User Rating Prediction

Ashwin Sattiraju
University of Illinois at Chicago
asatti2@uic.edu

Gowdhaman Sadhasivam
University of Illinois at Chicago
gsadha2@uic.edu

Abstract

Amount of data exist in the digital universe is huge and available in many formats. Online reviews about a product or business exists in the form of text. The length of these reviews may vary and opinions expressed in the text are less intuitive unless we read it completely. It is useful to measure opinion expressed in the review text in the form of a rating of scale 5 to make it easy for the users to quickly estimate how good or bad the business is. Such measure summarizes the business in the form of user's rating. In this project we propose an approach that predicts user's rating of a business by extracting TF-IDF, sentiment and topics from the review text. We considered only reviews belonging to restaurant business only. We compare models using individual features and also models combining all the features applying different machine learning techniques to those features.

Keywords – rating prediction, nlp, convolution neural net, svm, lda, sentiment analysis.

1 Introduction

The objective of our project is to predict user's star rating on restaurant business from his/her review text. Online reviews on restaurant are important as they help other customers to try different restaurant with consistent quality. Nowadays several online platforms are available to allow user to write their own reviews such as review of a product, movie reviews and restaurant reviews. Some of the examples of these platforms are Amazon, Yelp (used in our project), EAT24, etc. These written reviews are summarized either as star ratings or points on some scale.

This is a very challenging task as the reviews on yelp like tweets are very hard to interpret. They contain lot of internet lingo and also emphasis on words makes it harder for any language processing task. Coupled with the size of Yelp data, it needed lot of preprocessing before it can be used for our purposes. As this dataset is collected from real users it also has many unhelpful reviews and noise.

We tried several models on different algorithms to predict customer's rating from restaurant review text on the scale of 1 to 5. We experimented with the rating prediction system using SVM Linear Kernel, Random Forest and Neural Network algorithms on same datasets.

2 Related Work

The important aspects of our project involves topic modeling and neural network to achieve objective of the project. We searched and found the papers below which are interesting and thought provoking.

Firstly, Latent Dirichlet Allocation (LDA) proposed by Blei et al. [1], 2003. In Blei et al. [1], author proposed a novel approach for topic modeling which is similar to probabilistic latent semantic indexing (pLSI) except that in LDA model topic distribution is assumed to have a Dirichlet prior. Ganu et al. [2], 2009, author focuses on identifying sentiment from free-form text reviews to improve recommendation review accuracy. They manually annotate a set of sentences from restaurant reviews with topic and sentiment to train SVM to classify sentences in the dataset. Their regression-based text ratings are not very accurate, but they are able to improve recommendations to users using text based classification strategies compared to using numerical rating. They showed that both topic and sentiment information at sentence level are useful

information to leverage in a review. James et al. [3], 2013, online LDA, a generative model probabilistic model can be used to extract subtopics from yelp review text and stars per hidden topics predicted. Qu et al. [4], 2010, bag of opinions from review text in the form of root words, set of modifiers from same sentences and one or more negation words extracted. Rating is predicted by assigning numeric score which is learned by ridge regression from opinion words. Duyu et al. [5], 2015, user word composition vector model to capture user's influence on textual content of review is used for review rating prediction. Author used convolution neural net to perform this task, giving us idea to use neural network in language processing.

3 Data

3.1 Introduction

We used data from the Yelp Dataset Challenge [6] and its size is 1.77 GB. This dataset includes business, review, user, check-in and tips data in the form of separate JSON objects file. Each line in respective file indicates one JSON object.

A business object includes information about the type of business, location, rating, categories, and business name, as well as contains a unique business id. A review object has a rating, review text, and it is associated with a specific business id and user id. A user object has name, type, review count, average stars, friends, and votes etc., and unique user id. We focused on business, user and review data in this project. Furthermore, we considered businesses that are of the restaurant category and only reviews associated with restaurant businesses.

