Asia Pacific Journal of Information Systems
Vol. 29 No. 1 (March 2019), 35-49

ISSN 2288-5404 (Print) / ISSN 2288-6818 (Online)
https://doi.org/10.14329/apjis.2019.29.1.35

Applications of Machine Learning Models on Yelp Data

Ruchi Singh[a], Jongwook Woo[b,*]

[a] *Data Analyst, CISCO, USA*
[b] *Professor, IS, College of Business and Economics, California State University Los Angeles, USA*

ABSTRACT

The paper attempts to document the application of relevant Machine Learning (ML) models on Yelp (a crowd-sourced local business review and social networking site) dataset to analyze, predict and recommend business. Strategically using two cloud platforms to minimize the effort and time required for this project. Seven machine learning algorithms in Azure ML of which four algorithms are implemented in Databricks Spark ML.

The analyzed Yelp business dataset contained 70 business attributes for more than 350,000 registered business. Additionally, review tips and likes from 500,000 users have been processed for the project. A Recommendation Model is built to provide Yelp users with recommendations for business categories based on their previous business ratings, as well as the business ratings of other users. Classification Model is implemented to predict the popularity of the business as defining the popular business to have stars greater than 3 and unpopular business to have stars less than 3. Text Analysis model is developed by comparing two algorithms, uni-gram feature extraction and n-feature extraction in Azure ML studio and logistic regression model in Spark. Comparative conclusions have been made related to efficiency of Spark ML and Azure ML for these models.

Keywords: Machine learning, Yelp, Recommender, Predictive analytics, Text analysis

Ⅰ. Introduction

Yelp's website ("Yelp.com", n.d), is a crowd-sourced local business review and social networking site. Its user community is primarily active in major metropolitan areas in USA and Europe. The site has pages devoted to individual locations, such as restaurants, schools, dentists etc. Yelp can be accessed via iPhone, Android and web. Business owners can update contact information, hours and other basic listing information or add special deals. Yelp users can submit a review of their products or services, rate them using a one to five-star rating system, find events, lists and talk to other Yelpers. Platforms

*Corresponding Author. E-mail: jwoo5@exchange.calstatela.edu Tel: 13233432916

like Yelp play an important role in influencing in which business a consumer should spend their money or investors invest. To make decisions like these requires an analysis of the existing data and information. Yelp has a monthly average of 34 million unique visitors and more than 171 million reviews written ("Yelp.com", n.d). It generates vast data lake through user interaction which makes its analysis a challenge that can only be handled with Machine Learning models. With machine learning we have the capability to perform predictive analytics, pattern matching and text analysis etc. In recent times there have been a few powerful cloud platforms that enable us to perform machine learning computations affordably in terms of time and money. In this project we explore the Yelp data using Machine learning models using two cloud platforms, Azure and Databricks, to enable the platform with more intuitive features like, popularity prediction, recommender and words cloud.

1.1. Related Works

Yelp.com provides the free data and invites programmers to participate in yelp dataset challenge in order to come up with an algorithm which can predict the business rating efficiently. This acts as a huge motivation to participate in the challenge and take advantage of this opportunity to attempt to solve a real world problem using our knowledge of machine learning. Most of the work related to yelp business data is focused on rating prediction. Some of them have performed a detail analysis of business of a location, unlike this project. Our research does not take in account any particular location or business. For instance, Gwo-Hshiung Tzeng et al. (2002) concentrates on the criteria for a good restaurant location in Taipei. Whereas Tsung-Yu Chou et al. (2008) evaluates the importance of infrastructure cost and environmental factors responsible for setting up a hotel business. Predicting Usefulness of Yelp Reviews by Xinyue et al. (2016) uses MATLAB to perform language processing techniques for Yelp Review. We intend to use all the data to explore possible use of machine learning models to improve the usability of Yelp platform. Our work is a more general approach to bridge the gap between the business, customer and the available data.

