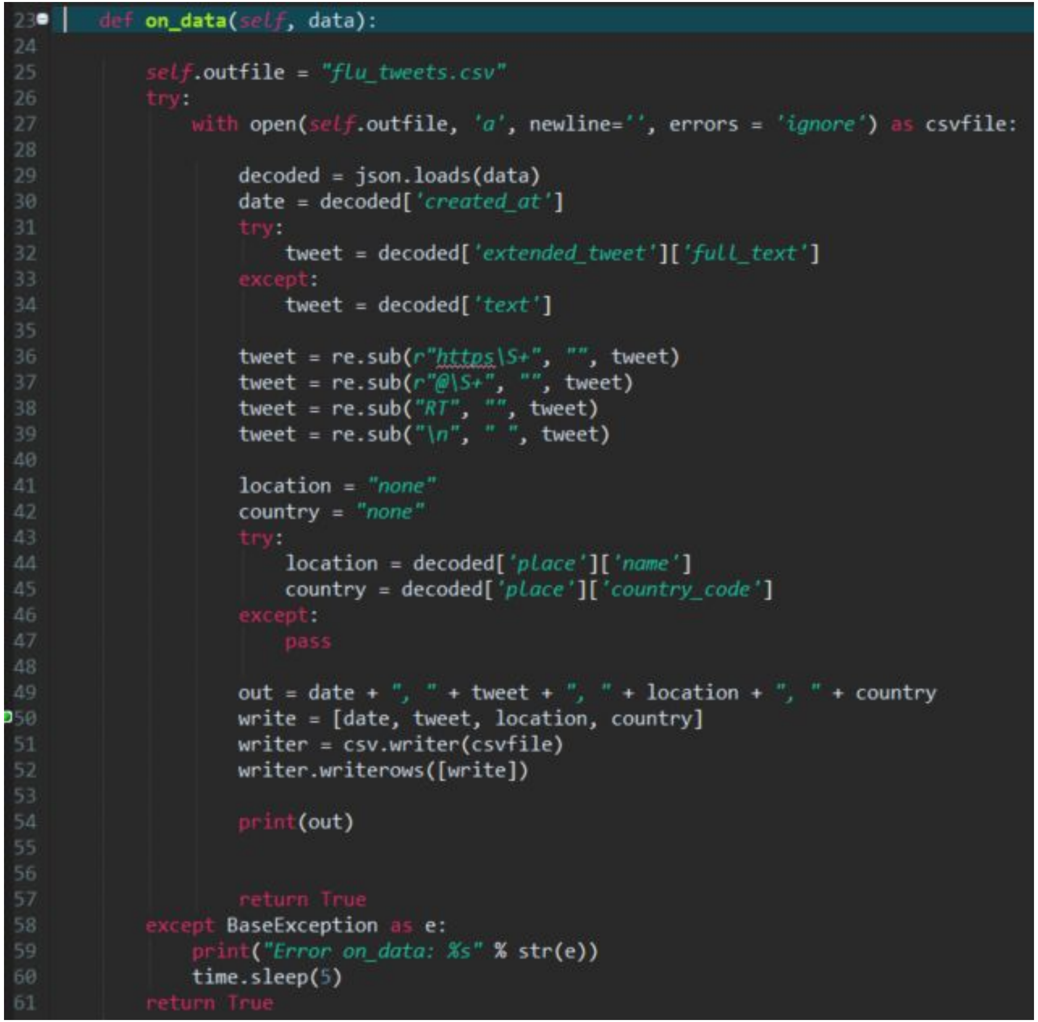
- Use Twitter and Machine Learning to analyze personality
- *Deep learning for constructing microblog behavior representation to identify social media user's personality* by Xiaoqian Liu, Tingshao Zhu
- Big Problem: Personality is not easy to accurately measure
- The thesis would need a better control variable

Tracking Influenza

- While being an inconvenience to many, the virus is deadly to some
  - Awareness is important
- CDC Reports contain useful information about cases of flu and location of cases
  - This method is slow
  - The “Current” weekly report is for April 21-28
- Could Twitter be used as an early prediction for future reports?

