Data Management Team Project Report

Yoav Kaliblotzsky Garrett Boseck Nathalia Sandoval David Liang Jasmin Adzic
Daniel Brown Nick Webb Shaifali Goyal Yuji Mori Steve Huynh Ronen Burd

ECS 171 Fall 2017

Abstract

The Database Team was tasked with creating models for the data and collecting it. This data was then compiled into a robust database that was used by teams designing machine learning algorithms to predict how much a user would like a given joke. We collected user, joke, and ratings data through online surveys, and then compiled it into an SQL database. After some basic cleaning and filtering, the databases were released. We then analyzed how effective the features chosen were when trying to predict user behavior in two ways, by being able to predict their rating for a certain type of joke and by predicting the variance of their ratings. We found moderate success in both of these methods.

0.1 Introduction

In machine learning, proper data is imperative to our learning model, since it is the data which drives the model itself. Sometimes, the data can be more critical in model performance than the algorithm in use. Therefore, there are many important aspects to consider when creating a data set for machine learning.

In order for a dataset to be usable, one must decide which features are relevant to the algorithm, to gather from samples. Good features should be independent, and have enough variance to distinguish samples (Brownlee). However, too much variance and too many features with too small a sample size can also lead to overfitting for certain traits (Amatriain).

Another factor that affects feature selection was how we were gathering data. We needed to survey students in order to get qualitative results to create user profiles. There are two main types of questionnaires we could have used. It can be very structured, which allows for collecting the data quickly and ensuring there is not too much variation, but they constrict users responses to fewer possibilities and to be less specific than they otherwise could be. (Gill et al, 2008). It could also be freeform, allowing users to respond however they please, creating more accurate responses but at the expense of consistency in the data (Gill et al, 2008).

Additionally, a dataset should be unbiased, as to ensure that the data doesn't skew towards any results (Withrow). This can be dealt with in a variety of ways, such as sampling from random sources in order to get a greater range features, or resampling data from underrepresented classes (Kotsiantis et al, 2007).

Finally, a big issue in getting data is dealing with missing data. Missing data can reduce precision dramatically, and effectively reduces the number of samples that can be used (altexsoft).

When we designed our database, we tried to create features that would find a balance in variations that would be effective in predicting a user's joke taste. We also needed to collect the data quickly and efficiently, so we created a structured questionnaire for this. This allowed us to get our data quickly and ensure that there was no missing data (the questionnaire could not be submitted unless it was fully filled out), but limited how detailed user responses could be. Also, with only 117 students, too many options would create too many different possibilities for answers, which would be unhelpful when creating an algorithm, as detailed previously. Thus, we created categories to choose from for each feature, to ensure that samples were consistent and didn't leave room for potentially overfitting.

Due to the nature of the project and the class, there was an enormous bias towards a very specific type of person in our dataset. This was unavoidable, and should be taken in consideration when using the dataset on people not in the class.

Once data has been collected, it must be preprocessed in order to make it usable. The sample We attempted to create features that would find a balance in variations that would be effective in predicting a user's joke taste size is small enough that repeat jokes and users are easy to spot. However, the bigger issues are outliers that have useless data. Using dif-

ferent methods like extreme value analysis, proximity methods (like the k-means algorithm) for mean ratings and ratings for specific jokes, projection methods (like principal component analysis), and even just checking user rating variances, outliers for the subset of data can be found. We used some of these techniques to spot outliers, but we primarily studied the variances of user ratings to determine whether or not someone had submitted an invalid response.

The organization of a database is often referred as the database schema. The schema defines how data is stored within tables and the relationships between tables. There are two underlying types of databases: relational databases and non-relational databases. Relational databases model organizes data into one or more table and certain rows in a table has its own key which links to rows in other tables. Non-relational databases are databases that do not follow the rules of a relational model. There are benefits and disadvantages to both kinds of databases. Some advantages of using relational databases are structured data, data integrity, limitless indexing, strong SQL, and relationships; however, the disadvantages of relational databases are that it requires knowledge of the database structure for queries, lack of support for complex data types, and that it costs money.

Although both have their pros and cons, ultimately, we decided to use a relational database, as the structure fit well with the structure of our samples, which made it easier to store and filter through. We sacrificed ease of use for better data tracking.