Some of the popular machine learning projects that contributed in selecting the right algorithms for our project are as follows: Qu et al. (2010) extracted feature to perform sentiment analysis on Amazon.com reviews, using unigram model. Earlier Leung et al. 2007 used recommender to recommend movies to viewers based on the reviews. Ganu et al. (2009) propose a method to use the text of the reviews to improve recommender systems, like the ones used by Netflix, which often rely solely on the structured metadata information of the product/business and the star ratings. Fan and Khademi (2014) used a combination of three feature generation methods as well as four machine-learning models to find the best prediction of star ratings for the businesses. Carbon et al. (2014) used yelp dataset and investigated potential factors that may affect business performance. They have found that the review sentiment is one of the main factor affecting review ratings and hence need to be further investigated for accurate prediction. Jong (2011) also performed business rating prediction based on sentiment analysis. He has also compared the strength and weakness of different sentiment analysis models. Li and Zhang (2014) have done similar work, predicting star rating based on sentiment analysis of business review data. Most of these studies were focused on the star rating prediction.

Having gone through rich scholarly works we feel there is a need for a recommender to recommend business to users on the Yelp platform. Like Jong we could also predict the number of likes on a review tip based on the popular words. In addition to that we realized that prediction of the popularity of business will add value to all the previous works done on this dataset.

1.2. Motivation and Goals

This dataset is chosen due to constantly growing popularity of the application and the importance of data for businesses today. According to Yelp fact-sheet, as of 2017, the site has about 157 million monthly visitors with 127 million comments. Having an insight of the businesses, users and their actions can be very beneficial for businesses in terms of gaining competitive advantage and customer satisfaction.

As a small business owner, there is always concern over the reputation of its business and Yelp has certainly positioned themselves as a leader in identifying consumer thoughts and experiences while featuring businesses to help educate consumers on where to go, what to do and who to spend their money with. As Jeremy Stoppelman, the Yelp.com's co-founder and chief executive puts in his words "We put the community first, the consumer second and businesses third". We aim to shift some focus from the consumer to the business and use machine learning models to analyze the business for the betterment of the business owners and create features to promote the business.

An added advantage of using this dataset was a good understanding of the dataset and past experience of data analysis on this data. Previously, we had analyzed Yelp dataset to find insightful correlations between U.S. states, categories, reviews, seasonal ratings, etc. This project has been a continuation of our work in terms of applying our knowledge of Machine Learning algorithms for predictive analytics.

There are various cloud computing platforms that claim to leverage machine learning capabilities to solve business problems. We decided to use two leading public cloud computing platforms for this project, Azure machine learning studio and Databricks. This gives us an opportunity to explore the platforms for the yelp data and use them in a way that both can add value to our project.

Ⅱ. Background

2.1. Microsoft Azure Machine Learning Studio

Azure Machine Learning Studio enables to build machine learning models with great ease. It has a drag-and-drop interface that doesn't require any coding (although you can add code if you want to). It supports a wide variety of algorithms, including different types of regression, classification, and anomaly detection, as well as a clustering algorithm for unsupervised learning. Developing a machine learning model is an iterative process. As the various functions and their parameters are modified, the results converge until we have a trained, effective model. Because of this ease to build, test and iterate various predictive models in very little time on the same data set, we used Azure as our first platform to choose which model could work best for our data set. The performance of Azure Studio deteriorates considerably with increase in the size of data and the number of modules per experiment. For this reason we worked with only a sample of original data in Azure.

2.2. Databricks

Databricks provide a just in time platform on top of Apache Spark that empowers to build and deploy advance analytic solution. It is orchestrated with open source Spark Core with underlying general execution engine which supports a wide variety of application, Java, Scala and Python API for the ease of development. It had integrated workspace in the form of notebooks and dashboards. It is easy to start a cluster with a version of Spark of choice and instance of type- on demand, spot or mixed. Databricks gives enough flexibility to scale and build. The notebook can be used to code as well as visualize the data and results simultaneously. It is easy to work in teams in collaboration by giving access to the notebook with edit control. The key feature that makes Databricks our ultimate platform for this project is performance. Databricks provides a series of performance enhancements on top of regular Apache Spark. These include caching, indexing and advanced query optimizations. Its runs eight times faster than any other platform and is capable of consuming any amount of data.

2.3. Preliminary Work on the Machine Learning Models

In our project we have implemented seven algorithms in AzureML and four in SparkML. Most of our models are based on our existing works but with a different scope. These smaller projects have been critical in forming our understanding of how to use these models aptly for the Yelp dataset. The below discussion describes of how we implemented these model on different datasets prior to this project.

