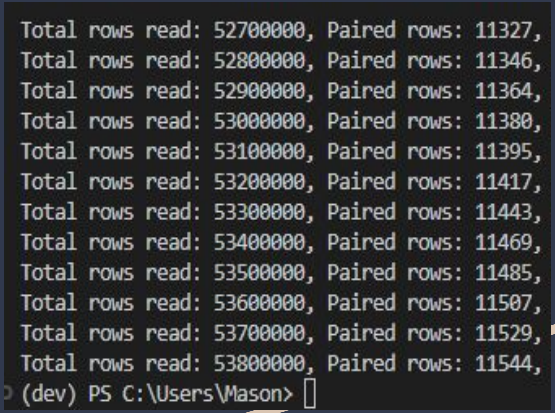
Building a ChatBot

With Mason, Steven, and Ted

StockBot

- We decided to go with a stock based chatbot as the topic for our project.
- The goal was to create a ChatBot using NLP machine learning that you could answer any basic questions regarding the stock market and it give you the proper response based off user Input.
- The bot would parse user input and with the use of the Alpha Vantage API, it would give details on a specific ticker.
- Our “Stock Bot” operates in a HTML webpage with the theme of a chatroom.

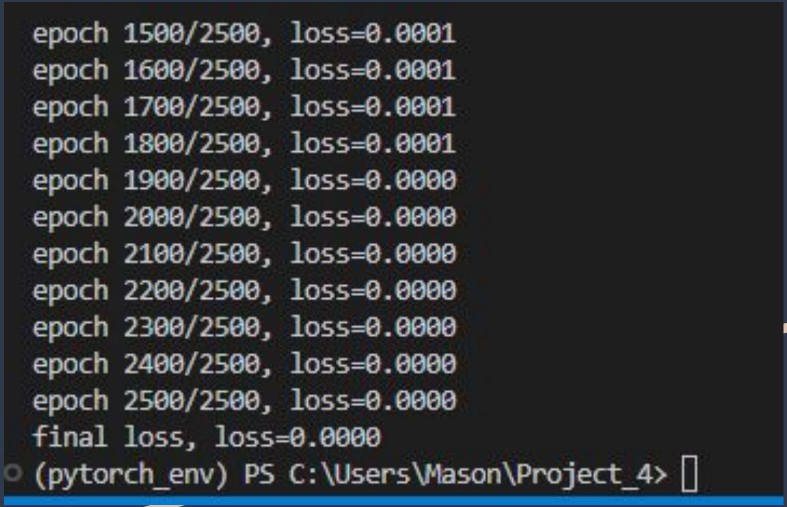
Gathering and cleaning the Data



```
Total rows read: 52700000, Paired rows: 11327,
Total rows read: 52800000, Paired rows: 11346,
Total rows read: 52900000, Paired rows: 11364,
Total rows read: 53000000, Paired rows: 11380,
Total rows read: 53100000, Paired rows: 11395,
Total rows read: 53200000, Paired rows: 11417,
Total rows read: 53300000, Paired rows: 11443,
Total rows read: 53400000, Paired rows: 11469,
Total rows read: 53500000, Paired rows: 11485,
Total rows read: 53600000, Paired rows: 11507,
Total rows read: 53700000, Paired rows: 11529,
Total rows read: 53800000, Paired rows: 11544,
(dev) PS C:\Users\Mason>
```

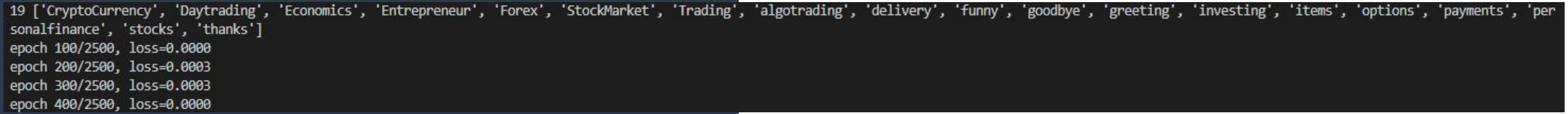
- The dataset that we used is a collection of all public comments and replies posted on Reddit from 2007-2015.
- Raw Data was received in JSON format. (Over 51GB of text!)
- Data parsed to only take subreddits that pertain to stocks and finance (I.E. r/Economics, r/Stocks) and was stored in an SQLite database.
- To train our model how to “talk” we made sure to take user comments and their replies, sorting them from highest scoring to lowest.This essentially shows our bot what a conversation looks like and can respond accordingly.

