ERAU

Reddit Regression

CS 595

Justin Duhaime and Kaitlyn Leta
4-27-2020

1 Introduction

Reddit is a web based social media platform where users submit creative content (posts). Other users on the site can then upvote or downvote the content at their leisure. The posts that receive more upvotes appear higher on the site where the posts that receive more downvotes are lower. This creates a feed of content that is community driven where only the collective best posts are seen by the majority. When a user makes a post that gets voted on, they are awarded 'karma' or points toward their accounts. For this project, we are assuming a 1:1 ratio between composite post score (upvotes minus downvotes) and karma. Our project attempts to use past post data to predict the optimal time of day to post your content to get the highest composite score. We gathered this data from Pushshift.io [2] where we could analyze the Reddit posts and scores.

The main goal of our problem is to solve when the best post time is to gain the most positive attention to the post. This can be useful for businesses since social media is mostly used for advertisement of their product or work. Knowing what times to post on social media to gain the most attention and followers is a huge advantage to increase awareness and/or sales. Many companies struggle with their social media platforms, mostly due to lack of a media team and skilled graphic design. However, one huge aspect to social media is when to post and who is the target audience. In this project we aim to predict the best post times to help solve part of the social media issue. In order to do our analysis, we needed to start with an approach that involves comparing multiple algorithms to find our best fit for our data.

Our overall approach was to gather Reddit data and pre-process it for our models. Once we filtered the data to a point where we could analyze it, we applied three different approaches. These approaches were linear, logistic and polynomial regression. After we analyzed the results from those models, we decided to do some additional analysis on the data to see what else we can get to determine the optimal time to post by splitting the data into time blocks to see which block would have the highest mean score. This additional analysis also included filtering the data down to one theme of subreddits related to 'meme'. After all the analysis was concluded, we were able to see at what times it would be optimal to post on Reddit from various approaches. Therefore, we believe that our approach was a success.

2 Related Work

In researching this topic, we came across a similar study performed by Andrei Terentiev and Alanna Tempest from Stanford University [1]. Their project focused on predicting how well a Reddit post would do (regarding its overall score) depending on how many comments the post had relative to its age. They first set a threshold value to test whether the post would surpass it or

not. Their conclusion stated that “[r]eddit post popularity definitely has an element of randomness but using characteristics of its early comments, a post’s popularity with respect to a fixed threshold can be predicted with up to 89% accuracy.” [1]. These results coincide with ours as we were able to demonstrate a clear correlation between the time a post was made and the relative score the post received.

3 Approach

Our first step in the process was to find a data source that would give us a data-dump of posts for a given time frame. Luckily, there is a public website that provides zip files of all Reddit posts broken up by month and year of posting. We chose the most recent data source on the website at the time which was August 2019. The data in this single file totaled to about ~120 Gb of data. We knew this would be too much and would skew the algorithms we planned to run. The first round of data sanitization was to remove posts that had a negative score or a score lower than a set threshold. We did not care about posts that had less than 100 score, nor did we care about posts that were in adult only subreddits. After trimming these scores, we still had ~3 million posts worth of data. The next trim was to only use posts from a single day in a 24-hour window. Once this was done, we had a more manageable dataset, but our files were still extremely large. We noticed in the json object of the posts that about 90% of the data in the file was never going to be used. The only datapoints we care about are score, time of post and subreddit community it was posted to. The final data sanitization that took place was to limit to just one post per time value. After noticing that there were multiple (> 1000) posts made at any given time (hour:minute), we decided to only choose 1 value at each time. This brought the dataset down to 1,440 records, or one for every minute in a day.

The first algorithm we decided to use was linear regression. We could see from the data plot of the dataset that there were more clusters of data points with lower scores towards the bottom of the plot. This helped with our evaluation of the model. In order to implement the linear regression algorithm, we used the sklearn import function. Then we split the data and used a test size of 10 and random state equal to zero. Then fit the data accordingly. Once we plotted the data, we also ran the prediction function with X_test and showed the results for the actual and predicted values.

The next algorithm we used was logistic regression. We essentially followed the same approach for linear regression where we used sklearn for the logistic function, split the data with test size of 10 and random state as zero, and fit the data. Once we plotted the data and got our results, we then ran the predict function and displayed the results for the actual and predicted values.