This results in almost 20,000 restaurants, and around 1M corresponding reviews. This dataset, specifically the reviews associated with restaurants, will allow us to extract the topics and sentiment from whole review text.

3.2 Dataset Statistics

Below are the statistics about the datasets,

- 1.6M reviews and 500K tips by 366K users for 61K businesses
- 481K business attributes, e.g., hours, parking availability, ambience.
- Social network of 366K users for a total of 2.9M social edges.
- Aggregated check-ins over time for each of the 61K businesses

Countries and Cities,

- U.K.: Edinburgh
- Germany: Karlsruhe
- Canada: Montreal and Waterloo
- U.S.: Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison

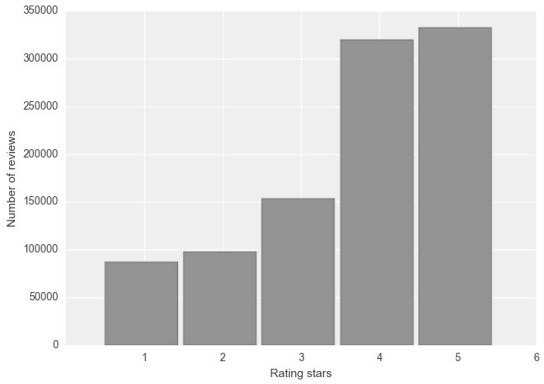

Figure 1 Class distribution for the dataset

In this project we experimented with reviews of length greater than 25 words and lesser than 150 words. Eventually, this results in almost 51K reviews. This is done in order to make the algorithms run on our local machine with limited capacity.

3.3 Format of data

Business data format is as follows

```
{
  'type': 'business',
  'business_id': (encrypted business id),
  'name': (business name),
  'neighborhoods': [(hood names)],
  'full_address': (localized address),
  'city': (city),
  'state': (state),
  'stars': (star rating, rounded to half-stars),
  'review_count': review count,
  'categories': [(localized category names)]
```

```
  'open': True / False (corresponds to closed, not
business hours),
  'hours': {
    (day_of_week): {
      'open': (HH:MM),
      'close': (HH:MM)
    },
    ...
  },
  'attributes': {
    (attribute_name): (attribute_value),
    ...
  },
}
```

Review data format is as folows

```
{
  'type': 'review',
  'business_id': (encrypted business id),
  'user_id': (encrypted user id),
  'stars': (star rating, rounded to half-stars),
  'text': (review text),
  'date': (date, formatted like '2012-03-14'),
  'votes': {(vote type): (count)},
}
```

User data format is as follows

```
{
  'type': 'user',
  'user_id': (encrypted user id),
  'name': (first name),
  'review_count': (review count),
  'average_stars': (floating point average, like
4.31),
  'votes': {(vote type): (count)},
  'friends': [(friend user_ids)],
  'elite': [(years_elite)],
  'yelping_since': (date, formatted like '2012-03'),
  'compliments': {
    (compliment_type):
(num_compliments_of_this_type),
    ...
  },
  'fans': (num_fans),
}
```

4 Models

Our approach used multiple models and methods. We have used 4 models namely frequency, topics, sentiment and combined and used 3 methods SVM, Random Forests and convolution neural network. The baseline models were trained using SVM and same testing data was used throughout project for all other models.

4.1 Baseline Models

Below are the models we considered as baseline for this project. Each model is trained on SVM Linear Kernel classification algorithm. Each model can be used separately to predict review rating.

We combined these three models and benchmark the performance against individual models. Since we are using combination of models, we consider individual models as our baseline. All models use same review dataset but use different features such as term frequency, sentiment and topics extracted from the review text. The dataset used for classification is split randomly into 80% training data and 20% test data.

4.1.1 Term Frequency Model

In this approach word frequency is considered as feature. From the review text, term frequency and inverse-document frequency is extracted and model is trained based on this feature. We also fine tuned the frequencies in order to ignore most frequent and very rare words to reduce over-fitting of data. Once model is trained using word frequencies, we give test dataset to the classifier to predict the ratings.

