Help Yelp! Course Project: Midterm Report

Group 26: Data Wranglers
Lucas Jenking, Tejasvi Kasturi, Matthew Nuesca, Cole Zimmerman

November 10, 2018

Abstract

The Help Yelp! Course Project demonstrates the practical applications of the different algorithms and data structures taught in class. As a group, we are tasked with developing a multi-class classification model to predict ratings of a user that visits a new business. We are given various types of data about businesses, customers, and reviews recorded from Yelp. Using this data as training data, our model is supposed to take in a business's name and a user who hasn't been there before and output a prediction of the customer's star rating of the place.

1 Proposed Method of Model Development

Developing an ideal model requires us to develop multiple models and see which fares best with our evaluation method (RMSE). Our proposed models include the following each of which will be explained below: 1) linear regression 2) ridge regression 3) neural network. Additionally, the preprocessing and data cleaning process is required before inputting the data into proposed models and will also be explained.

1.1 Data Cleaning and Preprocessing

The data given on Kaggle is unsuitable for our models to use. As a result, we need to develop a library to clean and preprocess given data for use in our proposed models. The current data has many features that we perceived as insignificant in contributing to most user's ratings of the businesses. Based off of personal opinion, percentage of null values, and variance of values among data points, we filtered columns into important to use features, possibly useful features, and unused features. We one-hot encoded any features that were booleans (0 for false and 1 for true) as well as any attributes that took on string values. For features that still contained empty values, we replaced boolean and numeric features with the mean. Empty string values were hot encoded as a NaN.

Additional utility functions were implemented that, given the business id and user id, returns important or important features combined with the possibly useful features given the business, the average user rating, and the star rating the user gave to that business in their review of the business. These functions will be able to generate the design matrix used in our models.

1.2 Simple Linear Regression

Our current linear regression focuses on using 2 features. The average star rating of the business, and the average user rating's offset. This average offset is taken by calculating the difference between the user's review and the rating of the business that the user gave a review about for each review of the user and averaging that difference. Currently no weights are given to the two features and can be used in the future to provide a better model fit. Additional features can be added to this linear regression to take into account to also reduce this model's error.

1.3 Ridge Regression

Since multiple values in our definite features list can result in the same star rating, we believed that using ridge regression may be a good alternative to our simple linear regression. Ridge Regression is a model that takes into account many-to-one relationships which is evident in our current dataset. We use ridge regression on all of our features that we believed are to be important and by using the given Ridge function in the scikit-learn module. While this performs better than our simple linear regression, it can be improved upon by adding features found in our maybe list.

1.4 Neural Network

Another proposed model is to use a neural network. Due to it's power to create weighted relations between features and the possibility of using the root mean square error to adjust the weights of these relations, we believe that this model would work best for predicting ratings under our evaluation criteria. While we know what loss function to use (RMSE), we have to still experiment with activation functions and network topology. Our neural network will be implemented using Pytorch's neural network package.

Evaluation Criteria

Root Mean Square Error is the function used to evaluate our model. It will be used to see how we compare to other models developped by other groups. The given equation for RMSE is as follows:

$$Err_{RMS} = \frac{1}{N}\sqrt{\sum_{j=i}^{N}(y_{predict} - y_{expect})^2} \tag{1}$$

N corresponds to number of samples in the test set, $y_{predict}$ corresponds to the output from our model, and y_{expect} corresponds to the expected output from the test data. RMSE uses distance as a measure of how far off our model is from the actual prediction which provides an intuitive understanding of the difference between our model and actual behavior. Additionally, RMSE is differentiable which makes it a good error function to adjust weights with gradient descent.

When selecting a model, we will mainly use RMSE to determine which model performs the best. In general, smaller error means a better fit to the scenario so we would choose the model with the minimum RMSE. Currently, our ridge regression model performs best with an RMSE of 1.1077.

Discussion

While discussing our RMSE as an error function, we realized that its downside is its sensitivity to outliers. As a result, we have concerns over how to remove outliers and implementing a method to remove outliers is an important portion that needs to be implemented in our data processing and cleaning library.

After testing our baseline linear regression out, we realized that it basically just outputs the average rating of the business. This is probably due to the fact that the average offset is a smaller number compared to the business's average rating. As a result, we may make improvements by adding weights as well as adding more features. As stated above, ridge regression was an alternative to our simple linear regression. We tested our implementation and as expected, the ridge regression provides a better fit to predicting ratings.

We predict that the neural network would work best, but we currently have no working implementation of it. As shown in our schedule, most of our time will be used to develop the neural network.

Schedule

Below is our planned schedule in developing our model.

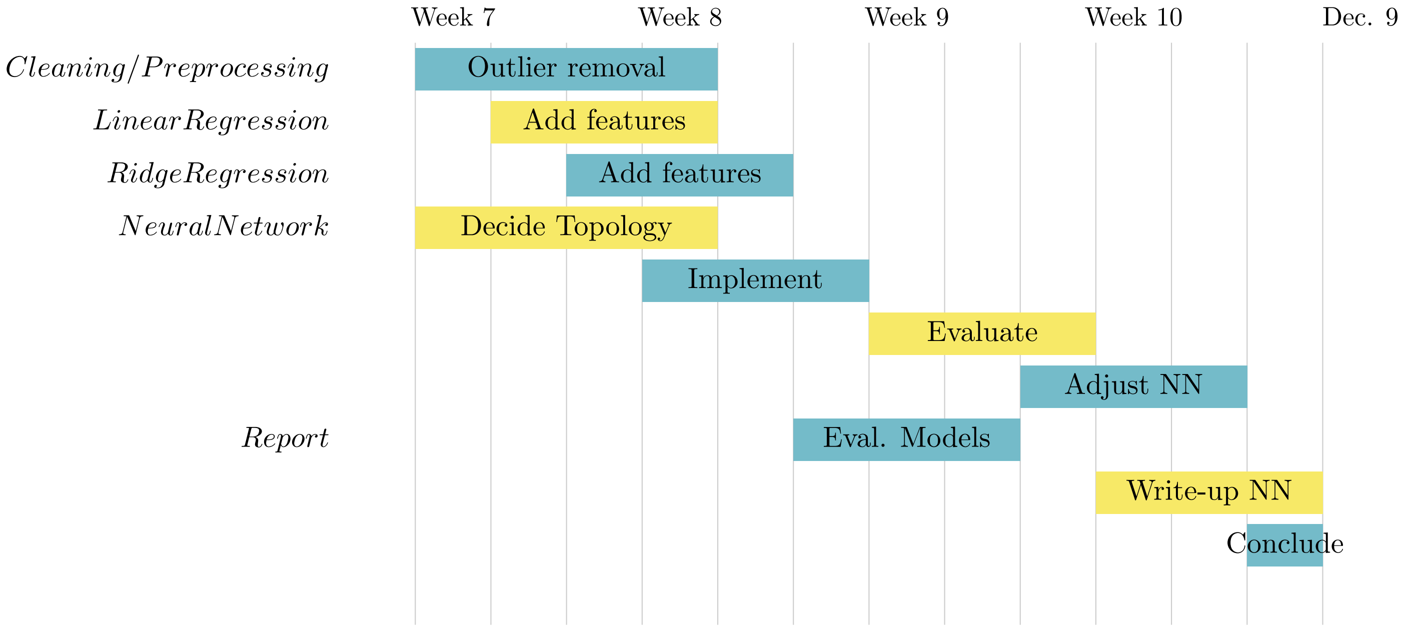

Figure 1: Model Development Schedule

References