Predicting Bull Bear Stock Market with Billboard Top Hits

Yu Yao Hsu yh832@nyu.edu

Tina Yuan thy258@nyu.edu

Abstract

People's moods influence their decision making and music can often play the role of reflecting people's inner emotions. A study suggested that people tend to consume lighter media or more cheerful tunes during recessive periods. We want to learn if the weekly top hits, which may act as a reflection of public sentiment at that time, are correlated with fluctuations in the S&P500 index. In addition, we want to see which attributes of a song such as tempo, mode, energy etc. have predictive power of changes in the stock market. Ultimately, we would like to answer the question: what type of music is the most popular during a bull/bear market. Weekly Billboard chart and stock market data are collected from July 1999 to July 2019. A *Long Short-Term Memory* model is trained on the song attributes to predict future stock market changes.

Keywords: Predictive Analytics, LSTM, RNN

1. Introduction

People's moods influence their decision making and music can often play the role of reflecting people's inner emotions. A study suggested that people tend to consume lighter media or more cheerful tunes during recessive periods. The study investigated the relationship between music key signature and the economy and suggested that the economy was stronger when minor key signature music became top hits(Hershfield and Alter, 2019). Study also has shown that public sentiments can cause emotional fluctuations in investors and intervene in their decision making(Li et al., 2014). Pleasant mood can be an indicator of increase in stock prices, while unpleasant mood can be an indicator of decrease in stock prices as shown in Cohen-Charash Y (2013)'s work. Hershfield and Alter (2019)'s work prompted us to further investigate the relationship between hit music and the economy. We want to learn if the weekly top hits, which may act as a reflection of public sentiment at that time, are correlated with fluctuations in the S&P500 index. In addition, we want to see which attributes of a song such as tempo, mode, energy etc. have predictive power of changes in the stock market. Per Hershfield and Alter (2019)'s study, we hypothesize that a song with faster tempo and performed in a major key should correlate to a bear market. Ultimately, we would like to answer the question: what type of music is the most popular during a bull/bear market. For the model since we aim to predict time series data, as suggested in many research such as Nelson et al. (2017) and Chen et al. (2015)'s work, we will be using LSTM model. LSTM models are capable of associating memories and input remote in time and use them for future prediction (Chen et al., 2015). It also aims to solve the vanishing gradient issue that recurrent networks would suffer when dealing with long data sequences(Nelson et al., 2017).

