Laura Daniela Vargas Herrera A01203961
María Fernanda Vargas Herrera A01203957

Connect Four and Twitter Sentiment Analysis

Connect Four

Connect Four is a two-player game in which each player has a color and they have to drop the piece into a 6 x 7 grid. The objective is to connect 4 pieces of the same color, doesn't matter if they are connected horizontally, vertically or in diagonal.

We choose Connect Four because it is a solved game; it means that the state of the board can be predicted assuming that both players play perfectly, minimax and alpha-beta algorithms do that, they minimize the possible loss in the worst case (perfect move of the opponent).

The algorithm used is Alpha-beta pruning, which is a search algorithm that decreases the visited nodes depending on the alpha-beta values. At first instance, we implemented the min-max algorithm, as it solves multiplayer game looking for minimizing the opponents game and maximizing your game. What minimax algorithm does is given a perfect information game (any player can make a move being perfectly informed of the previous events). The main difference between them is that in min max all the nodes need to be checked and if we are talking about the first game it will check more than 2500 nodes with 4 as depth, while alpha-beta pruning reduces the nodes visited by half.

The difference in time execution between min max and alpha-beta pruning is extremely different. Using the first algorithm the time execution is 1.09 seconds while in the other it only takes .049 seconds, both with depth 4

```
Computer's turn
0.0492959022522
```

```
Computer's turn
1.09271597862
```

The implementation first looks for a winning move, if a winning move ain't found, then it searches for the opponent's winning move, it neither of this moves exists then it goes with the alpha-beta pruning algorithm with a depth of 4.

Our implementation compared with others is not as good as others, we found "Connect four perfect solver" (http://connect4.gamesolver.org/), this implementation tells you how many moves are missing to lose because you just cannot win. We think that what makes this game that good is its heuristics. Our heuristic takes into account if the human

will win the game it returns -100000000 (-inf) otherwise it returns how many four, tree or two pieces in a row were found for the AI.

To improve the movements we tried to change again the heuristic by checking all the possible winning spaces, this means by checking if there were 3 pieces in a row connected, doesn't matter if there was a blank space (Ex. X-XXXO). But we obtained the same result, the behavior was exactly the same as with the last heuristic. This new heuristic can be found in the newHeuristic.py file.

The problem with both heuristics is that the utilities of some node are the same, and that is the reason why the movement goes to the first column, as it searches for the best move.

Run:

python connect4.py

Dependencies:

numpy

Twitter Analysis

Twitter is a social network service where users can tweet and chat through messages. Twitter was created in 2006 and by 2010 it already had more than 100 million users, in 2016 it had over 319 million active users posting 600,000,000 tweets per day.
We choose twitter for the quantity of data per day and for the type of posts it has, it is a social network in which people can express their feelings and sentiments more than in others. The goal was to given a twitter profile, classify the tweets in positive and negative and with this, we can have a personality analysis of the user.

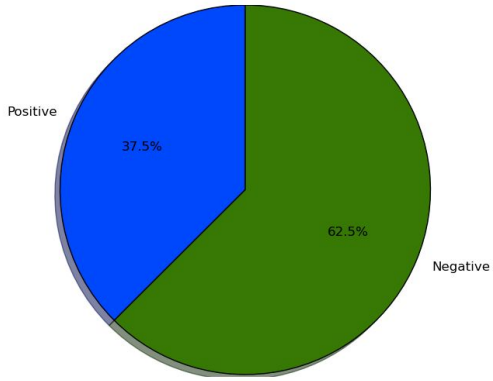
We thought a Twitter profile analysis was interesting as challenge because we think a lot of people show themselves in twitter, it can be a good way to know if someone is depressed or feeling lonely so he or she can have professional attention and fix their issues, with this in a more accurate and detailed way we can know when someone needs help and avoid unnecessary acts such as suicide or low self-esteem. (Dramatization)

