**Predicting 5-Star Yelp Reviews:**

*An Experiment Into Building*

*Artificial Neural Network Models*

Programming for Analytics  
Group 9

*Vinav Chauhan*

*Leiwen Lin*

*James Phillips*

*Marcus Ranney*

Table of Contents

[GUIDING QUESTION 3](#_Toc500706515)

[STORYLINE 4](#_Toc500706516)

[DIFFICULT MOMENTS 6](#_Toc500706517)

[POSITIVE MOMENTS 7](#_Toc500706518)

[DIVISION OF WORK 7](#_Toc500706519)

[WORKFLOW 8](#_Toc500706520)

[RESOURCES 9](#_Toc500706521)

# **GUIDING QUESTION**

The question that our team is looking to tackle can be stated as follows: “To develop a model, that after analyzing attributes of a business and the attributes of the user predicts whether that user will give that particular business a 5-star review or not, solely for restaurants”. This question is important from a marketing point of view, especially targeted marketing. We looked at the question from a business point of view and wanted to provide the tools for restaurants to gain a leg-up and help them operationally overall using the power of data.

The Yelp dataset provides most of the important information that is required to build the model. The Business, Review, User and Category are the datasets that we used to help build the model. We decided to use the KERAS library, which is a high-level neural API written in Python and capable of running on TensorFlow to construct our model.

In terms of the data that is lacking, more information about demographics would be helpful. For instance, information regarding age and sex would be beneficial. Age is major factor that determines the preferences of an individual when it comes to a restaurant. For example: You would expect an older individual to prefer a casual laid-back atmosphere and rate it higher than an upbeat resto-pub. From the point of view of targeted marketing, data or information about a yelp reviewer’s friends and network would be helpful. User location is another important piece of information that is missing. An individual’s location could provide valuable information about their economic and social status, which in turn influences their perceptions and expectations from a restaurant. The ‘business’ data includes entries that describe the cuisine instead of tagging it as a restaurant. For example: some businesses are tagged as pizza instead of ‘restaurant’. With respect to difficulty regarding the execution of this project, we are only focusing on business tagged as ‘restaurants’. It would have been much better if the restaurants had a ‘restaurant’ tag along with the cuisine in the dataset.

# **STORYLINE**

We started thinking about our guiding question by coming up with a question for ourselves ‘How can we use this data to help a Restaurant?’. Our first response was that we couldn’t help them with anything related to operations or the actual running of the restaurant, but we could help them with their sales and brand image. This first led us to the conclusion that we could help restaurants target certain kind of customers that would be best for their business. We could therefore do an analysis for different kind of restaurants, for example: Italian, Mexican, American and find the attributes of customers that like these restaurants. This would help increase sales or better their brand image, but what about protecting it? A restaurant cannot control the patrons that it receives. So, this led us to the question – “Can we increase the likelihood of someone giving a good review?’ or what if the restaurant already knew what review it would receive from a certain patron. This would definitely help the restaurant be better prepared to serve all kinds of patrons.

The restaurant business is extremely cut-throat and word-of-mouth can make or break a restaurant. Since there are tons and tons of restaurants and it isn’t possible to go to all of them, people like to go to the ones that are rated highly by others as this is the most reliable way of ensuring a good quality product and service. Also, it is imperative that a patron have a good experience because you rarely get a second chance in the restaurant business. Once a customer have a bad experience, they are very unlikely to return and are more likely to share and remember their bad experiences rather than good ones. In this day of digitalization, online reviews are best way for a restaurant to gain publicity. Positive reviews would mean not only more customers but also the possibility of added publicity from important individuals of the industry such as other chefs, food critics, bloggers etc. These ‘important’ individuals are more likely to visit a restaurant with ‘positive’ reviews and they may broadcast positive views and help increase popularity and awareness of the restaurant. With all this in mind, we decided to build a model that would help predict the rating a user would give to a particular restaurant depending on both their attributes.

With this model, restaurants would be able to conduct targeted marketing. They could browse yelp users and target those users that have attributes that would result in a 5-star review. Restaurants could also prepare and pay attention to guests that have attributes that are more likely to give a negative rating. Our model returns the probability of a Boolean response – It is used to predict only whether a customer will give 5-star rating or not. We can develop similar models that predict 1-star, 2- star, 3-star and 4-star models and run the data through all these models to predict the likelihood of what rating the user is most likely to give. From a business perspective, this model is something that Yelp could offer to all restaurants on its platform. It could provide a service wherein if a patron confirms a reservation through Yelp’s reservation system then the system could run the model, comparing attributes of the patron and the restaurant to directly provide the probability of a rating to the restaurant with the reservation information.

Although, the workflow of the entire project is described further in this paper, the general procedure of our project was as follows – We started by creating the dataset relevant for our project by combining information from the ‘business’ ‘review’ ‘user’ and ‘category’ dataset. After data manipulation, we split the data as 70% training and 30% test. We used the training data to build our model that was tested. We went onto calculating the accuracy, specificity and precision. We ended up with a good model with precision of just under 87%.