At the end of all of this, we analyzed our dataset. We primarily wanted to see if the features we chose were effective for predicting user behavior. We did this in two ways: first comparing specific user features, and seeing if there are any correlations to how they rate jokes of different genres or categories; then, checking if we could predict what the variance of ratings would be for users with specific profiles.

0.2 Methods

(i) *Data Collection* - For data collection, we created three surveys. The first was for collecting user profiles. Users were asked to specify their gender (male, female, or other), age, country of origin, their major, their preferred joke genre (10 possible options), their preferred joke category (6 possible options), their favorite movie genre (10 possible options), and their favorite music genre (10 possible options). We also asked students to each enter a joke using a survey, which asked for the joke text, the joke subject, the joke genre (10 possible options) and the joke type (6 possible options). However, we did not receive as many jokes as we would have liked, so the team found more independently. Finally, we asked every user to rate every joke we had on a scale from 1-5. There was no missing data in jokes and users; however, not every user rated every joke.

(ii) *Preprocessing* - Before we released the joke rating survey, the team combed through the database for repeated or inappropriate jokes, and deleted them. Then, after the jokes had been rated, we first looked for users with too many joke ratings and deleted the repeat ratings. After this, we set bounds for outliers by observing that 81 out of 94 users had variances for their ratings between .5 and 2. However, we kept in users with extreme ratings variance, at the request of other teams. We also located users who did not rate every joke or even at least 90 percent of the jokes, but they were also kept by the request of other teams. Finally, there were some cases of users who had rated an invalid joke, and those ratings, as well as the joke, were found and removed.

(iii) *Database Design* - The database was split into three tables (Figure 1). The first table was for user profiles, and was named "JokeRater". It had a column for each feature we specified before, and every row corresponded to a user. The second table contained jokes, and was named "Joke". It contained columns for the joke's id, the joke's genre, the joke's category, and the joke's text, as well as which user submitted it. Every row corresponded to a joke. Finally, the last table was for each user's rating of each joke, and

was named "JokeRating". It had three columns: the first was the id of the user who rated the joke, the second was the id of the joke that was rated, and the last column was the rating.

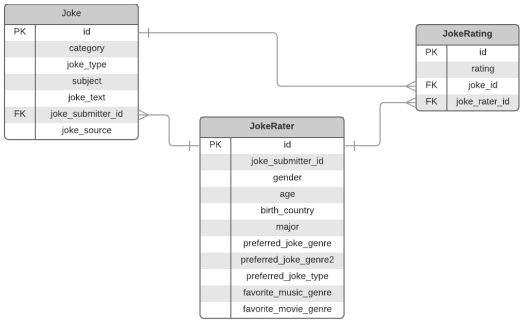

Figure 1: Structure of the Joke Database

(iv) *Analysis of Correlation* - The primary purpose of this dataset was to act as training data for an algorithm that would recommend jokes to users. Thus, it is important to test whether or not there is a significant difference between user's ratings for certain jokes based on their features. We did this by comparing how jokes of each genre were rated on average by users who stated they preferred that genre of joke versus users who did not state they preferred that genre of joke (Figure 6). Then, we compared users ratings for jokes of a genre they stated they preferred versus ratings of those same users for jokes that are of a different genre (Figure 7). We repeated this process with joke category and users' preferred category (Figures 9 and 10). With this data, we compared the difference in sample means for each genre and category against the null hypothesis that the difference was zero, and found p-values for significance, comparing the sample means of people who preferred the genre or category of the joke with people who didn't, and then comparing ratings for jokes people rated that were in the genre or category they preferred versus jokes that weren't (Figures 8 and 11).

(v) *Analysis of Predicting Variance* - The experiment is performed as a regression on variances (continuous) as the response and profile features (categorical, except "Age") as predictors. Note that the 2 users with zero-variance ratings were removed prior to model building. Because the variables must be one-hot encoded and very few are expected to be informative (due to having a low number of samples, for example), we employ both Ridge and LASSO regularization methods in order to determine the k-most informative user profile features.

The 92 users were split into training (66%) and testing sets (33%), and both prediction models were fitted. Lambda values were selected by testing values at [1.0 , 0.1 ,0.01, 0.001] under the criteria of Minimizing Mean Squared Error.