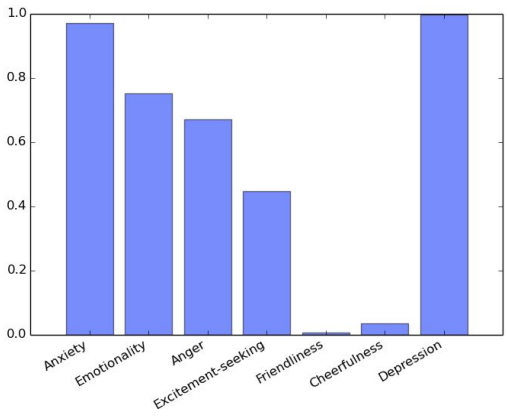
The biggest problem with using Twitter for sentiment analysis is the amount of sarcasm that some tweets can have and also because most of the tweets don't use a formal language, most of them use extra exclamation marks or not a proper use of punctuation.

Laura Daniela Vargas Herrera A01203961
María Fernanda Vargas Herrera A01203957

For the implementation, we used nklt (Natural Language Toolkit) to work with human language data (tweets). Given a set of training tweets, we can predict if a tweet tends to be more positive or negative. We also used the Personality Insight of Watson Developer Cloud to verify if the profile of the user tends to be more depressing or friendly. It basically receives all the tweets and extracts personality characteristics based on how a person writes. To make it more accurate you need at least 6,000 words for a meaningful analysis.

After analyzing the account we get a pie chart of the positive and negative tweets and also a bar chart with the personality insight, we are considering anxiety, emotionality, anger, excitement-seeking, friendliness, cheerfulness and depression.





For the training classifier, we used Naive Bayes Classifier, which is a popular method for text categorization, as it categorizes the document with word frequencies. Naive Bayes is a supervised machine learning, it is used to predict things based on its prior knowledge.

It took 20.08 minutes to classify the data (10000 records)

```
Accuracy: 73.6%
Data:
Training data 10000 records
Testing data 2000 records
```

Then we checked the accuracy by testing 2000 record with classified data, this took 5 minutes and we obtain a 73.6% accuracy.

We tried to improve the accuracy by two different ways:

Our first try was doing some research and we found that there were words with the same root such as happy, happiness and happily. We decided to use another module from nltk, Stemmers that remove morphological affixes from words. With this approach, we get an accuracy of 73.65%, a little bit higher than the actual.
And the second one was by increasing the training data to 20000 and testing to 3000 the final accuracy was 70.31%, lower than the actual

Run:

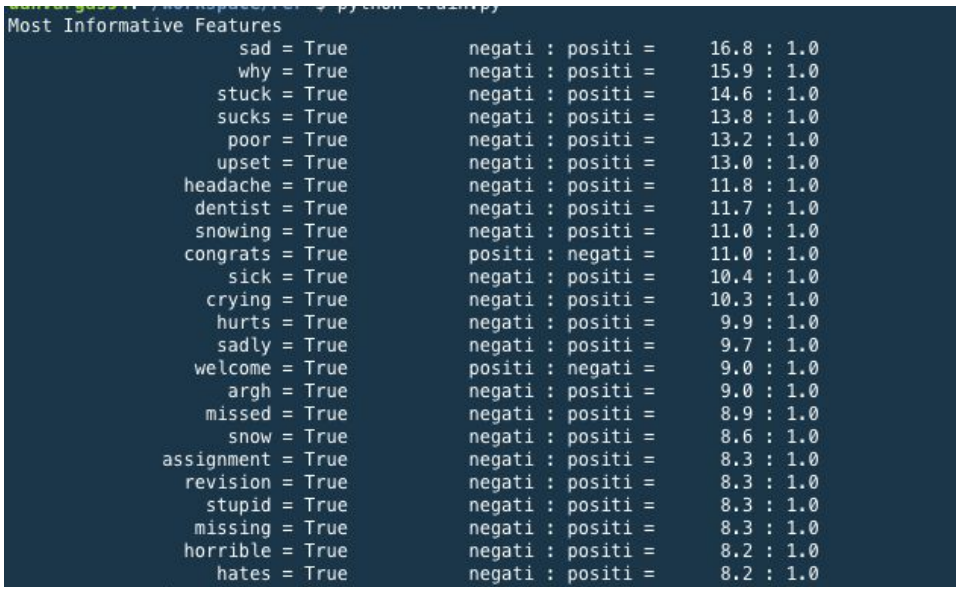
python csv.py

Given a CSV file it separates a training and testing dataset, we save 5000 positive and negative (10000 total) tweets for training and 1000 positive and negative (2000 total) tweets for testing. The data is saved in training.csv and testing.csv.

