Hunter Berry

Dr. Hamid

HON 450

February 12, 2020

Annotated Bibliography

Brownlee, Jason. “Naive Bayes Classifier From Scratch in Python.” *Machine Learning Mastery*, 24 Oct. 2019, machinelearningmastery.com/naive-bayes-classifier-scratch-python/.

* This source is a website that provides online tutorials for topics relating to machine learning and analysis. The website is run and upkept by a professor with a PhD in Machine Learning, which leads me to believe that the source as a whole is refutable and that the tutorials provided are accurate representations of real-word algorithms and programs. This specific article details information on how to create a Naïve Bayes Classification system in Python, which we are currently looking at as a possible way to identify the likelihood of certain words being connected to certain areas and locations and/or certain political parties or ideological levels. I used this article to determine a way to combine my current code with the basis of the NBC to create a function that would determine the likelihood of any word being connected to a party/location given some dictionary containing the frequency of the words and their given usages.

Dandrea, Alessia, et al. “Approaches, Tools and Applications for Sentiment Analysis

Implementation.” *International Journal of Computer Applications*, vol. 125, no. 3, 2015,

pp. 26–33., doi:10.5120/ijca2015905866.

* Overall, this article is published by a number of professors and researchers working in conjunction with Italy’s Institute for Research on Population and Social Policies and the National Research Council. The organization and the authors of the paper all seem to be refutable, with decades of experience in the fields of machine learning, sentiment analysis, language acquisition and analysis, and other similar areas. The source is a scholarly article that functions as an overview and an introduction to sentiment analysis for people attempting to learn more about the field. The article is broken down into a few different sections, two of which are specifically of interest for the purposes of this project, those sections being sentence-based sentiment analysis and aspect-based sentiment analysis. The article goes into detail about methods that are used in these studies, and the advantages and limitations of different approaches such as machine learning based analysis, lexicon-based analysis, and hybrid analysis. The article also includes a number of tools and resources that are helpful to people getting into the field of sentiment analysis. Overall, the article was extremely helpful to me in helping me understand some of the background knowledge required to do well in sentiment analysis, and understand how I can possibly shift my project if needed from a solely machine-learning based approach to one that also includes some lexicon analysis. This source is something that, like the textbook for this course, I know I can refer back to from time to time simply to refresh my memory on the topic and also look for new approaches and tools that can help me improve my software.

GovTrack.us. 2013. Ideology Analysis of Members of Congress. Accessed at <https://www.govtrack.us/about/analysis>.

* This source is an organization that tracks members of Congress and their legislative actions, including information about voting on bills and Congressional actions, committee meetings and agendas, the voting patterns and ideologies of Congressional members, and much more. For the purposes of this project, I used the source to learn about some possible ideological analysis algorithms and methods. The website details how the organization uses voting patterns and comparative measures to rank each politician on a scale of 1.0 (Republican/Conservative) to 0 (Democrat/Liberal). They also include their source code and extensive documentation describing how to replicate the process and include it in a Python project. Overall, I believe the source is useful because it provides me with a possible method of ranking ideology on a more continuous scale rather than a dichotomous one, which could help when using any type of regression-based machine learning model, like the XGBRegressor that I’ve looked at using for a later part of the project.

Hasan, Ali, et al. “Machine Learning-Based Sentiment Analysis for Twitter Accounts.” *Mathematical and Computational Applications*, vol. 23, no. 1, 2018, p. 11., doi:10.3390/mca23010011.

* This source was published by a number of PhD students and professors with expertise in the field of machine learning and sentiment analysis. This article goes into more detail about the hybrid approach to sentiment analysis that was briefly covered in Dandrea’s article Approaches, Tools and Applications for Sentiment Analysis Implementation.” The article goes into depth about the use of hybrid approaches that take machine learning models and algorithms, such as Naïve Bayes or support vector machines (SVM), and combine them with lexicon analysis to come up with a ‘state of the art’ and ‘standardized’ practice that can be used across multiple industries and ‘genres’ of development. While this information is helpful, the part of the article that is most beneficial is its discussion on the connection between this hybrid approach to sentiment analysis and its use on Twitter, such as how sentiment analysis can make use of things like hashtags, retweets, and other Twitter-specific types of data. Thus, the article has given me some ideas that I’d like to look into further for the purposes of my project, such as the use of hashtags in determining sentiment, because some hashtags do offer extensive information about the accounts that are using them. Overall, I hope to use this article and some of the tools, libraries, and ideas mentioned within it to further build my project, and also use it as a source to refer back to throughout the course of this project.

Liu, Bing. *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge University Press, 2017.

* Overall, this source is a book functioning as my textbook for this course. The book is written by Bing Liu, a computer science professor with a PhD that specializes in data mining, machine learning, and natural language processing. Dr. Liu has worked in the field for years, both inside and outside of academia, and has also written numerous books on the topic, proving that he is a refutable source with extensive knowledge in the fields. Overall, the book provides a plethora of different types of information, ranging from covering basic algorithms that may be used in aspect based sentiment analysis (like Hidden Markov Models and CRFs) to discussing the data analytics and filtering required to ensure that our data is clean and usable within our models. The book also goes into more detail on some of the topics covered by other sources, such as the “corpus” and “body of words” approach used to associate words and sentiments. Although I have not finished the book yet, the information I have read so far has been extremely insightful and helpful in allowing me to understand the actual processes and theories that are behind the idea of sentiment analysis and actually begin working on my code. As a result, this source is an invaluable source within my project.

