Big Data Management and Analytics

CS 6350.0U1 Final Project Report

YELP DATASET ANALYSIS

Team Members:

Tulika Mithal (txm172030)

Nadita Koppisetty (nxk174230)

Introduction and Problem Statement

Yelp has a large volumes of data about various businesses. This data includes useful information like customer reviews, number of checkins etc. But this data is of no use to the business owners in its raw form, since they cannot go through all the data manually and come to any conclusion. Example, they cannot go through gigabytes of review text and find any insight. We require techniques to process this data and gather some insights through it. Machine learning techniques can help us in finding patterns in the data and making predictions. However, ML alone would not suffice, since the data is too huge to be computed in a standalone manner. Here, big data techniques come to our rescue. In this project, we have tried to use ML and Big Data techniques to analyse the yelp data for our specific use cases.

Dataset Description

Dataset used is the yelp open dataset made available as a part of yelp dataset challenge. It is a subset of Yelp's businesses, reviews and user data made open for use in academic purposes. Dataset consists of 6 JSON files. Each file is composed of a single object type, one JSON-object per-line.

File Name	Number of rows	Description
business.json	174,000	Contains business data including location data, attributes, and categories.
review.json	5,200,000	Contains full review text data including the user_id that wrote the review and the business_id the review is written for.
user.json	1,300,000	Contains user data including the user's friend mapping and all the metadata associated with the user.
tips.json	1,100,000	Contains tips written by a user on a business. Tips are shorter than reviews and tend to convey quick suggestions.
checkin.json	174,000	Aggregated check-ins over time for each of the 174,000 businesses.

Tools and Languages used:

1) Spark Mllib
   a) Pipelines
   b) Classification
   c) Clustering
   d) Feature
   e) Tuning
      - Cross Validator
      - ParamGridBuilder
   f) Evaluation
   g) Latent Dirichlet Allocation (LDA)
2) Scala
3) IntelliJ

Preprocessing Techniques Used:

1. Removed review data with NULL 'text' column value.

2. Tokenized the review text into individual terms.

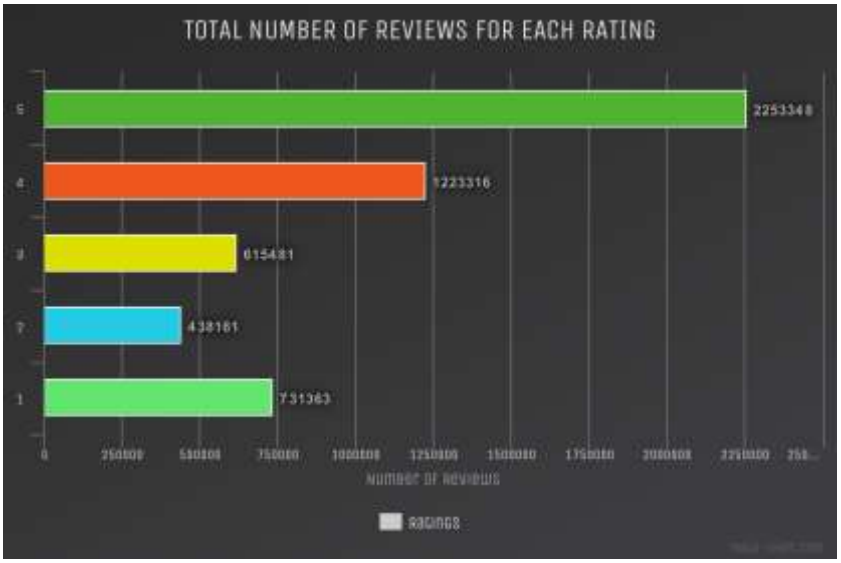
3. Removed stopwords that don't carry much meaning.

4. Converted the collection of filtered tokenized documents to vectors of token counts using **CountVectorizer** which is then passed to algorithm – LDA for topic modelling use case.