# **DIFFICULT MOMENTS**

The majority of our difficult moments came during the actual programming workload, yet, getting all of our team members systems updated with the needed software packages took a longer, more complete effort than we all anticipated. This was highlighted by the group dynamic of having conflicting schedules and a team member that is working a full-time job, thus mainly available on nights and weekends.

Initially, we struggled to settle on a business problem to address. We began digging into the data with only a rough mission idea, luckily, the data available drove our team to settling on developing a predictive model. While this was only a base, it gave us a direction to guide our next steps. Our next troubles arose when we were attempting to load the json data sets from Juptyer Notebook into SQL. Leiwen was able to solve this by writing a function to load 10,000 data points at a time (the max allowed).

For our first predictive model, we used regression analysis, and the results were underwhelming. The model was hardly more predictive than the datasets established distribution of 5-star to not 5-star reviews of 35:65. At this point we took a step back to re-evaluate if a predictive model was going to be possible or if a pivot towards descriptive analysis would be more beneficial. Fortunately, we decided to stick to our original mission, and utilize the Keras neural network package to attempt a more precise model. Running the new model ended up taking over an hour to process on each attempt, but we avoided major processing interruptions throughout the process.

Apart from these temporary roadblocks, our group experienced the typical programming headaches of missing comma’s, function attributes, and stack overflow searches.

# **POSITIVE MOMENTS**

Any time our code ran without any errors was considered a positive moment. There were really the most gratifying moments of our project, but as a whole, our group enjoyed working together.

From a project management standpoint, an effective and efficient group dynamic was established early. A work breakdown structure was created based on the individual strengths of our group members, this facilitated smaller working sessions that proved to drive the overall progress of the project.

Finally, the best coding moments the successful running of our neural network model and the precision results that the model produced. If the model was as ineffective as our previous regression model, we would have been in a scramble to reassess our entire project. Additionally, the completion of our data wrangling process was welcome forward progress.

# **DIVISION OF WORK**

The division of responsibilities in our team was decided based on the strengths and weaknesses of each of our team members. Leiwen excels with writing code, a skillset he honed before this class had begun, so it made sense for him to take the lead in coding and code documentation. Aiding him in the creative process and structure of the code and overall project, Marcus was extremely adept. So not only did Marcus aid Leiwen with the coding and coding documentation, but he also documented the best and most difficult moments that occurred during this group project. In addition, Marcus and Leiwen worked together to track and document the workflow needs through the development of our work product. Vinav and James tackled the guiding question of our project effort and Vinav followed with all the subsequent questions associated with the overview. He did this by asking the tough questions and ensuring that our efforts and work product remained in scope of the requirements. James also took on the YouTube video presentation. The "clean up" of the project, which included the administrative efforts such as paper editing was a collaboration of all team members.

# **WORKFLOW**

1. Get randomly assigned into a group of all-stars
2. Downloaded the needed datasets from Yelp
3. Import Packages needed – Numpy & Pandas
4. Read all datasets into Jupyter Notebook
5. Filter users down to those with >=5 reviews or >=2 reviews with an account created on or after 1/1/2016
6. Filter businesses to those with a category variable of “Restaurants”
7. Filter location of businesses to only the top 20 locations
8. Normalize attributes column in the business dataset to only include the top 20 attributes with the least missing values, and create columns for each of the 20 business attributes
9. Connect Jupyter Notebook to MySQL
10. Import chosen columns from pandas objects into MySQL database to run queries if need be (we end up using pandas for final table creation)
11. Merge, transform, rename chosen columns for individual datasets to create final table: table\_whole.pickle
12. In a new Jupyter Notebook, read our created final table
13. Set our target variable to all reviews with 5-stars, and determine what percentage of our dataset is our target (almost 36%)
14. Determine the datatypes for all the variables in our table, reorganize to group categorical and numeric variables together
15. Create dummy variables for all categorical variables
16. Assign our data into training and test datasets (70:30)
17. Create the new datasets based on the newly assigned training and test attributes
18. Standardize the independent variables prior to entering into Keras
19. Install and import the Keras package
20. Add layers to the ANN model (input and two hidden layers)
21. Compile our neural network model 300 times on our training data based off the following:
    1. *Optimize*: Adam1
    2. *Loss*: Binary Cross Entropy
    3. *Metrics*: Accuracy
22. Wait
23. Wait
24. Determine results: AUC (training & test), accuracy, specificity, and precision
25. Celebrate

# **RESOURCES**

**1Adam: A Method for Stochastic Optimization**

We introduce Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients, and is well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. The hyper-parameters have intuitive interpretations and typically require little tuning. Some connections to related algorithms, on which Adam was inspired, are discussed. We also analyze the theoretical convergence properties of the algorithm and provide a regret bound on the convergence rate that is comparable to the best known results under the online convex optimization framework. Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods. Finally, we discuss AdaMax, a variant of Adam based on the infinity norm.

**Source:**

Kingma, D. P., & BA, J. (2014). Adam: A Method for Stochastic Optimization [Abstract]. *CoRR,* *Abs/*(1412), 6980th ser. Retrieved December 1, 2017, from https://arxiv.org/abs/1412.6980v8.