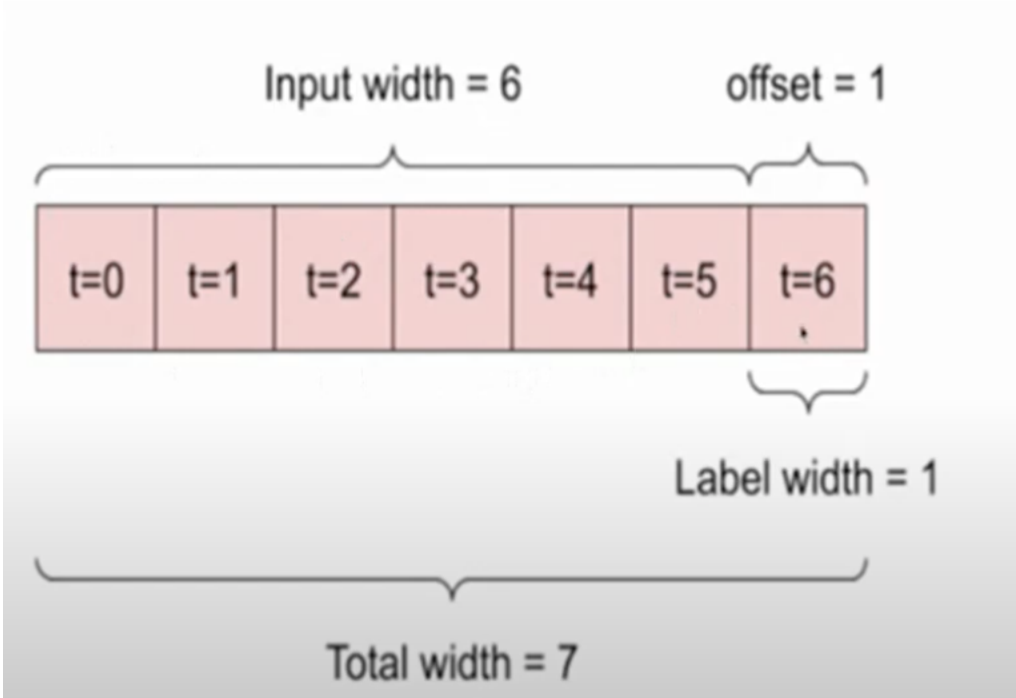

Figure 1: Example of input matrix with history length of 6 and predicting the following week's label

2. Data and Methods

2.1 Data Collection

Hit songs data is collected from the *Billboard Hot 100 Chart.* We chose the Hot 100 chart as the data source because it aggregates data from the U.S. music retail market, several online streaming platforms, and radio airplay data (Bil). The fact that the chart uses multiple channels to determine song rankings suggests that the rankings are likely to be reflective of different groups of people who listen to music using different channels. There is an existing data set on Kaggle that includes the Hot 100 charts from July 1999 to July 2019 (please see appendix for link to data set). We downloaded the data set from Kaggle to use for the project. In addition to weekly charts data, the same data set on Kaggle includes song attributes data. The song attributes include tempo, mode (major or minor key), energy, and many other values that relate to a song's musicality. We believe that these features may be indicative of public sentiments and therefore be correlated to the economy.