The last algorithm we chose to use was polynomial regression. For polynomial regression, we used PolynomialFeatures from sklearn. We used the PolynomialFeatures to add extra features to

the linear regression function from sklearn and then trained it. Once we were done, we used the predict function and plotted the results.

Source File	Description
san.py	Python script to sanitize the raw Reddit data
Project_Reddit.ipynb	Jupyter Notebook containing the ML algorithms and results
README.md	Contains author and file information

4 Evaluation and Results

Metrics

The time we use to display in our plots is in a 24-hour format. (0 is 12:00Am EST)

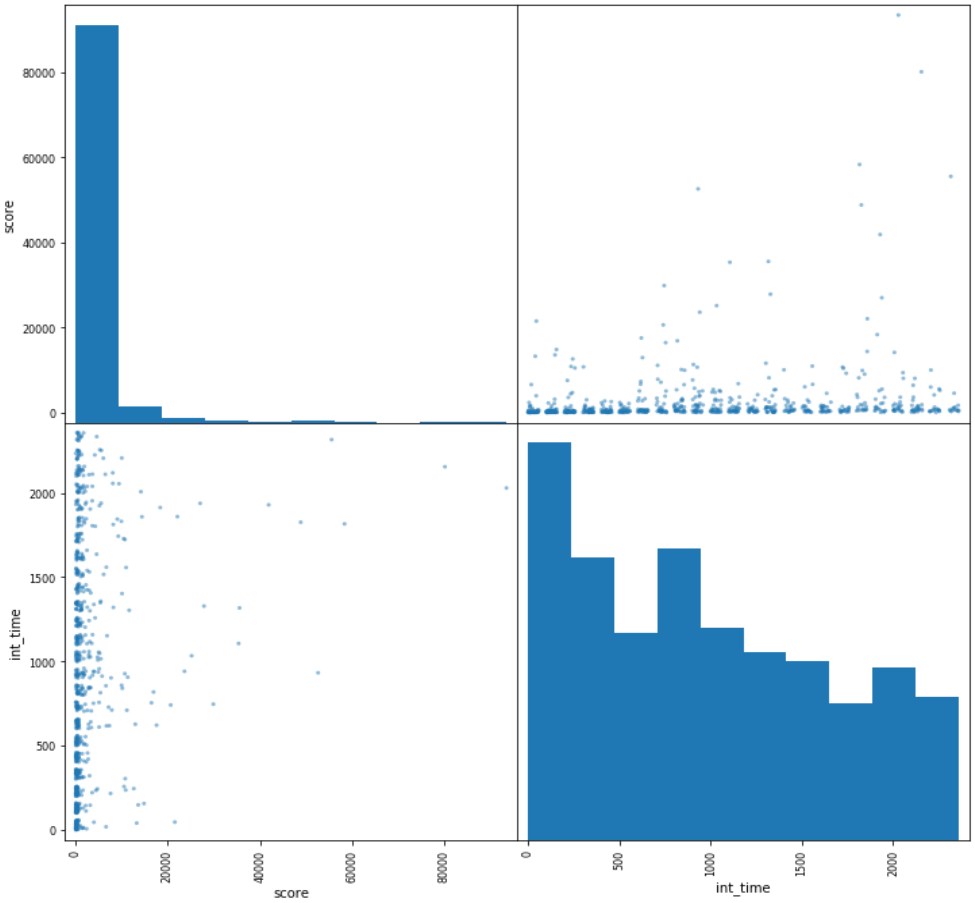

Figure 1: Scatter Matrix

4a Linear Regression

For linear regression we found that the model was a little underwhelming from the results. However, in figure 2 below you can see that it did follow a similar pattern of increasing in score as the time of day passed. This becomes useful as we can see some similarity of score and time and that it correlates with our conclusions. Linear regression approach was helpful, but there were better approaches to solve our problem.

When trying to plot the ROC curve or precision and recall values, we got multiclass errors that we could not solve for the linear and logistic regression. However, we were able to plot the learning curves for linear and polynomial regression. For future reference, we could improve this by spending more time understanding how the data is being translated and fit so that we know what the error was and how to fix it.