MSE of Ridge Regression: 0.587997003849

MSE of LASSO Regression: 0.49832605088

LASSO outperforms Ridge Regression in this case, and higher (and equivalent) values of lambda would cause Ridge to overfit and artificially inflate the MSE metric. The LASSO regularization coefficient was manually adjusted to 0.01 in order to avoid all-zero coefficients in results and to display an ordered set of predictors. Finally, the top 10 features were identified based on absolute value of model coefficients (Figure 12).

0.3 Results

In total, we were able to get data for 94 valid user profiles, and 152 valid jokes. The distributions of users' favorite joke categories and joke genres are shown in Figure 2 and 3. It is worth noting that the dataset was taken from a computer science course, which might explain why the programming genre was stated as the favorite for the most students.

Figures 4 and 5 are the distributions of jokes for each category and each genre.

Figure 6 compares the average rating of jokes of each genre by users who stated they preferred the genre and users who didn't, and Figure 7 compares the average ratings of jokes from the genre users preferred versus jokes not from that genre. It can be seen that

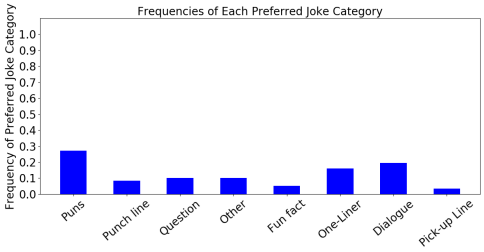

Figure 2: Frequencies of Stated Preferred Joke Category

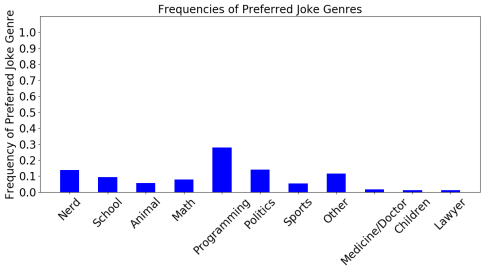

Figure 3: Frequencies of Stated Preferred Joke Genre

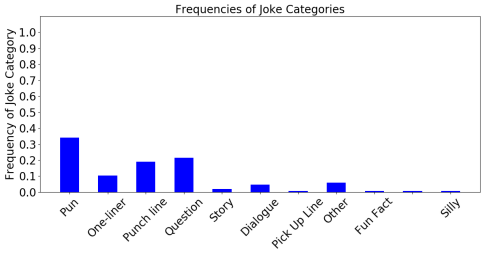

Figure 4: Frequencies of Jokes of Each Category

six out of the nine genres have a higher average rating when rated by users who stated that genre is their preferred genre. Figure 8 shows the p-values for hypothesis that the true differences are the same, and we only failed to reject the null for the Nerd and Politics genres with 99 percent confidence. Overall,

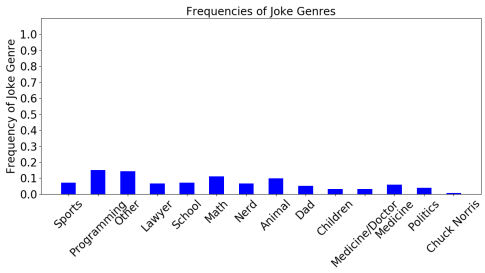

Figure 5: Frequencies of Jokes of Each Genre

across every class, the p-value is small, showing that there is a significant difference between the average ratings of users who prefer a joke's genre versus those who don't.

Meanwhile, the average rating of jokes whose genre is the user's preferred genre is only higher than user's ratings of jokes in other genres in four out of the nine genres. In all genres but Programming and Doctor/Medicine there are very high p-values, meaning the null hypothesis that the users have the same average ratings, regardless of whether or not joke is of the user's preferred genre, cannot be rejected. That said, the overall differences are still significant with a p-value of zero, which is very suspicious and possibly an error.

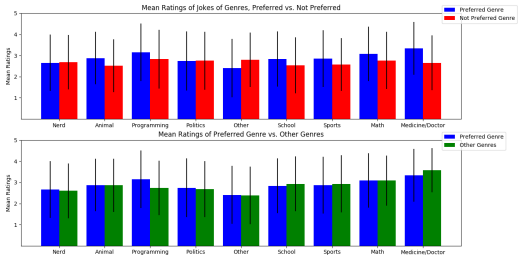

Figure 6: Figure 6 (Top): Average rating of jokes of each genre by those who preferred the genre and those who didn't. Figure 7 (Bottom): Average rating of jokes of user's preferred genre and average rating of jokes that were not a user's preferred genre.

