



AMAZON REVIEW GENDER PREDICTION

CAPSTONE PROJECT REPORT

EXECUTIVE SUMMARY

The aim of this project is to apply natural language processing techniques and various ML models to predict the gender of Amazon reviewers. Using a convolutional neural network with TensorFlow backend, our team was able to successfully classify the reviewer gender with 72.11 % accuracy. Our results and demo are available below:

GitHub Repo:

<https://github.com/xyhacker/>

Web Demo:

<http://genderpredict.co/>

TABLE OF CONTENT

Executive Summary	2
Project Introduction	4
Main Challenges	5
Labeling the Data	7
Text Processing using the NLTK Library	8
Machine Learning Models	9
Results	11
Concluding Remarks	12
Our Team	13
Endnotes	14

PROJECT INTRODUCTION

Amazon prides itself in being “Earth’s most customer-centric company”. For the 8th year in a row, customers ranked Amazon #1 in The American Customer Satisfaction Index¹ and the company ranks similarly in customer satisfaction surveys around the world.


Product and seller reviews are core to improving customer experience, and thus these have been generated and collected for decades. Cleaned and preprocessed review data in the JSON format are publicly available². The dataset features reviewer and product IDs, usernames, helpfulness scores, review text and summary, overall score for the product, and time of posting. We use a subset of this dataset in our analysis.

The aim of this project is to apply natural language processing techniques and various ML models to predict the gender of Amazon reviewers. Originally, the grocery and gourmet food reviews were sampled, however we decided to expand the dataset to also include movie and TV reviews to increase the number of overall usable samples.

MAIN CHALLENGES

1. The data does not explicitly state whether or not someone is male or female so the team must solve this piece of data engineering first.
2. There's about 1.3 million data entries so our team is working with large dataset requiring significant processing power.
3. Next, we needed to develop a flask application to encapsulate our model to be ready to use in the form of a web app.
4. Finally, we faced the challenge of integrating the frontend and backend.

JSON FILE SAMPLE

A window titled "JSON Record Sample" with a dark gray background and rounded corners. It contains a JSON object with the following fields: reviewerID, asin, reviewerName, helpful, reviewText, overall, summary, unixReviewTime, and reviewTime. The reviewText field contains a multi-line string describing a piano and hymns.

```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays
the piano. He is having a wonderful time playing these
old hymns. The music is at times hard to read because we
think the book was published for singing from more than
playing from. Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


LABELING THE DATA

Although the data came labeled with user-selected nicknames, most of these could not be easily classified as either female or male without further data preparation.

Our team chose the Gender Guesser Python library that contains a list of more than 40,000 first names and gender, plus some 600 pairs of “equivalent” names. This list covers the vast majority of first names in all European countries and in some overseas countries (e.g. China, India, Japan, U.S.A.) as well. GenderGuesser labels data as “male/female”, “mostly male/female” and “unknown”. To ensure high quality of the data used in our prediction model, we only used unambiguous data labels.

The initial data cleanup and GenderGuesser library pass identified only approximately 10% samples from our original dataset. We soon realized we need to adjust the case of the name (first letter must be upper-case), and names such as “J. McDonald” from the sample displayed above would not be recognized by GenderGuesser at all.

Additional data processing & adjustment got us to the total of 50% data labeled, however even with all these steps taken, most of the dataset still remained unclassified. Hence, our team made the decision to add reviews from additional category (“Movie & TV Reviews”).

TEXT PREPROCESSING USING THE NLTK

The NLTK (The Natural Language Toolkit) is one of the most well-known language processing libraries in the industry. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

Our preprocessing steps included:

- Eliminating stopwords
- Expanding contractions
- Stemming the text
- Formatting and removing unwanted characters with RE
- Encoding genders
- Save the adjustments in the clean_data dataset

MACHINE LEARNING MODELS

The image on the right represents the best performing model used. Our Keras CNN model has 7 convolution layers, 4 maxpooling layers and 2 dense layers.

The first layer is an Embedding layer. This layer takes the integer-encoded vocabulary and looks up the embedding vector for each word-index. These vectors are learned as the model trains. The vectors add a dimension to the output array. The resulting dimensions are: (batch, sequence, embedding). Between them, we are using dropout to prevent overfitting. In the output-layer, we use the sigmoid function, which maps the values between 0 and 1.

After running for 2 epochs, with a batch size of 256 samples, this model achieves an accuracy of 71.89%.

We have also tested the Multinomial Naive Bayes, Gradient Descent Classifier and other models with limited results. generally ranging within the 50 - 70% accuracy scores.



Model Definition