2.3.1. Recommender

Our Matchbox Recommender module is based on the lab work where we constructed and evaluated a recommender using a sample of user movie rating data. Movie Ratings and Movie Titles datasets were joined in AzureML. The four score recommenders recommend different metrics: a) item recommendation, b) rating prediction, c) similar items and d) similar users. After this step, each metric is evaluated by Evaluate Recommender module and the success of the model is determined.

Our SparkML recommender is based on Collaborative Filtering project that uses movie titles and rating datasets. Similar to AzureML, the datasets were joined in Spark ML. We have used ALS (Alternating Least Squares) algorithm to build the recommender. Additionally, we've defined parameters and used fit method to train the model. Then we test the model to see the recommended movie for each user. The evaluation is conducted to show us the RMSE.

2.3.2. Classification

The Two class Logistic Regression and Two class Boosted Decision Tree are used in the project based on the study of the prediction of flights delay. In the prediction of flight delay all the flights with delay time more than 15 minutes were classified as delayed and flights delayed less than 15 minute were classified as not delayed. The logistic regression is considered to be the best in fast training and linear classification model whereas boosted decision tree is known for its accuracy, fast training and large memory footprint, apt for big data. The implementation of two class logistic regression in Spark ML using Train Validation Split and Binary Classification Evaluator was earlier tested on the flight dataset. The regularization para meters used to avoid the imbalances in

data are 0.01, 0.5 and 2.0. The PramGridBuilder is used to generate all possible combinations of regularization parameter, max iterator and threshold.

2.3.3. Clustering

Our K-Means Clustering model in AzureML is based on the lab work where we trained and evaluated a k-means clustering model for the Forest Fire dataset to cluster high priority/big forest fire and separate them from the smaller fires. After the data was cleaned and normalized, two-cluster and three-cluster models were created and trained. A comparison was made to see which one was a better choice. In this case, the two-clusters worked better with clear separation in the clusters and the three clusters didn't have distinct separation between the clusters.

Our SparkML work was based on clustering model which clustered customers into 5 clusters. The customer dataset had several customer attributes varying from age to numbers of cars. The model used the Income as the basis of the clustering.

2.3.4. Text Analysis

The text analysis part of our project aimed at finding the most frequently used word in the review tip and predict the likes based on the text. Each word in the text processed to have a vector representation. For training a classifier the term-frequency (TF), is multiplied by the inverse document frequency, and the TF-IDF scores are used as feature values. N gram and uni-gram vector representation of data are tried to find out which works best with our data. Using R script we created the word cloud of the most frequently used words in the text. In spark we used the text to predict the number of 'likes' customer will give to the text using the multi-class logistic regression as the likes are from 1-10.

Ⅲ. Methodology

3.1. Data Description

The dataset has been downloaded from Yelp Open Dataset Challenge ("Yelp.com", n.d). This is a subset of Yelps actual businesses, reviews, and user data released for use in personal, educational, and academic purposes. It is rich in information about 42,153 local businesses, in various cities from 14 states in U.S and 4 other cites that include: Edinburgh (U.K), Karlsruhe (Germany), Montreal (Canada) and Waterloo (Canada). The yelp data set is of size 90MB in JSON file format with 334,335 rows and 108 columns. The review tip dataset is of size 70MB in JSON file format with 591,865 rows and 7 columns.

The two files used in this project are the 'business.json' and 'tips.json'. Business related data in 'business.json' file has a JSON object which specifies the business ID, its name, location, stars, review count, opening hours, etc. A tips (short text review) file has a JSON object which specifies the business ID, user ID, tips text and likes. Data wrangling is done to convert the JSON files to CSV. There are 452 categories of business in the business data file that has been grouped into broader categories like Education, Entertainment, Finance, Food, Government, Medical, Real Estate, Services and Shopping for the ease of classification. The columns in consideration for analysis in the business csv are business ID, category, review count, stars and 80 attribute columns describing the business (e.g. Accepts insurance, by appointment only, accepts credit cards, parking garage, parking street, parking validated, parking lot, parking valet etc.). In review tip csv we shall take

<Table 1> Column Name and Description for Business.json and Tips.json

Column name	Column description
business_id	Unique business id
name	Business name
neighborhood	Neighborhood
city	City
state	State
postal_code	Postal code or Zip code
latitude	Latitude
longitude	Longitude
stars	Number of stars
review_count	Number of reviews
is_open	Is it open?
categories	Business category
text	User tips about the business
date	Date of the tip
likes	Number of Likes
business_id	Unique business ID
user_id	Unique user ID