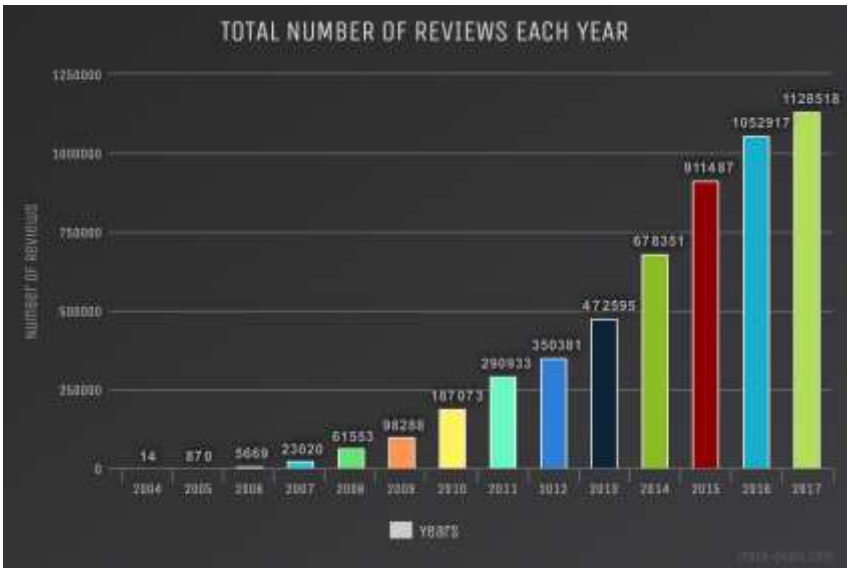
5. For star rating prediction use case, converted the collection of filtered tokenized documents to fixed length feature vectors using **HashingTF** and then used **IDF (Inverse document frequency)** estimator to scale the features i.e. it down-weighting features which appear frequently in a corpus.

Exploratory Data Analysis

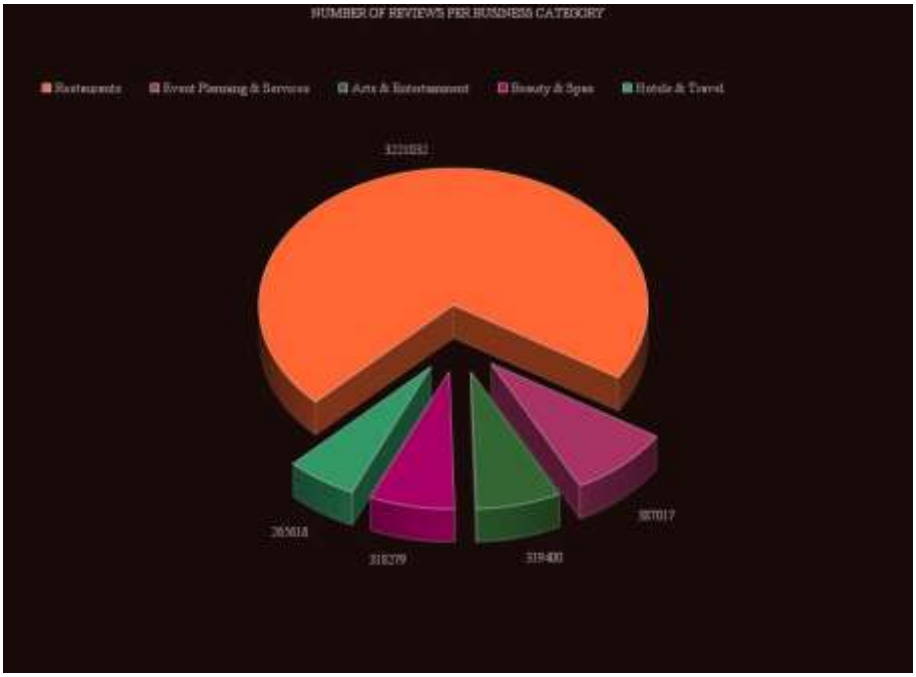
a) In order to analyze the distribution of review text data, we found the count of reviews grouped by star ratings and came to the conclusion, that the count of reviews with 4 and 5 star ratings are comparitively much higher in number than 1,2 and 3 stars review counts. This analysis is beneficial as it gives us an insight that our model would be biased towards higher star rating, when trained on this data.



b) We analysed the annual increase in number of reviews and found that there has been almost a 1000 fold increase in review counts from 2005 to 2017. This shows the growth of yelp as a platform.



c) On analysing the distribution of various business categories data, we found that the maximum reviews are regarding ‘Restaurants’ business category. Thereby, we have focused our analysis on Restaurants data.



Use Case 1 *(Topic Modelling of Reviews text)*

Yelp data consists of customer reviews for individual businesses. This information is of great value to the business owner if properly analysed.

Our Approach:

a) Extracted the restaurants data in **Phoenix** city of **Arizona** state.
b) Further narrowed down upon the restaurants with categories: **Italian & Pizza.**
c) Once we got the filtered data in step (b), we found the resaturant with the highest average rating and the one with the lowest average rating.
d) Did topic modelling for the review text of both, the best and worst restaurants.

Algorithm Used:

We used **Latent Dirichlet Allocation** algorithm for topic modelling. LDA is a clustering technique that extracts the abstract topics that occur in a collection of documents. In our case, we are trying to find out what topics customer's are talking about in their reviews, with respect to the best and the worst restaurant.