Revised Research

- *Dengue prediction by the web: Tweets are a useful tool for estimating and forecasting Dengue at country and city level*
- I shifted my focus to comparing tweets about the Flu to actual reported cases of the Flu
- My approach:
  - Mine the tweets
  - Use Machine Learning to classify the tweets
  - Compare the data with Official Reports

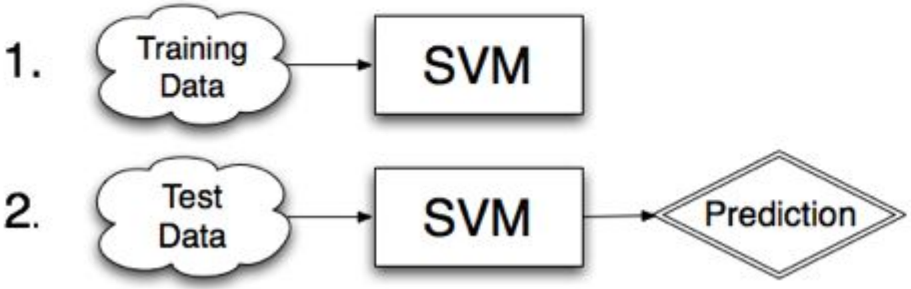


```python
    def on_data(self, data):

        self.outfile = "flu_tweets.csv"
        try:
            with open(self.outfile, 'a', newline='', errors = 'ignore') as csvfile:

                decoded = json.loads(data)
                date = decoded['created_at']
                try:
                    tweet = decoded['extended_tweet']['full_text']
                except:
                    tweet = decoded['text']

                tweet = re.sub(r"https\S+", "", tweet)
                tweet = re.sub(r"@\S+", "", tweet)
                tweet = re.sub("RT", "", tweet)
                tweet = re.sub("\n", " ", tweet)

                location = "none"
                country = "none"
                try:
                    location = decoded['place']['name']
                    country = decoded['place']['country_code']
                except:
                    pass

                out = date + ", " + tweet + ", " + location + ", " + country
                write = [date, tweet, location, country]
                writer = csv.writer(csvfile)
                writer.writerows([write])

                print(out)


                return True
        except BaseException as e:
            print("Error on_data: %s" % str(e))
            time.sleep(5)
        return True
```

Twitter API

{"retweet_count":0,"text":"Man I like me some @twitterapi","entities":{"urls":[],"hashtags":[],"user_mentions":[{"indices":[19,30],"name":"Twitter API","id":6253282,"screen_name":"twitterapi","id_str":"6253282"}]},"retweeted":false,"in_reply_to_status_id_str":null,"place":null,"in_reply_to_user_id_str":null,"coordinates":null,"source":"web","in_reply_to_screen_name":null,"in_reply_to_user_id":null,"in_reply_to_status_id":null,"favorited":false,"contributors":null,"geo":null,"truncated":false,"created_at":"Wed Feb 29 19:42:02 +0000 2012","user":{"is_translator":false,"follow_request_sent":null,"statuses_count":142,"profile_background_color":"C0DEED","default_profile":false,"lang":"en","notifications":null,"profile_background_tile":true,"location":"","profile_sidebar_fill_color":"ffffff","followers_count":8,"profile_image_url":"http:\/\/a1.twimg.com\/profile_images\/1540298033\/phatkicks_normal.jpg","contributors_enabled":false,"profile_background_image_url_https":"https:\/\/si0.twimg.com\/profile_background_images\/365782739\/doof.jpg","description":"I am just a testing account, following me probably won't gain you very much","following":null,"profile_sidebar_border_color":"C0DEED","profile_image_url_https":"https:\/\/si0.twimg.com\/profile_images\/1540298033\/phatkicks_normal.jpg","default_profile_image":false,"show_all_inline_media":false,"verified":false,"profile_use_background_image":true,"favourites_count":1,"friends_count":5,"profile_text_color":"333333","protected":false,"profile_background_image_url":"http:\/\/a3.twimg.com\/profile_background_images\/365782739\/doof.jpg","time_zone":"Pacific Time (US & Canada)","created_at":"Fri Sep 09 16:13:20 +0000 2011","name":"fakekurrik","geo_enabled":true,"profile_link_color":"0084B4","url":"http:\/\/blog.roomanna.com","id":370773112,"id_str":"370773112","listed_count":0,"utc_offset":-28800,"screen_name":"fakekurrik"},"id":174942523154894848,"id_str":"174942523154894848"}