<Table 2> Platform Specification

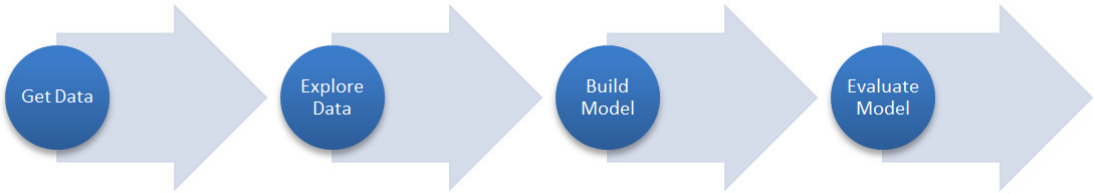
Azure		Databricks (Spark 2.1)	
Memory	10 GB	Memory	6 GB
No. of nodes	1	No. of nodes	1 Driver (0.88 cores, 1 DBU), 0 Worker
No. of modules /experiment	100	File System	Databricks file system

in account the business ID, user ID, text and likes. The two datasets are joint by the common column business_id (Primary key).

3.2. Hardware Specification

Hardware is an important consideration when it comes to machine learning workloads. Training a model to recognize a pattern or text analysis requires major parallel computing resources, which could take days on traditional CPU-based processors. In comparison, powerful graphics processing units (GPUs) are the processor of choice for many AI and machine learning workloads because they significantly reduce processing time.

For this project, we have used Microsoft Azure Machine Learning Studio and Databricks community edition to implement Spark ML using Python and R programing languages. Below are the hardware specifications offered by these public cloud comput-



<Figure 1> Machine Learning Project Processflow

<Figure 2> Azure ML Studio Processflow

ing platforms:

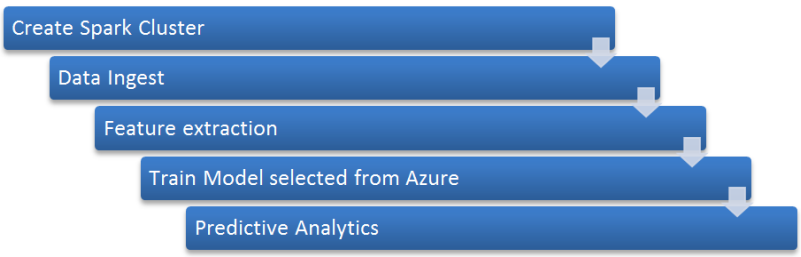
3.3. Machine Learning Workflow

Machine Learning projects typically follow a process to solve the problem in hand. The workflow of such a project can be depicted as below <Figure 1>. Once a good understanding of the problem is established the dataset is obtained and the feature columns are determined which could influence the target variable. This step also helps in creating new features if required. Not everything can be determined by looking at the data. Further exploring the data and studying the variables (skewed, missing, zero variance feature) so that they can be treated properly. In this step we also impute missing / null values remove space, irregular tabs and correct date time format. In the next step we create new feature depending on the machine learning algorithm to be used and problem in hand. Then we choose a suitable algorithm and we train the model on the given data set. In the last step we evaluate the model's performance by an error metrics. We also evaluate the feature importance of the selected feature and shortlist the best variables and retain the model.

Dataset for this module consisted of Yelp Local Business file and Tip file which were initially converted from json to csv format. The both csv files were joined in AzureML using business_id as the common field. In the next step, we prepared and transformed the data by removing duplicate rows, cleaning missing values, selecting our target columns.

3.4. Process in Azure

Implementing the above workflow of Machine learning project in Azure ML studio is easy to set up as an experiment using drag-and-drop modules preprogrammed with predictive modeling techniques. A small portion (only 20% of the total no. of rows) of the original dataset is used as data source here. To prepare the data we have mostly used the modules Select Columns in Dataset, Clean Missing Data. Next we chose and applied learning algorithms like Matchbox Recommender, Two Class Logistic regression, Two Class Boosted decision tree, K Means 5 cluster, K Means 3 cluster, N-gram and Unigram. We used the module Split Data in the ration 2:3



<Figure 3> Databricks Process-flow

to Train data and rest to score the model. Lastly we used the Evaluate Model to select the model with a better score to be implemented in Spark on Databricks.