4.1.2 Sentiment Model

Opinion expressed in the review text has more impact on review rating. These sentiments are hidden inside reviews and can be exploited as features. For example, a review text says, “Food is amazing and their service is good. We had fun.”. We calculate polarity and subjectivity of this whole review text instead of calculating on sentence basis. Polarity score is a float value within the range from -1.0 to 1.0 where -1.0 indicates review text is negative and 1.0 indicates review text is positive. Subjectivity score is also a float value within the range from 0.0 to 1.0 where 0.0 is very objective and 1.0 is very subjective. These features are extracted from review text and model is trained using extracted feature. Once model is trained using sentiment feature, we give

test dataset to the classifier for review rating prediction.

4.1.3 Topic based Model

The previous models considered word frequencies and sentiment as feature which are extracted from review text. Here we represent each review with topics talked about in the review. These topics are then used as features representing their corresponding review. This way we can get to know usually what topics will fetch us better review and which topics actually result in a poor rating.

We used Latent Dirichlet Allocation method proposed by Blei et al. [1], 2003, to extract the topics from the review text. We extracted 15 topics and considered this as feature to train the model. Once model is trained using this feature, we pass test dataset to SVM classifier for review rating prediction.

Sample Topics for a quick run using LDA
Topic 0: burger, greasy , away , waiting , soggy
Topic 1: bad, closed , minutes , poor , food
Topic 2: go, wrong , order , food , half
Topic 3: food, rude , good , like , wasn't
…

4.1.4 Combined Model

As observed in the results above, individual features, i.e. tf-idf model performed well when compared to topic model and sentiment model. Performance metrics values of sentiment model is better than topic model. However, these values are less when compared to tf-idf model. This could be due that topics discussed in review text do not fully correlate with the rating. Similarly, sentiment expressed in each sentence on review text could be mixed rendering it useless in some reviews. This thought led us to combine sentiment and topic model along with better tf-idf model. This way we can boost tf-idf with additional information regarding sentiment and topics user is talking about and obtain more accuracy. This was the hypothesis we wanted to prove.

5 Experimental Setup

The experiment was broken into 2 parts. The first part included regular machine learning approaches SVM and Random Forests. The second part was to build a deep learning network to do the classification for us

In Figure 2. The review, user and business JSON files are converted into .csv files and combined into single data-frame. We removed attributes which have null values from the data-frame and filtered the review of business that has category restaurant. This results clean data-frame.

TF-IDF, Sentiment and topics are extracted from the resulting data-frame and combined them into single feature set as usable features.

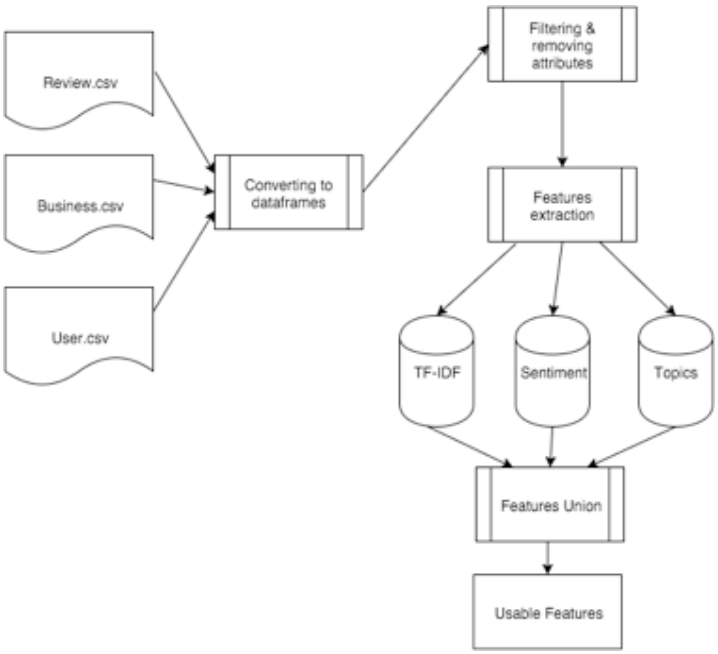

Figure 2 Experiment setup architecture

5.1 Regular Approach

In this approach we used SVM Linear Kernel and Random Forest algorithms.