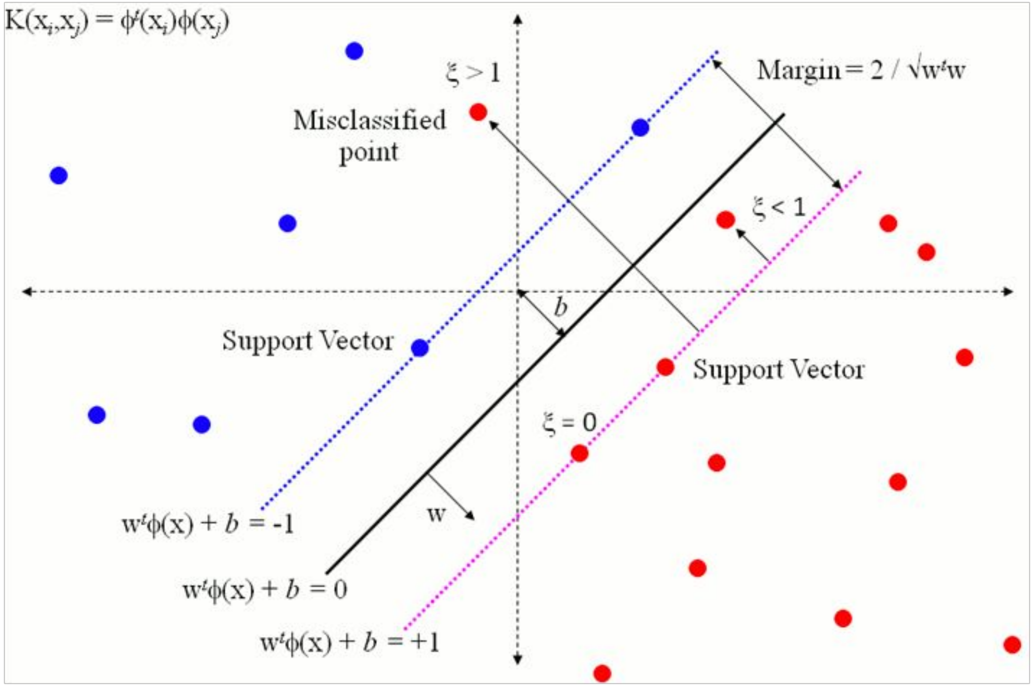
Support Vector Machine



- Supervised learning, so it needs to train with a buddy/human
- The complete training set makes up a training subset and a testing subset
- More detail on the math involved later

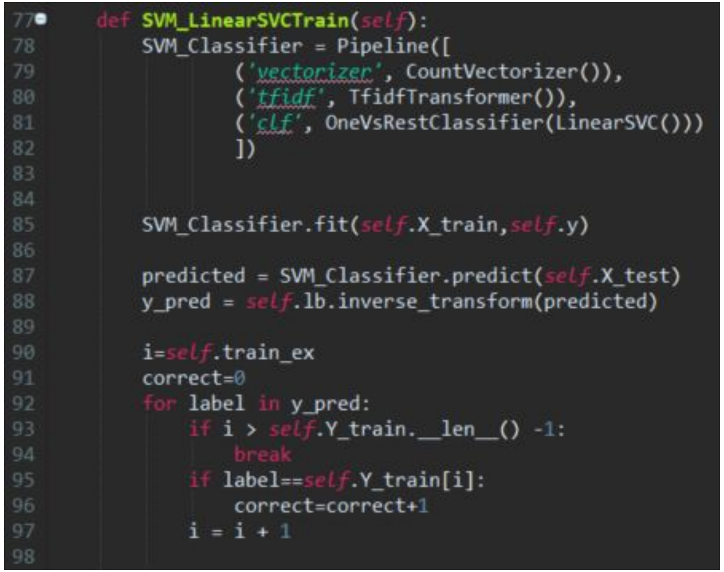
SVM's cont.

- Closest points are the "Support Vectors"
- Hyper-plane wants the greatest margin, which means minimizing 'w'
- Also possible to tweak SVMs



My Support Vector Machine

- Uses sci-kit SVM model
- 1,000 tweet training set
  - 90/10 split
  - Obtained 90% accuracy