3.5. Process in Databricks

Once the Machine Learning algorithm is selected to be implemented on Databricks, the workflow followed was as below. Create cluster in Databricks and load the complete dataset using Spark. We used One-Hot Encoding to convert into binary vectors as a part of feature extraction to improve prediction accuracy. To build model selected from Azure ML studio, we tune the parameters with *ParamGrid* and 5 folds Cross Validation. Lastly to evaluate from the Cross Validator using the test set. The default matric used is area under ROC.

Ⅳ. Results and Discussions

Using the above mentioned related works as templates for our models, we have conducted a thorough analysis of the Yelp datasets, cleaned and prepared our data and build our models.

4.1. Matchbox Recommender

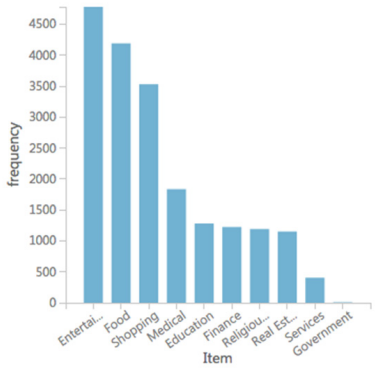
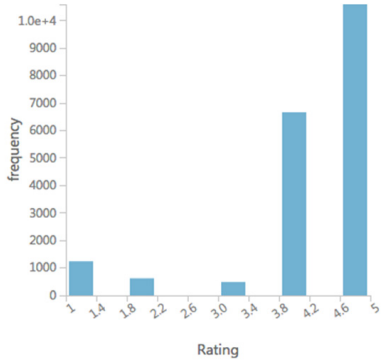
The goal of the recommender is to provide Yelp users with recommendations for business categories based on their previous business ratings, as well as the business ratings of other users. Moreover, the model has a feature to predict the future ratings by user for a category.

SQL transformation was conducted to select the average number of stars that each user has given to a category. After column were selected, the dataset was split into training and testing fractions by .75 to .25 ratio. After the split, the training fraction is connected to Train Matchbox Recommender module and test fraction to four Score Recommender modules. Each of the four score recommenders represent different metrics: a) item recommendation b) rating prediction, c) similar items and d) similar users. The <Table 3> and <Table 4> depict the visualizations of the model for Rating Prediction and Item Recommendation respectively.

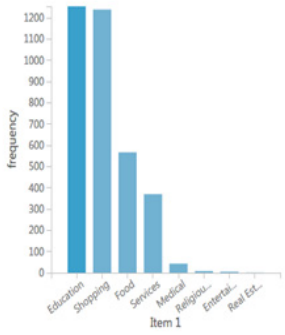
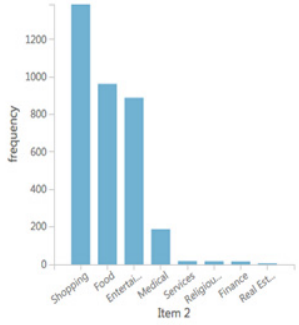
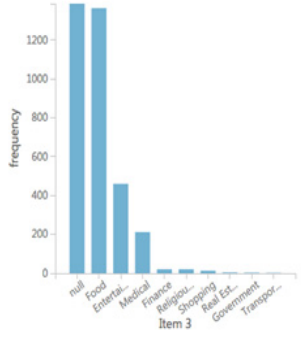
After this step, each metric is evaluated by Evaluate Recommender module and the success of the model is determined. <Table 5> shows the evaluation result for the above-mentioned metrics.

The Item Recommendation is successful since the NDCG is close to 1. This indicates that the recommendations made using this model is very good (NDCG = 0 indicates that model would recommend items for which there is no user feedback). However, the Rating Prediction RMSE of 1.05 could be

<Table 3> Rating Recommendation

User	Item	Rating
		
User 1	Services	5
User 1	Food	4
User 1	Shopping	5
User 2	Entertainment	4
User 2	Shopping	5
User 3	Food	2
User 3	Entertainment	2
User 3	Shopping	5

<Table 4> Item Recommendation

User	Item 1	Item 2	Item 3
			
User 1	Services	Shopping	Food
User 2	Shopping	Entertainment	N/A
User 3	Shopping	Food	Entertainment
User 4	Education	Shopping	Food
User 5	Education	Shopping	Food
User 6	Food	Entertainment	N/A
User 7	Education	Shopping	Food
User 8	Education	Shopping	Food

<Table 5> NDCG for Item Recommendation and RMSE for Rating Prediction

Metric	Score
NDCG(Normalized discount Cumulative Gain)	0.982998
RMSE(Root Mean Square Error)	1.04649

improved. Therefore, a different recommendation model was conducted in SparkML to see if the number could be improved.