After analysis:

1) Best Restaurant : **Tommy's Place** (4.7 avg rating)
2) Worst Restaurant: **Domino's Place** (1.0 avg rating)

Results:

a) Topic 1 for Best Italian Pizza Restaurant (Tommy's Place) in Phoenix, Arizona. Below table shows the most important terms in the topic with their weightage.

Term	Weightage
great	0.02892086218588004
food	0.020707523605243985
place	0.012971499982852495
tommy	0.009936323035861803
best	0.009719789727670985
italian	0.008921053429635652
staff	0.008642294410709336
definitely	0.008343003504557328
owner	0.007357856211675605
back	0.0072465188399823835
service	0.006997967680902337
love	0.006921702200826057
go	0.006108565263624186

b) Topic 2 for Best Italian Pizza Restaurant (Tommy's Place) in Phoenix, Arizona. Below table shows the most important terms in the topic with their weightage.

Term	Weightage
place	0.01904033963453325
food	0.01250742060596976
really	0.012162278395580322

new	0.010267307709315016
good	0.009945517507128764
italian	0.009515826095622638
restaurant	0.009311040023327085
sauce	0.007802592924072879
back	0.007791109004882142
like	0.007777539316599083
great	0.0076250748852572285
chicken	0.007483914362759061
tommy's	0.007417718417695721
pasta	0.0070967758532754005

c) Topic 1 for worst Italian Pizza Restaurant (Domino's Place) in Phoenix, Arizona. Below table shows the most important terms in the topic with their weightage.

Term	**Weightage**
call	0.06787429955174064
like	0.02681194918420738
employee	0.02681194918420738
another	0.02681194918420738
dominos	0.02681194918420738
again	0.02218549895906914
customer	0.022185460304550673
hang	0.02218545742335552
guy	0.02218544898754643
hope	0.015310310131577682

award	0.013154907717118717
difficult	0.013154907717118717
reasonable	0.013154907717118717
rating	0.013154907717118717

d) Topic 2 for worst Italian Pizza Restaurant (Domino's Place) in Phoenix, Arizona. Below table shows the most important terms in the topic with their weightage

Term	Weightage
pizza	0.029461482316293395
ordered	0.026633248180641528
products	0.026633248180641528
location	0.026633248180641528
pre	0.026633248180641528
food	0.026633248180641528
back	0.026098625543021208
get	0.026098625543020438
hold	0.02518417254151605
times	0.025184172521765292
called	0.025184172512583355
multiple	0.02518417250457788
gets	0.012989049383106155
quality	0.012989049383106155

Conclusion:

We see that the topics extracted from the review text provides great insights to the business owner. The topics for the best restaurant suggest that the people really like the Italian stuff and specifically pasta.

The topics for the worst restaurant can help the business owner to figure out why people are rating the restaurant low. The results show that the people are frustated with the customer call service. This insight could help the restaurant owner to work upon the shortcoming and increase his/her business, which otherwise would not have been possible with just the huge amount of review text.

Use Case 2 *(Star rating rediction based on review text)*

It is tedious task for a customer to star rate the restaurant based on his review. Also, the ratings given by the customer might not be accurate as there is a lot of ambiguity. In our project we have tried training the review- star rating data using Machine learning techniques and thereby predicting the star rating given a new review text.

Our Approach:

a) Extracted the restaurant's data (reviews and corresponding star rating).
b) Tokenized the review text into words.
c) Removed stopwords.
d) Hashed the sentence into feature vectors.
e) Used Inverse Document Frequency (IDF) to rescale the feature vectors.
f) Split the data into training and test set. (90:10)

Algorithm Used:

Logistic Regression classification model was used to train the data. 5-fold cross validation technique was used to avoid overfitting of data. In order to tune the parameters, we used ParamGridBuilder functionality provided by spark Mllib, in order to construct a grid of parameters to search over.

Result & Evaluation Metrics:

Accuracy was use as the evaluation metrics.

Accuracy	$ACC = \frac{TP}{TP+FP} = \frac{1}{N}\sum_{i=0}^{N-1} \hat{\delta}\,(\hat{\mathbf{y}}_i - \mathbf{y}_i)$

Accuracy obtained: 78.9 %

Conclusion:

Accuracy obtaines was pretty good. The future scope of this work can be to use wordToVec and Long short-term memory (LSTM) techniques to further improve upon the accuracy. Word to vector techniques creates representation for words that capture their meaning, semantic relationships and different type of context they are used in. For example our current model would not be able to distinguish between “good” and “not good” terms in review texts.

References

[1] https://www.yelp.com/dataset/

[2] https://spark.apache.org/docs/latest/ml-guide.html