In SVM, number of 4 and 5 rating reviews are higher when compared to other ratings. This led us to give weights to unequal classes in order to remove skewness of the data. Sampling would also work in this case.

On the other hand, in Random Forest classifier we used 100 decision trees and we considered maximum depth of each tree until all leaves are

pure. We ran the following random forest in parallel.

Although random forest did not yield better results. We learnt how to use ensemble techniques and how weak learners can be used to build better classifier.

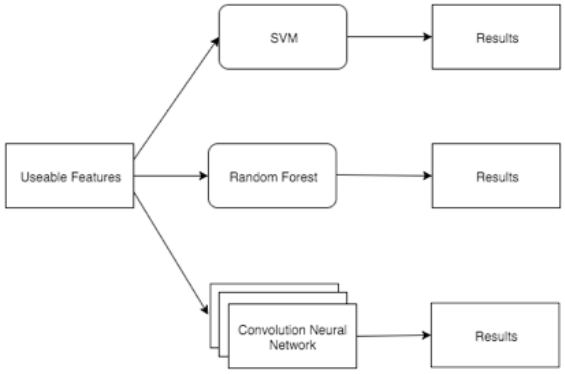

Figure 3 Regular approach architecture

5.2 Deep Learning Approach

In this approach we use 2 convolutional neural nets with different architectures, one deeper than the other. Both the convolution networks use 1D convolution. For each model we used 0.1% of train data as validation set to tune architecture parameters. Also we used categorical cross entropy with Stochastic Gradient Descent for optimizing weights for the network. Later we had also used Adam optimizer. Below we will discuss both architectures in detail.

For optimization of weight in both architectures we used learning rate of 0.01 and a decal of 1e-7 with 0.9 momentum for stochastic gradient descent and for the Adam optimizer we used the 2 betas to be 0.9 and 099.

5.2.1 Simple Architecture (CNN1)

In Figure 4, we used 1 input layer, 3 convolution layers, 3 max-pool layers and 1 output layer. Each convolution layer has 32, 64, 128 filters respectively. Each filter is of the size 3 and pooling size is 2. The stride is 1.

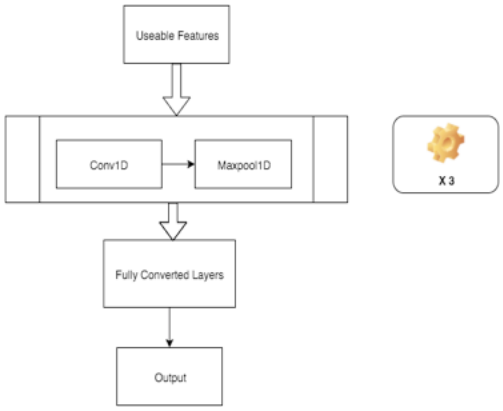

Figure 4 Simple Convolution Architecture

Rectified Linear Unit (ReLU) activation function is used for each layer. In Figure 5, The blue line in below graph is ReLU function and the green one is softplus.

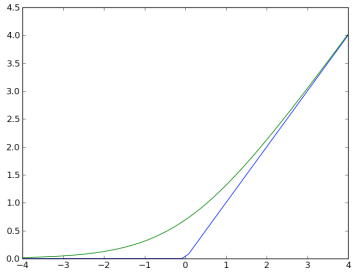

Figure 5 Relu Activation Function

$$f(x) = \max(0, x)$$

Output layer which is a fully connected layer has soft-max activation function along with 5 neurons one for each class/ rating.