“Members of the U.S. Congress.” *Congress.gov*, U.S. Congress, Feb. 2020, [www.congress.gov/members?searchResultViewType=expanded&KWICView=false](http://www.congress.gov/members?searchResultViewType=expanded&KWICView=false).

* Overall, this source is simply a government record containing information about all of the Congressmen and Congresswomen that are currently serving in either the Senate or the House of Representatives. It contains the state that each member currently represents, and the party that each member is a part of. I used this source to help me categorize each member and begin forming some dictionaries that would allow me to begin tracking things like which parts of speech were used most often used my members of Congress in a certain area, or of a certain party, something I hope to extend to help analyze normal citizens.

Paul, Sayak. “Simplifying Sentiment Analysis in Python.” *DataCamp Community*, DataCamp, 7 Jan. 2020, [www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-](http://www.datacamp.com/community/tutorials/simplifying-sentiment-analysis-)

Python

* This source is a tutorial series on DataCamp, a website offering both free and paid courses in data science, statistics, machine learning, and deep learning, through videos, coding assignments, articles, and more. The author of the article, Sayak Paul, seems to be experienced in the areas of deep learning as it relates to Python, as is evident by his work with DataCamp and his job doing deep learning and computer vision. Overall, this multi-part tutorial series goes over the basics of sentiment analysis and some of the key concepts, such as the “bag-of-words” (or what I simply refer to as my global dictionaries) and the use of classifiers like a Naïve Bayes Classifier and how it can work with the NLTK. Furthermore, this tutorial also gave me some ways that I can go improve my current code, such as through the use of the NLTK’s frequency distribution functions, which may work a little cleaner than my dictionaries. Overall, this source was extremely helpful in teaching me some of the basics of sentiment analysis and has given me a good starting point to keep working on my code and some of my basic classifiers.

“Regression Example with XGBRegressor in Python.” *Data Tech Notes*, 26 June 2019, [www.datatechnotes.com/2019/06/regression-example-with-xgbregressor-in.html](http://www.datatechnotes.com/2019/06/regression-example-with-xgbregressor-in.html).

* This source is run by a professional software developer and functions as a blog where the author writes about data science and analytics, machine learning, and deep learning, with a specific focus on Python. Despite it being a blog, I believe the source is still usable and refutable since it comes from an experienced software engineer with work experience in the field. Overall, I use the source to simply give me a little more information on the XGBRegressor machine learning model, a type of gradient boosting supervised learning problem model that according to most sources is one of the easiest yet most powerful models that can be used in machine learning. Since I have thought about using the XGBRegressor in my project at a later point to either compliment the Naïve Bayes classifier and boost it, or possibly use it for a different purpose, such as using it as the actual ideological identification/prediction model. Regardless, the source provided me some good information on how the model works, and also provided some starter code to use in Python, which provides me a method of incorporating the model in my project at a later date if needed.

Roesslein, Joshua. “Tweepy: Twitter for Python!” *GitHub*, GitHub, 19 Jan. 2020, github.com/tweepy/tweepy.

* This source is another Python library that I used for interacting with the Twitter API and developer account that I set up. This library provides a simple way to interact with the API. Through some of the functions and methods featured in the library, it is possible to handle many complex things like rate limiting, tweet classification, and more, in a simply, clean, and repeatable way, things I personally did not find prevalent when using the Twitter API by itself. The library, like the other libraries mentioned, is open source, and has extensive documentation online, which has and will allow me to interact with the API on a variety of levels, including getting the location a tweet was published from, information about the user who tweeted, and much more.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

* This source is a Python library for scientific data analysis. It contains a variety a resources, including dozens of machine learning models, including classification and regression based models, both of which I will most likely use at some point within the process of creating my software to help determine ideology, sentiment, the likelihood of certain terms being used, and much more. Furthermore, the website includes detailed documentation on the models, as well as the source code for the project, so that anything can be edited if needed. As a result of these features and extensive documentation, the library will be of immense help to me throughout the remainder of this project.

Sloria. “TextBlob: Simplified Text Processing.” *GitHub*, GitHub, 15 Jan. 2020, github.com/sloria/TextBlob/.

* This source is another Python library which acts as an extension of the Natural Language Toolkit (NLTK), a Python library with extensive functionality for a plethora of language-based problems. The TextBlob library simplifies the NLTK into a library that is much more user friendly, while retaining the same functionality and power offered by the NLTK. The library includes a variety of functions that will be useful and relevant to the project at hand, including a built-in sentiment and polarity analyzer, functions to break apart sentences into various parts of speech such as noun and verb phrases, and much more. The library, like the others mentioned, is also open source, and is extensively documented, allowing me to easily determine what functions and methods to use within the context of this project. Currently, I’ve used this library to begin breaking each tweet down into important noun phrases and use those to begin doing some basic analysis with a Naïve Bayes Classifier. I’ve also used this library to set up a simple sentiment analyzer, which I hope to build on and eventually transform into an aspect-based sentiment analyzer.