Manh Sy, Student ID: 219-386-155
Sharon Fitzpatrick, Student ID: 301-698-202 (Sharon dropped the course after finishing our project!)
Project 1, due date: 2/16/2021

Predict Yelp Rating using Neural Network

(1) Problem Statement

Ideally, Yelp would utilize customers' reviews to inform businesses of how they should adapt their business models to best serve their customers. The reality is there is no way for a small business to sift through thousands of reviews to find valuable customer feedback that impacts customers' perception of their business. This means that businesses are missing crucial feedback in customer reviews that could be hurting their public reception and prevents businesses from further optimizing their business to meet ever-changing customer needs. Fortunately, a model that can predict a business' rating based on customer feedback can show businesses what customer concerns and feedback truly matter to their businesses' reputation.

(2) Methodology

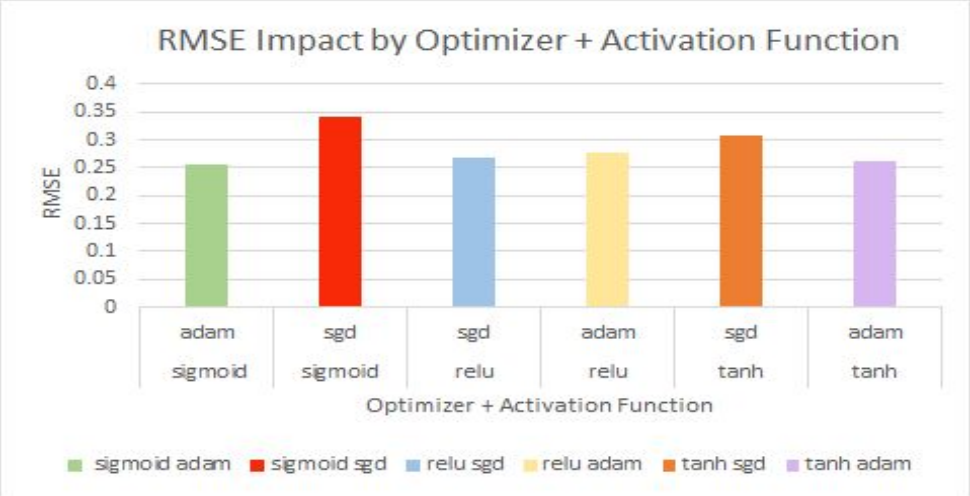
To gather the data necessary to build the review rating prediction model we utilized the yelp business and review JSON datasets, which were scraped directly from the yelp website. These reviews were left by millions of customers from diverse backgrounds making our data quite diverse. Once we had our initial data we needed to clean and organize the data to be utilized within our model. The first step was converting the JSON formatting files to pandas data frames. Once finished, we will drop any unnecessary columns/rows that were not relevant to our model, which included businesses with less than 20 reviews.

To discover how word choice in the reviews impacted the businesses we utilized a TF-IDF vectorizer to determine the importance of each word by measuring their appearance in each review, while also discounting fillers words. Once completed, our TF-IDF matrix will be transformed into pandas data frame which will be considered as our input feature, x. As for the target features, it will be the actual rating of each business. We will place our data within several models with various activation functions, such as relu, sigmoid, and tanh alongside different optimizers like Adam and Sgd.

To test our model, we will calculate the RMSE score for each combination of activation functions and optimizers, along with different numbers of neurons and layer counts, and deem the best combination of these elements to be the model with the lowest RMSE score. Finally, we

will put the best model to the test by predicting four businesses' ratings and calculate the difference between our model's prediction versus the actual rating.

(3) Experimental Results and Analysis



Activation / Optimizer	sigmoid adam	sigmoid sgd	relu sgd	relu adam	tanh sgd	tanh adam
Batch size	64	64	64	64	64	64
Input layer neuron count	64	64	64	64	64	64
Hidden layer neuron count	32	32	32	32	32	32
Epoch	19	50	47	14	18	13
RMSE	0.25651	0.3417	0.269	0.2773	0.3076	0.2616

Our worst model was produced using a combination of hyperparameters of activation function Sigmoid and optimizer SGD. Which ultimately yields the highest RMSE score of *0.34173786535948*. The best model we implemented was the Sigmoid activation function alongside an Adam optimizer, which resulted in the smallest RMSE of *0.2565103735458283*.

All models were tested with a batch size number of 64, input layer neuron count of 64, and hidden layer neuron count of 32. All of which were specifically hand-picked after we tested numerous combinations of layer and neuron counts that ultimately did not have a significant impact on the resulting RMSE score.

(4) Task Division and Project Reflection

A. Task Divisions:
   a. Data extraction and cleaning: Manh & Sharon
   b. Creation of tf-idf matrix: Manh
   c. Testing of different tf-idf parameters to extract the most relevant words: Sharon

d. Create the train/test models with different combinations of hyperparameters: Manh
e. Record RMSE and plot lift chart of the best model: Manh
f. Chose 4 random businesses of different categories to test against our best model: Sharon
g. Though different members of the group were assigned a specific task. We both consulted each other about the best method to solve each task at hand.

B. Challenges Encountered:

One of the challenges we faced was the amount of time it took to process the initial data and figure out how to effectively clean the data. We were not prepared for the enormous sizes of the files and discovered that operations such as converting the JSON to data frames could take hours to finish on our laptops. The solution to the problem was for the initial model to not read in the entire file, so our computations would be faster. After we trust that our models have been built properly, then we proceeded to process the whole dataset.

Another challenge we faced was the disagreement on how to initially approach the task at hand. Initially, we deduced different approaches to cleaning the data, both of which ultimately could be turned into a model. However, only one set of code could be chosen. After some discussion, it was decided we would use Manh's version since her approach was more straightforward.

The last problem we faced was figuring out the best method to share our programs and progress. We found a combination of using GitHub to maintain an up-to-date codebase along with using Discord to communicate our ideas was the best solution.

C. Reflection:

As a team, we learned that a major component of machine learning relies on data cleaning and formatting your data in such a way that is understandable to model. We also discovered the importance of having a fast computer to be able to read through massive data files, which we had not considered much before this project. Once our dataset was cleaned and filtered through we found that the activation functions and optimizers greatly impacted the performance of the model compared to the number of neurons and layer count. Overall, this project gave us a more realistic insight into the world of machine learning.