Yelp Price Predictor

Christopher Hartman / Michael Pauleen / Jared Schifrien

Utilizing the Yelp Challenge Data Set, we designed and built a Machine Learning model to predict the most likely price bracket for a business with the given attributes

I. INTRODUCTION

Yelp is a business review website that allows any person to offer anonymous reviews, tips and images to help other consumers choose which businesses to patronize. While Yelp allows for an easy interface for consumers to choose which business will best suit their needs, it does not offer tools to help businesses utilize this data to help themselves. To fix this we created a model to predict the most likely price bracket for a business with the given attributes. This can then be used to determine the ideal pricing for a business based on its city, business type and other attributes found in the yelp dataset so that they can benefit from Yelp as well.

II. DATA COLLECTION

Our project utilizes the dataset given as a part of the Yelp Dataset Challenge, and more specifically we are looking at the business component of the dataset. There are about 77k businesses with 14 mandatory attributes and each can have up to zero to 10+ optional attributes in the "attributes" section. We have explored different techniques of cleaning and separating the data such as only using ⅘+ ratings, only using businesses that have more than 20 reviews, and only using restaurants for more stable correlations between quality and price vs success

III. DATA PROCESSING & FEATURE SELECTION

We started by preprocessing the dataset to narrow the businesses down to only restaurants using R. This is the only pre-processing we did for the first stage of our work. After that, we noticed that the Weka attribute selection algorithms also often recommended we use the "categories" feature, but that feature is a string of comma separated categories each restaurant could fall into, so almost every restaurant has a different value for it. We attempted to use Weka's stringToWordVector filter in order to create useful data from the feature, but were unable to create new data. We then used the python library scikit-learn to vectorize the data and effectively create a booean feature for each word that could potentially be used in the "categories" description. This gave us a total of 425 features including the already existing attribute features. We then removed attributes with very few (<20) true labels.

IV. MACHINE LEARNING MODEL & RESULTS

A. Intial Modeling

We began by trying to simply load the records (not including categories) into Weka and building a decision tree (J48) with all of the features. Because of the large number of features in the data set, Weka was unable to build the tree in a timely manner, so we tried to reduce the number of predictors used. Using just the "ambience" attributes and the category of the restaurant, the J48 tree has a prediction accuracy of 66% in 10-fold cross validation, a 14% improvement over ZeroR's prediction accuracy of 48%. The BayesNet classified 63% of instances correctly, while 5-NN classified 66% correctly.

B. Feature Number Reduction

Upon seeing the initial results, we were able to achieve using a limited feature set, we used several of Weka's attribute selection function to determine which features could provide the most information to our machine learning algorithms. Each algorithm included a different set of attributes but almost all of them included alcohol (what kind of alcohol is served: null, beer and wine, full bar, or none), attire (expected attire at the restaurant: null, casual, dressy, or formal), take out (nullable boolean), takes reservations (nullable boolean), and waiter service (nullable boolean). Using only these features, our decision tree was able to achieve 74.68% accuracy in 10-fold cross validation. Interestingly, once we reduced the feature space to only those few features, almost every machine learning algorithm performed similarly. Table 1 examines these results in more detail. As can be seen in the table, most machine learning algorithms perform with negligible differences. This is likely because of the low dimensionality of the data and the relatively straightforward relationships between the features and the prices. OneR uses only alcohol, and simply classifies "null" and "none" as "1" while classifying "beer and wine" and "full bar" as "2". In order to simplify our tree, we removed the "Take out" feature and achieved even higher accuracy: 74.81%.

ML Algorithm	10-Fold Accuracy	Weighted F-Measure
J48	74.68%	.736
NaiveBayes	73.66%	.733
BayesNet	73.68%	.734
Ibk	74.64%	.736
OneR	69.77%	.671

Table 1: Initial Results Without Categorical Information

C. *Implementation with Categorical Data*

At this point we incorporated in the categories data. We used the BestFirst+CfsSubsetEval attribute selection algorithm to determine the best attributes for our machine learning techniques. These features were the same as before, except with the addition of categories containing the words "sushi", "seafood", "Mexican", "fast", and "dogs". We then ran the same set of algorithms on the extended data, and our results are presented in Table 2. Comparing the two tables, there is an increase across all algorithms in both 10-fold accuracy and weighted F-measure. While this increase is small, it indicates that using the categories data has some value, and we imagine that if we were to clean it in a more advanced way, it could give a more significant increase in accuracy.

ML Algorithm	10-Fold Accuracy	Weighted F-Measure
J48	75.16%	.742
NaiveBayes	74.26%	.739
BayesNet	74.27%	.739
Ibk	74.84%	.739
OneR	69.77%	.671

Table 2: Initial Results With Categorical Information

D. *Final Solutions*

Since trees appeared to be doing best by a small amount, we decided to further experiment with different tree algorithms and settings. The most successful tree was a J48graft tree using the following command: J48graft -C 0.22 -M 2 -E. This tree has a 10-fold accuracy of 75.23 and a F-Measure of .742. After experimenting with trees, we also experimented using different bayes algorithms, the most successful of which was WAODE, which had 75.22% accuracy and a F-Measure of .744.

V. Conclusion & Future work

A. *Analysis*

The features consistently selected by Weka's subset evaluation all have intuitive theory that supports their usefulness for predicting restaurant price category. It makes sense that restaurants that offer alcohol, waiter service, and require classy or formal dress will tend to be more expensive than those that don't. We can therefore say that there is strong theory behind the solution given by the J48 decision tree without category data. Adding category data only results in marginal improvements because the most powerful predictive categories (e.g. "sushi", "seafood", "fast") tend to have attributes already in the model (e.g. seafood restaurants tend to require classy dress and offer alcohol and waiter service, while fast food restaurants offer no alcohol or waiter service), so there's little additional useful information provided by the category to predict price. The other issue with adding categorical data is each feature only applies to a relatively small subset of our data, but including enough features to hit a lot of the data leads to too many dimensions.

B. *Future Work*

The current feature set consistently has problems properly classifying price levels 3 and 4 across ML methods. The most accurate model (J48graft) classified level 3 restaurants correctly only 32% of the time, and failed to ever correctly predict level 4. While these categories are relatively small, future work could focus on improving their prediction accuracy. Possible methods for improvement would be finding alternative that could better identify classes 3 and 4. Additionally, a better vectorizing algorithm, such as SVM, might allow better use of the restaurant category feature as SVMs may outperform our current algorithms in higher dimensions.

VI. Acknowledgment

We would like to thank Professor Doug Downey and the EECS 349 TA's for their feedback and support. Furthermore, while all teammates had an equal share in machine learning project investigation we would like to personally thank Michael Pauleen for spearheading the data processing, Christopher Hartman for spearheading the categorization investigation and Jared Schifrien for spearheading the project website creation and final report.