Machine Learning and NLP



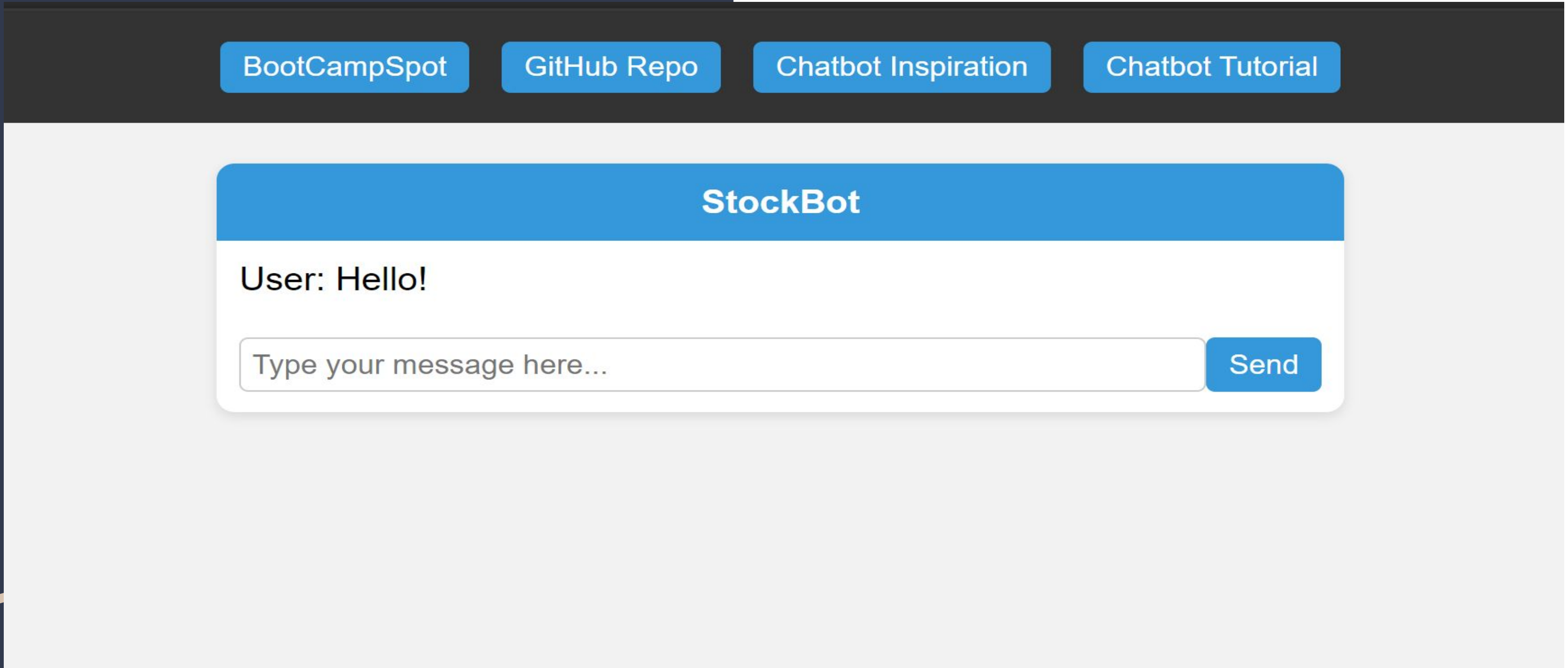
```
epoch 1500/2500, loss=0.0001
epoch 1600/2500, loss=0.0001
epoch 1700/2500, loss=0.0001
epoch 1800/2500, loss=0.0001
epoch 1900/2500, loss=0.0000
epoch 2000/2500, loss=0.0000
epoch 2100/2500, loss=0.0000
epoch 2200/2500, loss=0.0000
epoch 2300/2500, loss=0.0000
epoch 2400/2500, loss=0.0000
epoch 2500/2500, loss=0.0000
final loss, loss=0.0000
○ (pytorch_env) PS C:\Users\Mason\Project_4> 
```

- Used Pytorch for ease of use, as it was a fairly user friendly machine learning framework. Pytorch utilizes Nvidia's CUDA cores within their GPU as opposed to the standard training through CPU. This decreases time to train.
- NLTK was used to tokenize, stem, and attribute a value to our words. This helps the computer determine a response based off user response.
- Scikit-Learn was used for Test, Train, Split with parent comments as our features and replies as our labels.
- Training took 2.5 hrs with an average loss of 0.000,
- Our chosen model was Neural Network. This has an average F1 accuracy score was 78.5437. This could've been attributed to the abundance of talk that's all too familiar with reddit. This includes things like slang and URLs that our model may have considered as actual words



```
19 ['CryptoCurrency', 'Daytrading', 'Economics', 'Entrepreneur', 'Forex', 'StockMarket', 'Trading', 'algotrading', 'delivery', 'funny', 'goodbye', 'greeting', 'investing', 'items', 'options', 'payments', 'per
sonalfinance', 'stocks', 'thanks']
epoch 100/2500, loss=0.0000
epoch 200/2500, loss=0.0003
epoch 300/2500, loss=0.0003
epoch 400/2500, loss=0.0000
```

Website



StockBot

User: Hello!

Conclusion

Although we were not able achieve the desired results that we wanted for our project, we did make a good headway. While we were still able to get the results from training the ChatBot did not as the ChatBot replies incoherently. This may be attributed to the actual input data as we did use Reddit comments.