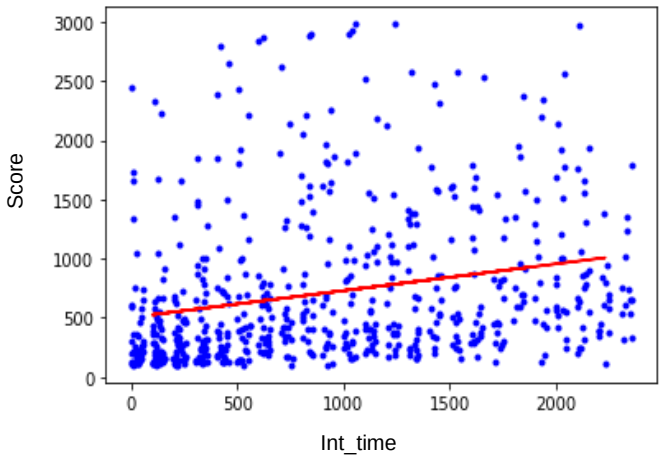

Figure 2: Linear regression on data

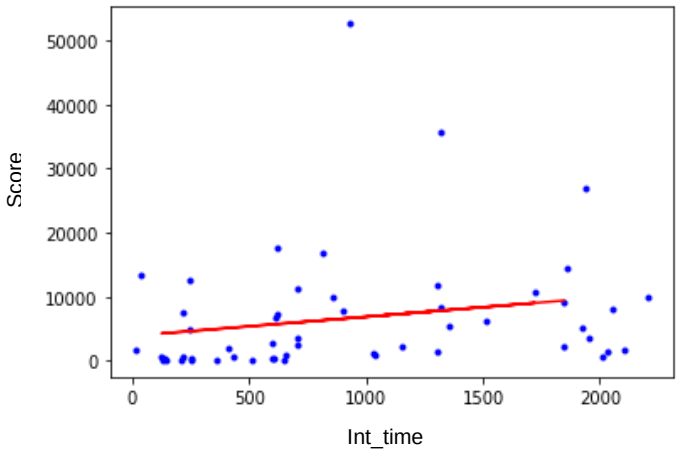

Figure 3: Linear regression on 'meme' subreddits

4b Logistic Regression

For logistic regression, this model proved to be the least useful as we got sort of scattered like intel from figure 4. There is no trend or pattern that we can gather from the logistic approach. However, it was insightful to learn how this model works with different data (meme subreddits) as we can see with figure 5 that it still skews the data. Overall, logistic regression was not as helpful in our problem as we would have liked which is why we decided to try polynomial regression next.

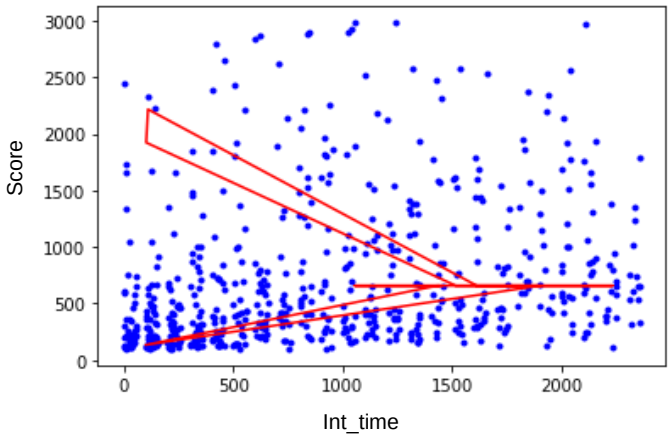

Figure 5: Logistic regression on data

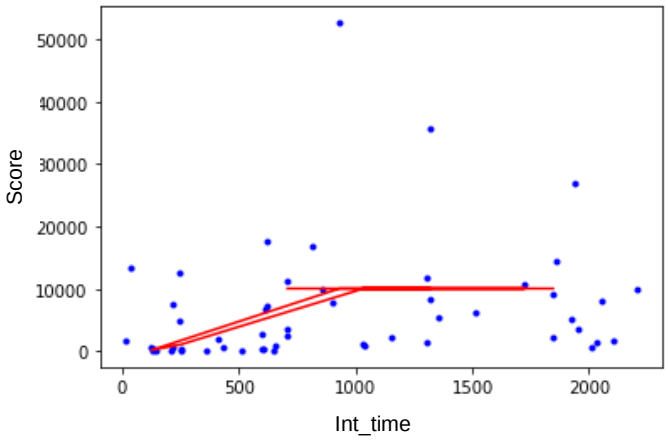

Figure 4: Logistic regression on 'meme' subreddits

4c Polynomial Regression

We found that the vast majority of successful posts had scores below 2,000. We also found that the scores trended upward throughout the day with a peak at around 9:00 PM Eastern.