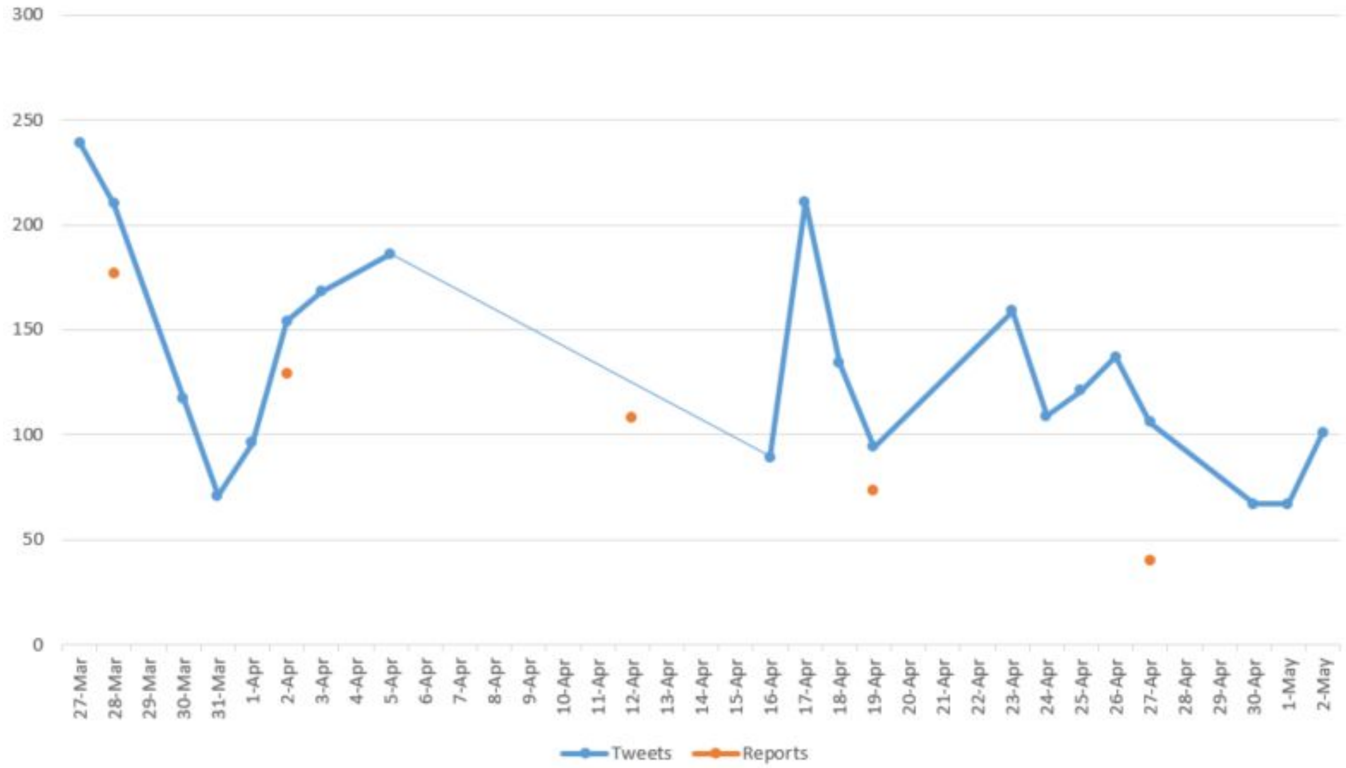


```python
    def SVM_LinearSVCTrain(self):
        SVM_Classifier = Pipeline([
                ('vectorizer', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('clf', OneVsRestClassifier(LinearSVC()))
                ])


        SVM_Classifier.fit(self.X_train,self.y)

        predicted = SVM_Classifier.predict(self.X_test)
        y_pred = self.lb.inverse_transform(predicted)

        i=self.train_ex
        correct=0
        for label in y_pred:
            if i > self.Y_train.__len__() -1:
                break
            if label==self.Y_train[i]:
                correct=correct+1
            i = i + 1

```

Results

- Obtained 8,700 tweets from 20 sessions
- 2,650 of these tweets would be labeled as valid
  - Only 60 of these tweets were “from America”
- Large assumption being most of these tweets were still from Americans

Results cont.



Conclusion

- I built tools using Twitter API and an existing Support Vector Machine
- I was able to gather tweets and classify them accordingly
- The results showed a promising relation to Official Flu Reports
- Twitter was indeed used for answering a Data Science problem
- Success!

Future Research

- Leaving Twitter stream open with multiple machines
- Finding another way to obtain location data
  - Compare results by region
- Extremely Versatile Framework to be used with many other problems

Thank you, dankeschön

SVM Advantages over NN (and Disadvantages)

- SVMs are simple, requiring little initial data and starting code
  - NNs need massive training sets and time to create working model
- The math of SVMs is faster and cheaper
  - A good NN will need heavy GPU resources
- SVMs can be tweaked and tuned, and even used for non-linear classification
- Because SVMs only work with support vectors, less prone to overfitting data
- But, SVMs with multi-classifiers need to each be trained and used separately
  - NNs can train and use all classifiers at once
- SVMs have no understanding of Grammar, just patterns and counting