5.2.2 Deeper Architecture (CNN2)

In Figure 6, we used 1 input layer, 5 convolution layers, 4 max-pool layers and 1 output layer. The first 2 convolution layers have 32 filters each and each convolution layer after that has 64, 128, 256 filters respectively. Each filter is of the size 3 and pooling size is 2. The stride is 1.

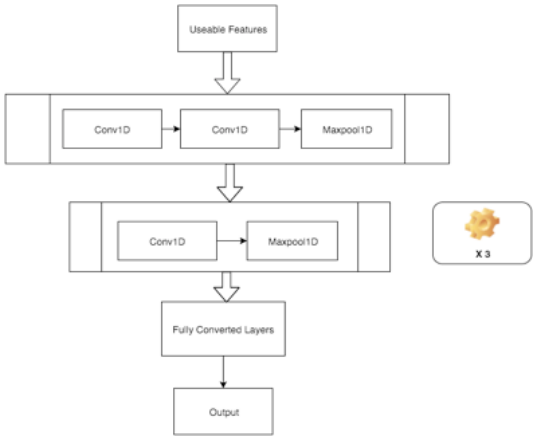

Figure 6 Deeper Convolution Architecture

Rectified Linear Unit (ReLU) activation function is used for each layer. Output layer which is a fully connected layer has softmax activation function along with 5 neurons one for each class/ rating.

6 Results

6.1 Baseline Results

Baseline with SVM	Precision	Recall	F1 Score	Accuracy
TF-IDF	45.79	51.16	47.74	75.58
Sentiment	30.41	37.57	31.7	63.92
Topic	27.24	31.35	26.57	56.19

Table 1 Baseline models results

From Table 1, it shows the result of three baseline models. It is observed from the table that, TF-IDF model outperforms sentiment and topic model. This could be due that topics found provide less information for review rating prediction since there was a little low correlation. However, recall score of topic model is more than precision score.

6.2 Combined Approach Results

From Table 2, it clearly shows that combined model outperforms baseline models. In the combined model with SVM Linear Kernel classifier, F1 score improved by 90% and 60% when compared to baseline topic model and sentiment model respectively.

Algo-rithms	Precision	Recall	F1 Score	Accuracy
SVM	48.49	54.85	50.85	76.88
Random Forest	53.93	41.60	43.60	78.43
CNN – model 1	53.18	48.72	50.35	77.80
CNN – model 2	57.76	53.35	55.04	81.73

Table 2 Combined models results

We tried different algorithms with same experiment setup and compared with SVM classifier. Random Forest algorithm accuracy is slightly higher than SVM but its recall and F1 score is lower. In the case of Convolution Neural Network model 1, model with three layers, showed results as similar to Random Forest classifier. On the other hand, Convolution neural network model 2, model with five layers, showed better results when compared to all other models. Though the recall score is lower than SVM model but it is less statistically significant. However, F1 score and accuracy values of CNN model 2 is better than Random Forest and baseline models.

7 Conclusion

To conclude, this project presents interesting results on deep learning techniques applied to a real world problem. Also the comparison of regular algorithms with a deep learning network. Although the overall F-Scores were mediocre there was significant improvement in the performance

8 Future Work

Future work in this project would be to explore more preprocessing techniques. Improving topic coherence would yield better results. Also combining user and business profiles with review text and using tip (short text summary) data will

also improve the results. Working with different bag of words model which contain few words based on the given review. Apply sentiment to sentence level and check if this yields better results.

Also formulate and extract more feature from given reviews. Use of Discourse. It would be interesting to compare a deterministic model with Deep Learning Networks.

References

[1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022.

[2] G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In WebDB, volume 9, pages 1–6, 2009.

[3] Improving Restaurants by Extracting Subtopics from Yelp Reviews - James Huang, Stephanie Rogers, Eunkwang Joo (2013)

[4] The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns – Qu , Ifrim & Weikum (2010)

[5] User Modeling with Neural Network for Review Rating Prediction - Duyu Tang, Bing Qin, Ting Liu & Yuekui Yang (2015)

[6] https://www.yelp.com/academic dataset