From our analysis, we decided to go with a degree value of 5 as it looked like from the plot in figure 6 that it properly represented the data better than other values for the degree. In figure 6 we can see that there is a trend to follow and certain peaks throughout the day. This corresponds to the time block analysis that was done to see what five-hour time blocks have the highest average scores.

We are happy about the polynomial results as we think it represents a good prediction of the best post times for Reddit. Polynomial regression shows more accurate results than the other two approaches which is useful for our problem and provides the best solution between them.

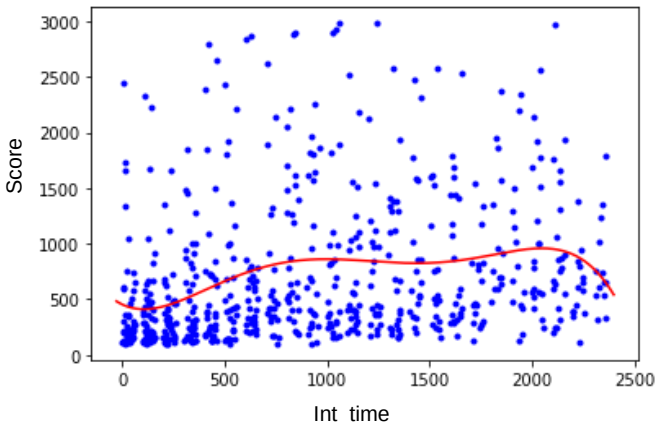

Figure 6: Polynomial regression on data, degree = 5

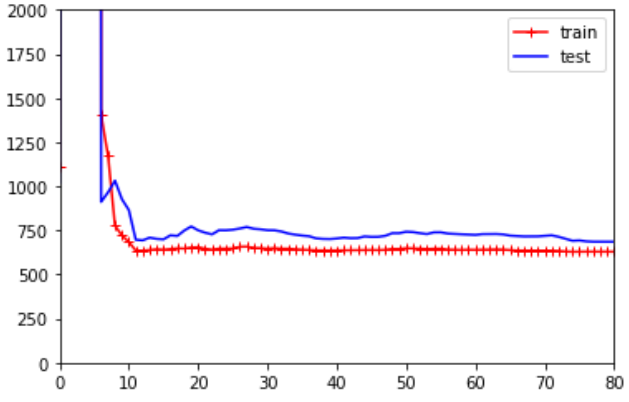

Figure 7: Polynomial learning curve

4d Other Experimental Methods

In our Jupyter Notebook we have the original data that was not filtered and show how cluttered the data looks when plotted (figure 8). This was important to our project because a majority of it was filtering the data and figuring out what was the most useful information. In figure ** below we can see the over cluttered data. However, in figure 9 we can see a clearer pattern than previously and decided that the filtering of the data was useful. Eventually we decided to use a max score of 3000 for our filter. This experiment was needed to show how cluttered data should be filtered but not too much.