In terms of stock market data, we collected daily closing values of the S&P500 index from Yahoo finance using the python library beautifulsoup4. The S&P500 index tracks the stocks of 500 large-cap U.S. companies and is therefore reflective of the U.S. economy as a whole.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 128)               73728
_________________________________________________________________
dropout (Dropout)            (None, 128)               0
_________________________________________________________________
dense (Dense)                (None, 100)               12900
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 202
=================================================================
Total params: 86,830
Trainable params: 86,830
Non-trainable params: 0
```

Figure 2: A simple model with one LSTM layer

2.2 Data Preparation

Once the data is collected, we first merged the Billboard chart data with the song attributes data to create a single table of all the input features we would use to predict the changes in the S&P500 index. After merging the data, we deleted any rows that have null values in its columns. At first we wanted to create a weekly composite score of all the songs that are in the chart and use the composite as the input vector to the model. However, we couldn't find the most appropriate way to combine different song's tempo, energy, and other attributes into a single value. Consequently, we decided to use the attributes of the top ranking song of that week as the input vector.

Since we only have weekly data for the input, we decided to extract weekly data for the output as well. We kept only the Monday or the first market open day value of the S&P500 index for each week. Week n is assigned a label of 1 if week n's value is greater than week $(n+1)$'s value. Similarly, week n is assigned a label of 0 if week n's value is less than week $(n+1)$'s value.

Our final data matrix consists of the weekly top ranking song and its attribute and the weekly S&P500 index change labels. For the input matrix to the neural network, we will be testing it out with different length of history data, and the label will be the following row of label (See 1).

2.3 Model

The data we collected is a time series so we wanted to build a recurrent neural network (RNN) that captures the characteristics of previous time steps' data and use that information with the new input data to predict future outputs. We decided to build a Long

Short-Term Memory (LSTM) model for time series classification. The use of a LSTM model would hopefully help avoid gradient-related problems that are commonly seen in RNN models.

We defined the model as having a single LSTM hidden layer. This is followed by a dropout layer intended to reduce overfitting of the model to the training data. Finally, a dense fully connected layer with ReLU activation function is used to interpret the features extracted by the LSTM hidden layer, before a final output layer with sigmoid activation function is used to make predictions (See 2). We compiled our model with the binary cross entropy as our loss function, and adam as our optimizer.

test size	**time step**	**loss**	**accuracy**
0.2	4	0.6746	0.5980
0.2	6	0.6756	0.5990
0.2	8	0.6777	0.6000
0.2	10	0.6751	0.6010
0.2	12	0.6701	0.6071
0.3	4	0.6777	0.5974
0.3	6	0.6744	0.6111
0.3	8	0.6739	0.6053
0.3	10	0.6734	0.6093
0.3	12	0.6760	0.6033
0.4	4	0.6789	0.5947
0.4	6	0.6806	0.5976
0.4	8	0.6827	0.5980
0.4	10	0.6774	0.5862
0.4	12	0.6865	0.5990
0.5	4	0.6883	0.5911
0.5	6	0.6805	0.5895
0.5	8	0.6815	0.5898
0.5	10	0.6810	0.5902
0.5	12	0.6838	0.5886
0.6	4	0.6859	0.5726
0.6	6	0.6855	0.5744
0.6	8	0.6828	0.5747
0.6	10	0.6891	0.5651
0.6	12	0.6855	0.5735
0.7	4	0.6890	0.5718
0.7	6	0.6897	0.5831
0.7	8	0.6891	0.5736
0.7	10	0.6864	0.5724
0.7	12	0.6917	0.5433

Table 1: Results using 3 LSTM layers and ReLu activation function

dropout	neurons	time step	regularizer
0.0, 0.2, 0.4, 0.6	64, 128, 256	4,6,8,10,12	L1, L2, L1L2

Table 2: Parameter values tested with 3-layer LSTM model

activation function	regularization type	dropout	neurons	time step	accuracy
tanh	no	0.6	256	8	0.6200
tanh	bias-L1L2	0.4	256	12	0.6200
tanh	input-L2	0.4	256	10	0.6233
tanh	recurrent-L2	0.4	256	10	0.6233
ReLU	no	0.6	256	10	0.6192
ReLU	bias-L2	0.6	256	6	0.6111
ReLU	input-L1	0.4	256	6	0.6111
ReLU	recurrent-L1L2	0.2	256	10	0.6159

Table 3: The best parameter values for 3-layer LSTM model with test size = 0.3

3. Results

While training and testing the model, we discovered that running the model on different platforms led to different results. Our model achieved approximately 60% accuracy running on the ARM architecture with TensorFlow 2.4.0; running on the x86 architecture with TensorFlow 2.5.3, the maximum accuracy achieved was around 40%. Therefore, results reported would be those ran on the ARM system with TensorFlow 2.4.0. We also tried several parameter adjustments to see if we can increase model accuracy.

First, we adjusted the length of history data that we read in. We tried learning the past 4, 6, 8, 10, 12 weeks of data to predict the following week's stock market index. Yet, the different time steps yielded similar results.

Next, we adjusted the portion of our training and testing size. We changed the test size from 20%, 30% all the way up to 70% of our total data. The results showed that the model has the highest accuracy when the test size is 30%. However, the accuracy is just slightly better than 60%.

We then turned our focus on adjusting the model. Currently, our model is a simple LSTM model with one hidden LSTM layer followed by a dense layer and an output layer. We tried to add more LSTM layers to test if a more complex model can lead to better accuracy on the prediction. The results turned out to be the same with around 60% accuracy (See 1).