4.2. Collaborative Filtering Recommender

Our SparkML recommender is based on Collaborative Filtering project, that uses a dataset called reviewstar, which we created in AzureML. It represents the cleaned and transformed data that we processed and downloaded from AzureML and contains four columns: user_id, category, review_count, and stars. User_id and category are selected as features and stars is selected as label. The dataset is split to train and test fractions by 0.7 to 0.3 ratio. We have used ALS (Alternating Least Squares) algorithm to build the recommender. Additionally, we've defined parameters and used fit method to train the model. Then we test the model to see the recommended category for each user.

The above figure depicts the output of the Prediction table and the RMSE of the model. Compared to the RMSE in AzureML (<Table 6>), the RMSE in SparkML (0.596) is much lower, thus making SparkML the better choice.

4.3. Classification Models

To predict the popularity of the business we defined the popular business to have stars greater than 3 and unpopular business to have stars less than 3. To select the feature columns and have the accurate prediction for the popularity of the business we chose the food category. All the attribute columns related to the food category like good for breakfast, lunch, dinner, take out, delivery, parking, alcohol, Wi-Fi, waiter service, wheelchair and noise level and considered as feature columns. We categorize all the columns for the classification model.

In Azure, we take a sample of the dataset (10%) and train it for Two Class Logistic regression and Two Class Boosted Decision Tree. The logistic regression is used to find the probability of the two states of the target variable. Whereas the boosted decision tree is an ensemble learning tree to make the prediction. The evaluation of both the models give an AUC of 0.72 and 0.73 respectively. This is a very good score with an accuracy of 0.8 and recall

<Table 6> Rating Prediction and RMSE

User	Category	Prediction	trueLable
User 1	1	5.0	5
User 2	1	4.0	5
User 3	1	4.0	4
User 4	1	4.0	5
User 5	1	4.0	4
Root Mean Square Error (RMSE):		0.595917109696	

of 0.9 for both the models. Thus, both the models are to be suitable for the prediction of the popularity of the business.

Implementation of Two Class Logistic Regression in Spark using Binary Classification Evaluator on the complete dataset gives a value of 0.7 and AUR of 0.617. The AUR value of model has dropped in Spark ML due to training the logistic regression for the complete dataset as oppose to the sampled dataset. The dataset had a very small percentage of food business having less than 3 stars. Thus, the model does not train well to predict unpopular business. The result could improve if the dataset was balanced with popular and unpopular business.

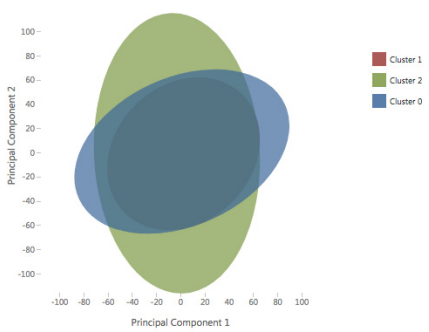
4.4. K-Means Clustering

The clean food category data is used for 3 cluster and 5 cluster K-Means clustering model in Azure with the feature columns selected for classification model. Following are results of training the dataset.

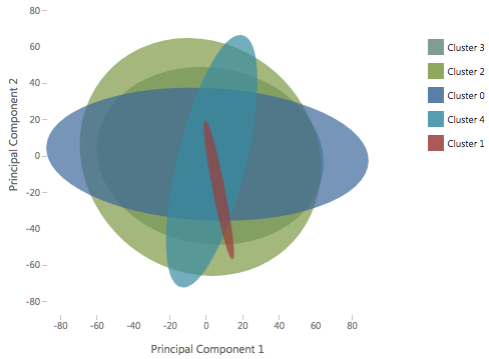
In SparkML we used Food data which we cleaned and transformed in AzureML. The table includes stars, review_count, and categorized columns describing whether the restaurant is good for breakfast, dinner, lunch or take-out. The latter attributes describing the restaurant are chosen as features. The model clusters the restaurants based on the count of reviews each restaurant has received. For example, review_count from 1 to 80 were put in one cluster, 81-200 in another, etc. By comparing the distance from center in SparkML and AzureML, we got a better result in SparkML (9.77 vs. 11.72).

