twitter nlp project - Elior Cohen

ABSTRACT

In the project, i am going to collect all hillary's and trump's tweets from the last elections in 2016 till 2018, im have used the tools that we have learned threw all the semester: Data collection, data cleaning, working with text data(vectorizing),machine learning algorithms such as logistic regression, KNN, etc…

My goal in this project is to predict the writer of a specific status(actually, to calculate the probability that this specific status wrote by hillary clinton or donald trump).

my biggest goal is to achieve the best probability based on the train data and based on the machine learning algorithms.

INTRODUCTION

I have tried to build up a good model, that will predict in a pretty good probability if a randomly picked status is belong to a specific candidate.

Data acquisition

firstly, i needed to look after twitter api rules about data acquisition, i saw that there are some limitations i need to go after.. i have collected the data via twitter api tool. i needed to register on twitter developers website, but it couldn't get enough tweets to build up a good model,because of the limitations i had with my free key, so i decided to get an enterprise key, i did it via family connections, more specifically: my uncle is working on a machine learning startup and got me an enterprise key, so i could collect enough tweets to build up my model. After that, i needed to write up a python class that will collect the data from twitter api and transform it into a pandas dataframe, so the data will be more suitable and comfortable to work with in the next steps. Also, i needed to take all the tweets from a specific date range, so i have used a twitter api parameter called max_id, with that, i could get the tweets from the last election to 2018. Twitter api gave me the data in a JSON format so i have needed to convert it to a python dictionary and then to a pandas dataframe. When i have collected a tweet in twitter api i have got some very important fields i will have to work with later in the project: 'text' , 'created by' , 'mined at', etc…

After i did all that i wanted to know that the dataframe contains everything i need to work with later so i printed it out to visually see the dataframe itself(columns, rows), after i saw that the data is organized in a pandas dataframe and everything is seems to be pretty good i decided to start the next step in my project : Data cleaning.

DATA CLEANING

When i looked into the tweets i saw that a lot of tweets are full in some very disturbing data that wont help me very much in the next steps of the project, so i decided to take care of it and i have used textacy library to preprocess the text in the dataframe.

I decided to delete: phone numbers, email addresses, urls and links, symbols, accents, punctuations, etc…

i did it with textacy library, i used preprocess_text function and passed the proper parameters.

After that, i wanted to see if the preprocess is actually done and the text is better for training the model, so i have just printed out the cleaned text and saw that it actually worked, so i did it to all the dataframe using for loop in python.

EDA

I wanted to see if there is something interesting in the data(popular words by each of the candidates, and number of appearances for each word), so i have converted the data into a vector, with n-grams of 2-5, and then converted it into a giant string(one for hillary, one for trump), then, i used counter library to print out the 20 most common set of words for each candidate.

I have noticed that trump likes to use these set of words: ‘fake news’,’america great’...

and hilary likes to use these set of words: ‘donald trump’,’make sure’...

it was very interesting to see that.

PREDICATIVE OR LEARNING EXPERIMENTS

Firstly, i needed to create a Y array to train the model. so i used lambda function and converted trump tweets to be the value if 1 and hillary’s tweets to be the value of 0 in the ‘handle’ column, Then i needed to build up a X matrix also, so i have used TFIDFVECTORIZER to create a X matrix that is actually vector contains the different words and their frequently appearances, i printed out the shape to see that everything is working fine.

I decided to use logistic regression because it was a classification problem, So i needed to find the optimal parameters to pass to the logistic regression so i have used GridSearchCV to find them. i saw that the best parameters that are going to give me the best score are L2 penalty and c = 1 and the best score i have got with these parameters is 0.85037, not bad at all.

After that, i had to fit the model, while passing X and Y. After that, i needed to check my model in a real-time situation! so i have tried to take a random status, wrote by hillary, same to trump, that status was not recognized by my model because it wasn't included in the training data set. surprisingly, my model was not very bad at all: when i passed in Hillary's status i have got a 0.899 probability that this specific status belong to her, and when i passed in a new status, wrote by trump, i have got a 0.686 probability that this status is wrote by him! So based on the model, the probability the first tweet came from Hillary is almost 90%! The probability the second tweet came from Trump is almost 70%. So I would say the model is performing pretty well.

CONCLUSIONS

like i have presented in the previous paragraph, my model was actually not bad at all, and i have reached a very high probabilities that was actually very accurate. it was very interesting to learn everything, i have used GOOGLE a lot and learned to work with that, i also used youtube videos and the course lectures to understand whether im doing something wrong or not, it was not very simple to create the model. The project actually gave me some great future ideas to do next and I will be very proud of myself if i will have the chance to work in this strongly interesting field of computer science in the future, i have learned that machine learning is a very powerful tool that let you do unbelievable things and predictions!