	Pref vs. Not Pref	Jokes of Pref vs. Other Jokes
Nerd	0.47457000234	0.499999999067
Animal	0.0	0.499988894237
Programming	0.0	0.0
Politics	0.498166602808	0.5
Other	0.0	0.499999999525
School	0.0	3.630256124668e-257
Sports	0.0	0.0786147128946
Math	0.0	0.499680166422
Medicine/Doctor	0.0	8.105796644086e-16
Overall	0.0	0.0

Figure 8: p-values for each Difference in Means in the Genre Average Ratings

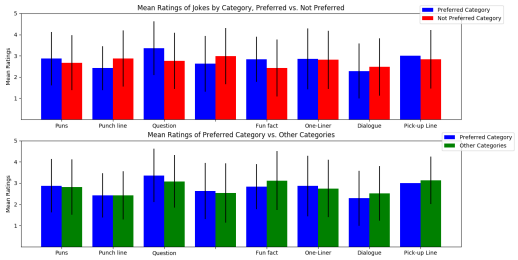

Figure 9: Figure 9 (Top): Average rating of jokes of each category by those who preferred the category and those who didn't. Figure 10 (Bottom): Average rating of jokes of user's preferred category and average rating of jokes that were not a user's preferred category.

Figure 9 compares the average rating of jokes of each category by users who stated they preferred the category and users who didn't, and Figure 10 compares the average ratings of jokes from the category users preferred versus jokes not from that category. It can be seen that five out of the eight categories have a higher average rating when rated by users who stated that category is their preferred category. Figure 11 shows the p-values for hypothesis that the true differences are the same, and we only failed to reject the null for the Fun Fact and One-Liner categories with 99 percent confidence. Overall, across every class, the p-value is small, showing that there is a significant difference between the average ratings of users who prefer a joke's category versus those who don't.

Meanwhile, the average rating of jokes whose category is the user's preferred category is only higher than user's ratings of jokes in other categories in four out of the eight categories. There are extremely low p-values for all categories but Punch-Line, Fun Fact, and unspecified, meaning that the differences in average ratings are significant. The overall differences are also significant with a p-value of zero.

The results indicate that the "Puns" joke type and the "Math" major are the most informative features and classes for making an accurate prediction on user variance, and they both correlate positively. Other features that are shared between the two methods are the Romantic Comedy genre, as well as the Rap Mu-

	Pref vs. Not Pref	Jokes of Pref vs. Other Jokes
Puns	0.0	1.828868718924e-41
Punch line	0.0	0.499999999998
Question	0.0	0.0
	0.478692220502	0.499999957099
Fun fact	0.378682233588	0.495033554194
One-Liner	0.499482886669	4.44610758884e-10
Dialogue	0.0	0.0
Pick-up Line	1.392244433151e-11	0.0
Overall	0.0	0.0

Figure 11: p-values for each Difference in Means for the Category Average Ratings

coef	feature	coef	feature
0.473161	preferred_joke_type_Puns	0.372458	preferred_joke_type_Puns
0.438239	major_Math	0.332417	major_Math
-0.409038	preferred_joke_genre_Math	-0.320729	favorite_music_genre_Rap
-0.403206	favorite_movie_genre_Romantic comedy	-0.272286	favorite_movie_genre_Romantic comedy
-0.316695	favorite_music_genre_Rap	-0.269505	preferred_joke_genre_Math
0.306770	favorite_movie_genre_Action	-0.257457	preferred_joke_type_
0.302013	preferred_joke_genre_Animal	0.246829	favorite_movie_genre_Action
0.298821	favorite_music_genre_Country	0.237287	preferred_joke_genre2_Politics
-0.284803	preferred_joke_type_	-0.226506	preferred_joke_genre_Politics
-0.274240	favorite_movie_genre_Superhero	0.223616	preferred_joke_genre_Other

Figure 12: Model Coefficients for each feature/class for determining user variance. Each feature was one-hot encoded

sic genre (correlated negatively with variance). The only features unique to each list are towards the bottom of the list, indicating that they may simply be uninformative features that were reported due to setting k=10. Despite how the models share similar top features, it is difficult to conclude whether they are effective models when the MSE for variance is nearly or above 0.5. Objectively, a 0.5 deviance from the true rating does not seem very significant, but as mentioned previously, this difference could be large for the way some users personally define their rating criteria.