4.5. Text Analysis

The text analysis is done by comparing two algorithms, uni-gram feature extraction and n-feature extraction in Azure ML studio and logistic regression model in Spark. The tip.csv dataset has six columns including text, likes, business_id, user_id, date and type. The operations are done on text and likes only because we only need the text and number of likes on that text. Data has been cleaned in AzureML studio. N-gram feature extraction includes the occurrence frequency of 2-gram with hashing bits 15 in the text instance. We have considered 1000 features. The uni-gram TF-IDF identifies the words that are frequent in document but rare in the corpus. R scripts were written to remove stop-words, uniform resource locater (URL), special characters, duplicates which is called preprocessing text. The word cloud has been



<Figure 4> Three Cluster Model



<Figure 5> Five Cluster Model

<Figure 6> Work Cloud of Most Frequent Words

created using r script which represents the most frequent words, relevant words with the polarity of negative and positive. The dataset is split in the ratio 70:30. We used Tune model Hyper-parameters finding the optimum settings for a model. Out of these two models n-gram feature extraction gave good score of 0.769. The accuracy was 0.931 and precision was 1. The frequency of the relevant and useful words showed the customers sentiments and business satisfaction. 2^15 = 32,768 entries of these words exist in the file.

In AzureML studio, we extracted the high frequency words, so it was necessary to predict likes to a piece of text written by users. In Spark, we used the classification model to predict likes. Spark has various libraries HashingTF, Tokenizer, StopWordsRemover, pipeline, etc. Some SQL queries were used to access the dataframe. We used pipeline algorithm with Tokenizer to split the text into individual words, StopWordsRemover to remove common words such as “a” or “the” that have little predictive value. A HashingTF class to generate numeric vectors from the text values.

A Logistic Regression algorithm to train a binary classification model. So, the stop words are removed and likes are predicted with respect to relevant text. The pipeline is used as an estimator and run with fit() method on training data to train the model. Classifiers are created with confusion matrices which gave true positives 2055.0, precision 1.0 and recall 1.0. The BinaryClassificationEvaluator class evaluator is used to measure the area under a ROC curve for the model which was 1.0 and is an ideal value. So the logistic regression model gave perfect score for predicting the likes.

Azure ML and Spark ML are powerful platforms for machine learning. Azure ML studio gave the flexibility to try various machine learning algorithms on the sampled dataset with simple drag and drop. Spark ML could train the model with large data volumes in relatively small time. This project could otherwise be four separate projects of full length.

<Table 7> Result Table

Azure	Evaluation	Spark	Evaluation
Matchbox Recommender	RMSE=1.04	Collaborative Filtering Recommender	RMSE=0.595
Two Class Logistic Regression	AUC=0.725	Two Class Logistic Regression	AUR=0.617
Two Class Boosted Decision Tree	AUC=0.738		
K-means 5 cluster	Max dist=11.7	K-means 5 cluster	Max dist=9.77
K-means 3 cluster	Max dist=33.4		
N-gram	AUR=0.76	Logistic Regression	AUR=1
Uni-gram	AUR=0.5		

Ⅴ. Conclusions and Future Work

Having used two powerful machine learning cloud platforms in this project, Azure and Databricks, we learnt the potential and limitations of both and how they complement each other. Azure ML Studio is a convenient (drag and drop) platform to implement various data science models to decide which one fits best and then fine tune the features but the data size needs to be a sample (few megabytes in size). Databricks is a more scalable platform and can be used for any data size with the effort to code in order to implement data science models. This strategy of using two platforms to our advantage saved many hours of coding.