```
model = Sequential()

model.add(Embedding(input_dim=top_words,
                    output_dim=embedding_size,
                    input_length=max_tokens))

model.add(Convolution1D(256, 3, padding='same'))
model.add(Convolution1D(128, 3, padding='same'))
model.add(MaxPooling1D(pool_size=2))
model.add(Convolution1D(64, 3, padding='same'))
model.add(MaxPooling1D(pool_size=2))
model.add(Convolution1D(32, 3, padding='same'))
model.add(MaxPooling1D(pool_size=2))
model.add(Convolution1D(16, 3, padding='same'))
model.add(MaxPooling1D(pool_size=2))
model.add(Convolution1D(8, 3, padding='same'))
model.add(Convolution1D(8, 3, padding='same'))
model.add(Flatten())
model.add(Dropout(0.1))
model.add(Dense(180,activation='relu'))
#model.add(Dropout(0.2))
model.add(Dense(2,activation='sigmoid'))
#model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',
              metrics=['accuracy'])
model.fit(X_trainRos, Y_trainRosHot, epochs=2, batch_size =
256)

# Evaluation on the test set

scores = model.evaluate(X_testRos, Y_testRosHot, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

RESULTS

Having labeled, cleaned, and preprocessed the data, we were able to successfully input it to our TensorFlow Convolutional Neural Network model.

Although we tested numerous models and dataset variants, the CNN-based model performed the best, resulting in the 72.11% accuracy score. Since the number of samples was imbalanced, our team is using SMOTE oversampled data with Keras model for deploying to production.

CONCLUDING REMARKS

This Capstone Project represents the culmination of the 6-month Data Science & Machine Learning program at Lambda School.

Throughout this project, our team successfully applied the concepts learned throughout the pilot iteration of the program and demonstrated in-depth knowledge of various machine learning concepts.

We would like to extend our heartfelt gratitude to:

Our Instructors:

Aaron Gallant
Thomson Comer
Ryan Herr

Our Project Managers:

Susanna McDonald
Behnam Arzaghi
Ryan Allred

The leadership and staff at Lambda School.

THE CAPSTONE PROJECT TEAM

TEAM LEAD: Susanna McDonald
susanna.mcdo@gmail.com

TEAM MEMBERS: Brenner Haverlock
brenner.haverlock@gmail.com

Kelvin Li
kelvinli.math@gmail.com

Lorin Fields
lorin.fields@gmail.com

Saranya Mandava
mandava807@gmail.com

Tina Kovacova
awenare@gmail.com

ENDNOTES

- 1 Amazon (2018). *2017 Annual Report*. Retrieved 20 September, 2018 from: <http://phx.corporate-ir.net/External.File?item=UGFyZW50SUQ9NjkyMDIxfENoaWxkSUQ9NDAYOTkyfFR5cGU9MQ==&t=1>
- 2 McAuley, J., (2018). *Amazon Product Data*. Retrieved 20 September, 2018 from: <http://jmcauley.ucsd.edu/data/amazon/links.html>
- 3 Cover photo by Magda Ehlers from Pexels. Retrieved from: <https://www.pexels.com/photo/boy-and-girl-cutout-decals-1386336/>