The labeled dataset was obtained from http://help.sentiment140.com/for-students/ it is the Google Drive link. The whole dataset was not added to the repository because it was very heavy.

python train.py

Given the training data, we create a Naive Bayes Classifier with nltk and print the 32 most informative features obtained from the tweets. We save the word features and the classifier in a pickle file.



```
Most Informative Features
                    sad = True           negati : positi =     16.8 : 1.0
                    why = True           negati : positi =     15.9 : 1.0
                  stuck = True           negati : positi =     14.6 : 1.0
                  sucks = True           negati : positi =     13.8 : 1.0
                   poor = True           negati : positi =     13.2 : 1.0
                  upset = True           negati : positi =     13.0 : 1.0
               headache = True           negati : positi =     11.8 : 1.0
                dentist = True           negati : positi =     11.7 : 1.0
                snowing = True           negati : positi =     11.0 : 1.0
               congrats = True           positi : negati =     11.0 : 1.0
                   sick = True           negati : positi =     10.4 : 1.0
                 crying = True           negati : positi =     10.3 : 1.0
                  hurts = True           negati : positi =      9.9 : 1.0
                  sadly = True           negati : positi =      9.7 : 1.0
                welcome = True           positi : negati =      9.0 : 1.0
                   argh = True           negati : positi =      9.0 : 1.0
                 missed = True           negati : positi =      8.9 : 1.0
                   snow = True           negati : positi =      8.6 : 1.0
             assignment = True           negati : positi =      8.3 : 1.0
               revision = True           negati : positi =      8.3 : 1.0
                 stupid = True           negati : positi =      8.3 : 1.0
                missing = True           negati : positi =      8.3 : 1.0
               horrible = True           negati : positi =      8.2 : 1.0
                  hates = True           negati : positi =      8.2 : 1.0
```

python test.py

Given the word features and classifier file ,we used them to test data and get the accuracy of the classifier.

python main.py @account

It receives as argument the account we want to analyze. We use a twitter connection to get the tweets of a user and predict the tweets are positive or negative. With this, we create a pie chart and also bar chart with Watson's Personality Insight.

Dependencies:
numpy
nltk
twitter
picke
matplotlib
watson_developer_cloud

Repo: https://github.com/fervargas94/AIChallenges

References:

Adam Pearce(s.f). Connect 4 AI: How it Works. Retrieved from: http://roadtolarissa.com/connect-4-ai-how-it-works/

Francisco Iacobelli. (June 22, 2015). minimax algorithm [Video]. Retrieved from: https://www.youtube.com/watch?v=6ELUvkSkCts&t=792s

Francisco Iacobelli. (June 22, 2015). alpha beta prunning [Video]. Retrieved from: https://www.youtube.com/watch?v=d2maa6k2gYE

Murphy, K. P. (2006, October 24). Naive Bayes classifiers. Naive Bayes Classifiers. Retrieved November 17, 2016, from https://datajobs.com/data-science-repo/Naive-Bayes-[Kevin-Murphy].pdf

Pascal Pons. Connect 4 perfect solver: Retrived from: http://connect4.gamesolver.org/?pos=

Personality Insights. IBM Watson Developer Cloud. Retrieved from: https://www.ibm.com/watson/developercloud/personality-insights/api/v2/

Sentex. (2015). Natural Language Processing With Python (1-21) . Retrieved from: NLTKhttps://www.youtube.com/watch?v=FLZvOKSCkxY&list=PLQVvvaa0QuDf2JswnfiGkliBInZnIC4HL

Sentiment 140. (n.d.). Retrieved November 14, 2016, from http://help.sentiment140.com/for-students/