0.4 Discussion

This report was designed to review the methods that were used to create the joke database, which was used for machine learning recommendation algorithms, and to determine whether or not the resulting dataset was effective. As was addressed in the previous section, we discovered that at least two of the features we used are effective in distinguishing users based on their profiles and the classes they fit into. We also were able to create a lasso regression model that was able to somewhat predict a user's variation. By these metrics, dataset is at least reasonably effective as a tool for creating a recommendation machine learning algorithm.

The next step, given more time, would be to analyze bias in this dataset. As was mentioned before, the students sampled were all in a Machine Learning class, which means the results likely skew toward the kinds of jokes students in that class would like. It seems probable that if the students sampled were all from, say, a philosophy class, the distribution of ratings might be different. This problem could also be solved simply by polling more students, or even polling the same amount of students but from many different backgrounds.

After that, it would be best to try to come up with better, more specific ways to classify jokes, as jokes tend to be nuanced and have a large variance within their genres or categories. On top of this, we only have two features for jokes (aside from the joke text) for classifying jokes.

Finally, there might be a better way to get user's preferred genre and category. This could be done by having users rate an initial pool of jokes of different genres and categories, and using those ratings to determine their preferred genre and category.

0.5 References

Gill, P, K. Stewart, E. Treasure, and B. Chadwick. "Methods of data collection in qualitative research: interviews and focus groups". BDJ 204, 291-295, 22 March 2008. https://www.nature.com/articles/bdj.2008.192

Kotsiantis, S.B., D. Kanellopoulos, and P. E. Pintelas. "Data Processing for Supervised Learning". World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering Vol:1, No:12, 2007. http://www.waset.org/publications/14136

Sáez, José A., J. Luengo, F. Herrera. "Noisy Data in Data Mining". Pattern Recognition, 2013. http://sci2s.ugr.es/noisydata#Noise_filtering_efficacy_prediction_by Data_Complexity_measures

Passantino, Frank. "5 Steps to a Squeaky Clean Database – Marketo". https://blog.marketo.com/2013/12/bad-data-in-bad- data-out- 5-steps- to-a- squeaky-clean-database.html

"Deleting Data from a Table". SQLToolPro. https://www.sqltoolpro.com/support/sql-delete

"Data Preprocessing". Techopedia. https://www.techopedia.com/definition/1-4650/data-preprocessing

Anita de Waard, Helena Cousijn, and IJsbrand Jan Aalbersberg. "10 Aspects of Highly Ef-

fective Research Data". Elsevier, 11 December 2015. https://www.elsevier.com/connect/10-aspects-of-highly-effective-research-data

Lehe, Lewis, and Victor Powell. "Principal Component Analysis". http://setosa.io/ev/principal-component-analysis/

Amatriain , Xavier. "In Machine Learning, What Is Better: More Data or Better Algorithms". KDnuggets Analytics Big Data Data Mining and Data Science. 2015. www.kdnuggets.com/2015/06/machine-learning-more-data-better-algorithms.html

Withrow, Scott. "Tips for Generating Meaningful Test Data". TechRepublic, 28 September 2001. www.techrepublic.com/article/tips-for-generating-meaningful-test-data/

Brownlee, Jason. "Discover Feature Engineering, How to Engineer Features and How to Get Good at It". Machine Learning Mastery, 26 September 2014. www.machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

"Preparing Your Dataset for Machine Learning: 8 Basic Techniques That Make Your Data Better". AltexSoft, 16 June 2017. www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better/

0.6 Author Contributions

Yoav Kaliblotzky – Statistical analysis methods, genre-mean graphs, and tables for p-scores.

Garrett Boseck – Editing, formatting, compiling the report

Nathalia Sandoval – Feature aspects in the introduction

David Liang – Database methods

Jasmin Adzic – Introduction preprocessing, and discussion future steps

Daniel Brown – Data collection methods

Nick Webb – Built and designed website

Shaifali Goyal – Frequencies of features and category-mean graphs

Yuji Mori – Variance prediction, methods, results and analysis

Steve Huynh – Preprocessing methods

Ronen Burd – Introduction database schema