Finally, we adjusted the parameters of the different layers. We adjusted the drop out rate from the drop out layer and also added regularization to hidden layers to reduce overfitting. We increased and decreased the neurons for the hidden layers. For the activation function, besides using ReLU, we also tried using tanh. However, all these parameter changes still gave us the same accuracy of around 60%.

To verify which parameter values give the best results, we tested a 3-layer LSTM model with a test size of 0.3. Using the tanh activation function with no regularization, bias regularization, input regularization, and recurrent regularization, we changed the drop out rate, neurons, and time step to find the best combination for each tanh-regularization combination. The same set up was used with the ReLu activation function with no regularization, bias regularization, input regularization, and recurrent regularization. The results are not that great and suggest that there may not be a causal relationship between hit music and the stock market (See 2 and 3).

4. Conclusion and Discussion

While the results did not turn out to be desirable, there are many ways to improve the study to further investigate the causal relationship between music and the stock market. Currently, the model is trained using a single top-ranking song for each week. The results could be improved by considering the top 20, or even all 100, songs in the Hot 100 charts when generating the feature vector for a single time step. The main challenge with this approach is to figure out how to calculate the composite value for each song attribute. Simply adding all the values together would be inappropriate because the summed values cannot represent all the songs. Another approach would be to have every single song and its attributes represent one column. A single song has 15 attributes (and thus 15 columns) so if we use 100 songs, the matrix would have 1500 columns; this approach would lead to an input matrix with many columns, which is also not desirable.

Additional features can also be added to help increase model accuracy. The data set does include information about each song's genre and lyrics. By performing sentiment analysis on the lyrics, we may acquire a sentiment value for the song that can be used as another feature when training the model. A song's genre can also be beneficial as some genres are typically associated with lighter (in terms of mood) songs and other genres with heavier songs. However, such assumptions about a song's genre and a song's cheerfulness would need to be grounded before the song's genre can be included as a useful feature.

Besides including more songs and relevant features to potentially help increase the accuracy of prediction, we also need to find a way to fill in missing values and to eliminate redundancy in the data. The data collected is not raw data. The data has already been prepossessed to generate song attributes such as acousticness, energy, or valence. When we merged the Billboard top hit data with song attributes data, not all of the songs in the Billboard data set have record in the song attributes data. If we do not have the raw data and a method of generating the song attributes, we would not be able to get the values for our missing data. This leads to the problem where the full list of top 100 songs cannot be used as the input for some weeks. Another problem we faced during data processing is that oftentimes the top songs for the current week will still be the top songs for the following weeks. Thus, when selecting only one highest ranking song to represent the week, we often use the same song many weeks in a row. Our final input matrix turned out to have a lot of duplicate songs, which does not help our model learn.

We believe that the reason for low model accuracy is mainly because the data that we prepared is not that useful. If we were able to solve the data processing problems discussed above, the accuracy of the model could be much higher. Finally, once the data processing problems are cleared, future studies can use a more complex model to explore the relationship between top hit Billboard songs and the stock market to achieve better results. The studies can also investigate if the reverse relationship has any causal effects.

References

Billboard charts legend. URL `https://www.billboard.com/p/billboard-charts-legend`.

Kai Chen, Yi Zhou, and Fangyan Dai. A lstm-based method for stock returns prediction: A case study of china stock market. *IEEE international conference on big data (big data). IEEE*, 2015.

Kammeyer-Mueller JD Staw BM Cohen-Charash Y, Scherbaum CA. Mood and the market: can press reports of investors' mood predict stock prices? *PloS one*, 8.8:e72031, 2013.

H. E. Hershfield and A. L. Alter. On the naturalistic relationship between mood and entertainment choice. *Journal of Experimental Psychology: Applied*, 25(3):458–476, 2019.

Qing Li, TieJun Wang, Ping Li, Ling Liu, Qixu Gong, and Yuanzhu Chen. The effect of news and public mood on stock movements. *Information Sciences*, 278:826–840, 2014.

David MQ Nelson, Adriano CM Pereira, and Renato A. de Oliveira. Stock market's price movement prediction with lstm neural networks. *International joint conference on neural networks (IJCNN). IEEE*, 2017.

Appendix A. Kaggle Data Set

The Billboard Hot 100 and Song Attributes data are downloaded from https://www.kaggle.com/danield2255/data-on-songs-from-billboard-19992019. Files used are "billboardHot100_1999-2019.csv" and "songAttributes_1999-2019.csv".

Appendix B. Code

Please see Google Drive for code and input data set. https://drive.google.com/drive/folders/1SkjXzWvOgdiELRFKHzfbssHbvIiYnY7Q?usp=sharing