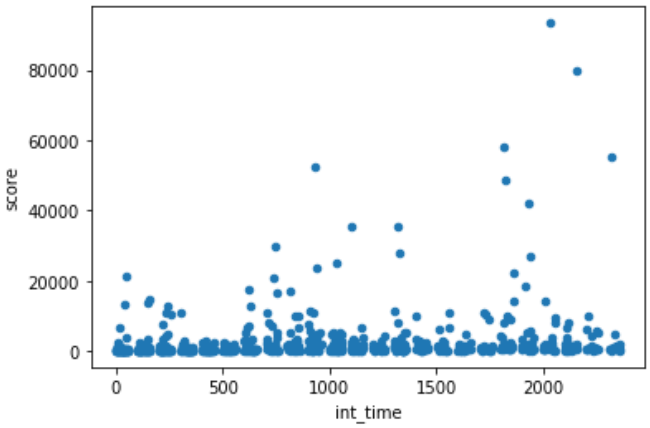

Figure 8: Unfiltered data

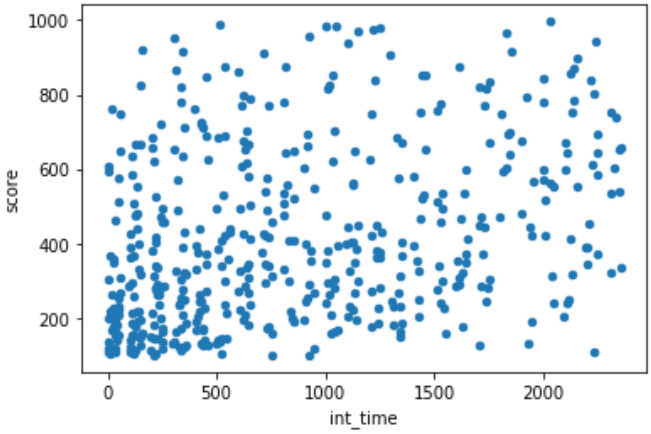

Figure 9: Filtered data with score < 1000

Another form of analysis we did was use time blocks to see if there was a noticeable difference in the average scores between different timeframes. Our conclusion was that it was noticeable and that it correlated with our polynomial regression approach.

Time Blocks	Average Score
12:00 AM to 5:00 AM	1098
5:00 AM to 10:00 AM	2746
10:00 AM to 3:00 PM	2294
3:00 PM to 8:00 PM	3876
8:00 PM to 12:00AM	4593

5 Conclusion

5a Lessons learned from results

We learned that while randomness accounted for the overall score of the posts, there was a clear correlation between the time of day and the overall post success. It was evident from the polynomial regression graph that posts submitted around 21:00 Eastern Time did better on average than any other time. So a user would likely want to submit a post around that time for a better chance at having a successful post. This could most likely be attributed to the active users on the site at that time being higher than other times.

5b Lessons learned from the experience (technical)

We learned that regression as a prediction tool is a great way to give insight into the behavior of the data but is in no way absolute fact when it comes to predictions. There will always be randomness within the data that won't allow for perfect predictions. However, with the help of these regression algorithms, we can skew the predictions to be better than random averages.

5c Next steps

Possible future work would be to analyze the Reddit data where we would use age, popularity, timing and other useful factors to determine possibly how to get the most upvotes. It could even be possible to analyze other social media platforms in a similar sense like Instagram, Facebook and Twitter. Of course, a big challenge for this would be being able to gather the proper data to do such analysis.

6 References

[1] A. Terentiev and A. Tempest, “Predicting reddit post popularity via initial commentary,” *Computer Science,* 2014. [Online]. Available: http://cs229.stanford.edu/proj2014/Terentiev%20Tempest,Predicting%20Reddit%20Post%20Popularity%20ViaInitial%20Commentary.pdf. [Accessed: 20-Apr-2020].

[2] *Pushshift.io*, Reddit, Mar. 2020. [Online]. Available: https://files.pushshift.io/reddit/ [Accessed: 15-Feb-2020].

[3] J. Segall and A. Zamoshchin, “Predicting reddit post popularity,” *Computer Science,* 2012. [Online]. Available: http://cs229.stanford.edu/proj2012/ZamoshchinSegall-PredictingRedditPostPopularity.pdf. [Accessed: 20-Apr-2020].

[4] Stansbury, Chapter 4 Demos, Github.com, Online at: https://github.com/richss/cs455-ml-demos/blob/master/Ch4-Demos.ipynb [accessed 4/22/2020]

7 Appendix I: Justin’s Personal Contributions

Created the sanitization python script
Created the Jupyter notebook
Imported the data
Helped build the slideshow presentation
Helped write the report

Lessons Learned
This project taught me a lot about the process of constructing a machine learning solution from scratch. I learned that the bulk of machine learning work is preparing and analyzing the data before any machine learning actually takes place. The data we were able to get was in a state that was unusable to us due to time and equipment limitations. We learned how to sanitize our data as well as pre-process it so that it would be accepted by the machine learning algorithms. I also learned that not all algorithms are created equal no matter how similar they may seem. This was shown through the results we got from three regression algorithms we used.

8 Appendix II: Kaitlyn’s Personal contributions

Logistic Regression
Polynomial Regression
Cleaned up notebook

Learning curves
Analysis on time blocks
Analysis on subreddits containing ‘meme’
Work on the report
Demo

Lessons Learned

This project was challenging and fun to work on. I learned a lot from analyzing data and utilizing different algorithms. I think there could be more research done in this topic outside of this class. Lesson learned is that sometimes the data may not come exactly how we wanted it which is why it is important to pre-process and filter it. Also, that not all algorithms are going to work as planned and that it may take multiple attempts to truly get your results.