Using machine learning, Yelp.com can be empowered with features like recommendation engine and word cloud. The recommender feature can help business retain their customers by providing them the tailored suggestion based on their previous rating. The Yelp data can be used to classify business as popular and unpopular, to understand what percentage of business in a category are successful. It can help investors invest in right business. The business is intuitively grouped based on their range of review counts and their attributes. These businesses have similar popularity amongst customers therefore they have almost same number of review counts. Another interesting feature that can be added to the existing platform is a Word cloud. It can act as a helpful tool to suggest reviews most frequently used words that can be used in their review. Text analysis of small text like Yelp using Machine learning enables us to predict the number of likes for a business based on the short review tip written by the users for that business. It is evident from the accuracy of sentiment analysis of reviews that it is a promising way to predict if the consumer likes or dislikes a business.

There is immense potential in using machine learning algorithms to filter fake reviews created by individuals or business owners for themselves or their competitors. This is a much needed analysis for this project as consumer reviews is the most important part of decision making on Yelp and committing review fraud is undermining business. An extension to this project will be unsupervised classification of reviews as fake and not fake. It will be rewarding to also analyze what impact, filtering the fake reviews can have on the business.

<References>

[1] AzureML Team for Microsoft (2015). *Text Classification*. March 18, Retrieved from https://gallery.azure.ai/Experiment/Text-Classification-Step-1-of-5-data-preparation-3

[2] Carbon, K., Fujii, K., and Veerina, P. (2014). *Applications of Machine Learning to Predict Yelp Ratings*. Stanford Univ., Stanford, CA.

[3] Chou, T. Y., Hsu, C. L., and Chen, M. C. (2008). A Fuzzy Multi-Criteria Decision Model for International Tourist Hotels Location Selection. *International Journal of Hospitality Management, 27*(2), 293-301.

[4] Fan, M., and Khademi, M. (2014). *Predicting a business star in yelp from its reviews text alone*. arXiv preprint arXiv:1401.0864.

[5] Ganu, G., Elhadad, N., and Marian, A. (2009). *Beyond the stars: Improving rating predictions using review text content*. WebDB.

[6] Jong, J. (2011). *Predicting Rating with Sentiment Analysis*. Stanford Univ., Stanford, CA.

[7] Leung, C. W., Chan, S. C., and Chung, F. (2007). Applying Cross-level Association Rule Mining to

Cold-Start Recommendations. *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Workshops*. pp. 133-136. Silicon Valley, California, USA.

[8] Li, C., and Zhang, J. (2014). *Prediction of Yelp Review Star Rating using Sentiment Analysis*. Stanford Univ., Stanford, CA.

[9] Liu, X., Michel, S., and Nan, Z. (2016). *Predicting Usefulness of Yelp Reviews*. Stanford Univ., Stanford, CA. Section 3.1.

[10] Qu, L., Ifrim, G., and Weikum, G. (2010). The bag-of-opinions method for review rating prediction from sparse text patterns. *Proceedings of the 23rd International Conference on Computational Linguistics* (*COLING 2010*), 913-921.

[11] Tzeng, G. H., Teng, M. H., Chen, J. J., and Opricovic, S. (2002). Multicriteria selection for a restaurant location in Taipei. *International Journal of Hospitality Management, 21*(2), 171-187.

[12] Yelp.com (2014), *Yelp Dataset Challenge*. October 15, 2014. Retrieved from http://www.yelp.com/dataset_challenge

◆ About the Authors ◆

Ruchi Singh

Ruchi Singh received her Master's in Information systems from California State University, Los Angeles and is currently working as Data Analyst. She is a data enthusiast, has worked on several data exploratory projects and presented her work in various conferences in USA and abroad. She envisions to use her knowledge and passion to solve real-life problems using Data Science. She has a Bachelors in Computer Science Engineering and 5 years of IT industry experience in India.

Jongwook Woo

Dr. Jongwook Woo received his Ph.D. from USC and went to Yonsei University. He is a Professor at CIS Department of California State University Los Angeles and serves as a Technical Advisor of Isaac Engineering, Council Member of IBM Spark Technology Center and as a president at KSEA-SC. He has consulted companies in Hollywood: CitySearch, ARM, E!, Warner Bros, SBC Interactive. He published more than 40 papers and his research interests include Big Data Analysis and Prediction. He has been awarded Teradata TUN faculty Scholarship and received grants from Amazon, IBM, Oracle, MicroSoft, DataBricks, Cloudera, Hortonworks, SAS, QlikView, Tableau. He is a founder of Hemosoo Inc and The Big Link.

Submitted: July 17, 2018; 1st Revision: December 11, 2018; Accepted: January 14, 2019