CASE STUDY

NATURAL LANGUAGE PROCESSING IN PYTHON

Presented by Stella Liu

March 15th 2020

OUTLINE

TODAY'S TOPICS

- Introduction
- Process
- Data
- Model Training
- Model Testing
- Learning

INTRODUCTION

CASE STUDY: PREDICTING WINE REVIEWER NAME USING WINE REVIEWS

In today's world, consumers rely heavily on the reviews of a product or service to help them make the right purchasing choice. This goes for the wine industry as well. When a wine reviewer provides a good amount of wine reviews and establishes a fan base, the fans of the reviewer would make the decision of which wine to try basing on the reviewer's comments.

This leads to a questions we would like to explore: "Can we use the information contained within wine reviews to predict which reviewer they have been written by?"

Stella Liu 2020

PROCESS

DATA SCIENCE PROJECT FLOW

DATA

Importing data set used for the case study. After obtaining the data, data evaluation, data manipulation, missing value handling, and features creation was performed. Lastly, splitting the data into training and testing set.

TRAINING

Start to train our model using the training data set. In this step, we also perform model selection, ensemble model, model optimization by tuning the features, and more.

TESTING

After training the model, we are now ready to do the model evaluation using the testing data set. This is a cross validation technique to help us understand how the model would perform with unseen data. Performing metrics like accuracy and confusion matrix.

LEARNING

In this final step, results and learning are shared as well as the key features for the model.

Stella Liu 2020

DATA

SOURCE

For this case study, we will be using the data set with roughly 130,000 wine reviews that were scrapped from the official publication "Wine Enthusiast" during the week of June 15th, 2017. We obtained the data that's stored in the CSV file format from site: https://www.kaggle.com/zynicide/wine-reviews

OVERVIEW

After loading the sample data, we reviewed the first three rows to understand the field names and types that we will be working with (see screenshot below).

Column1	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	title	variety	winery
0.0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87.0	NaN	Sicily & Sardinia	Etna		Kerin O'Keefe	@kerinokeefe	Nicosia 2013 Vulkà Bianco (Etna)	White Blend	Nicosia
1.0	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87.0	15.0	Douro			Roger Voss	@vossroger	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portuguese Red	Quinta dos Avidagos
2.0	US	Tart and snappy, the flavors of lime flesh		87.0	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rainstorm 2013 Pinot Gris (Willamette	Pinot Gris	Rainstorm

DATA

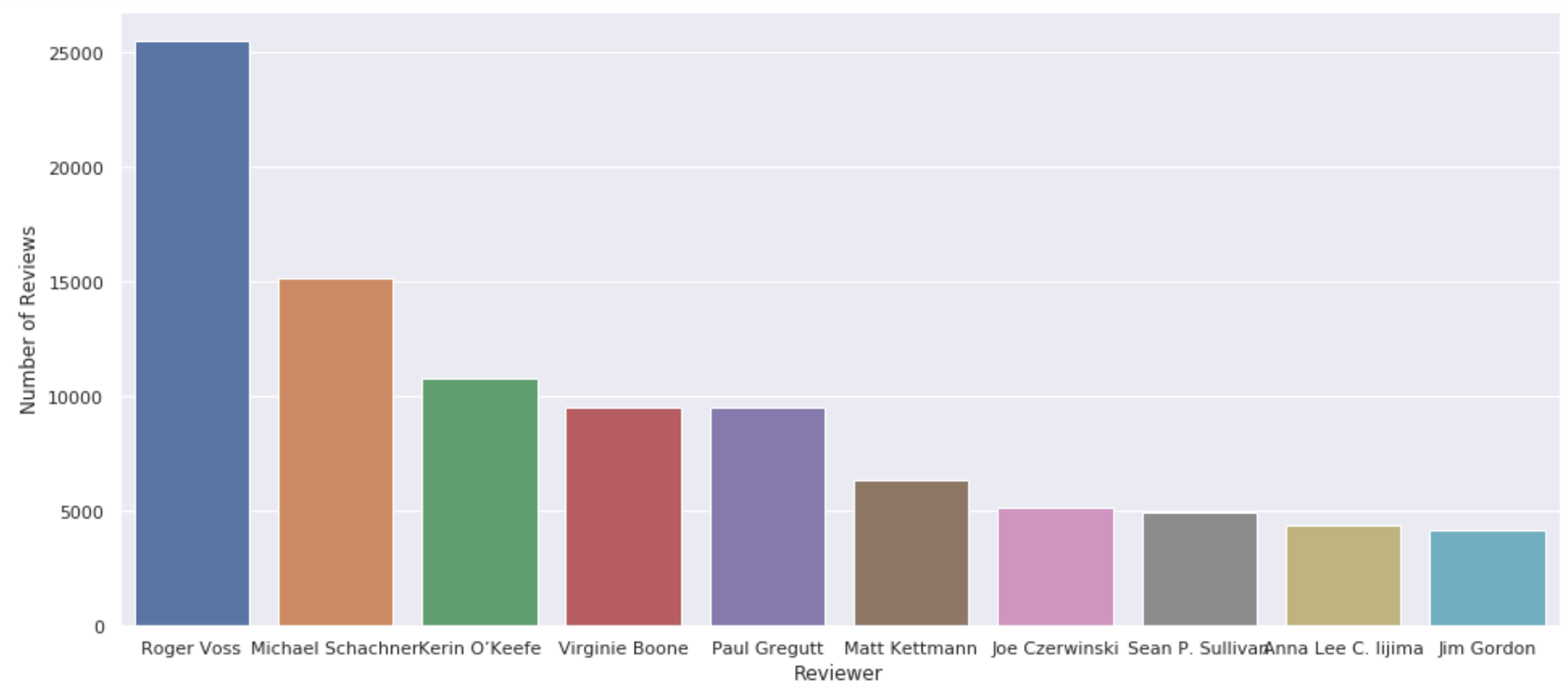
EXPLORATORY DATA ANALYSIS

Before diving into the modeling part of the project, it's important to perform deep dive into the data to understand what are the fields we are working with, whether there's missing, and what information to include as our target feature to make the prediction for.

During the EDA, we found that ~20% of the reviews do not have reviewer's name information. Since we are trying to predict who the reviewer is, we will exclude those observations. We also found out that there are 19 reviewers in our dataset with one of the reviewer, Roger Voss, having the most number of reviews (~25% after removing the missings in the prior step). In order to build the model with enough reviews, we will focus on the top 10 reviewer who gives the most reviews in our data. This approach would give us ~92% of the data to work with (see next page for top 10 reviewer distribution).

DATA

EXPLORATORY DATA ANALYSIS



Stella Liu 2020

DATA

TEXT CLEANING AND FEATURE CREATION

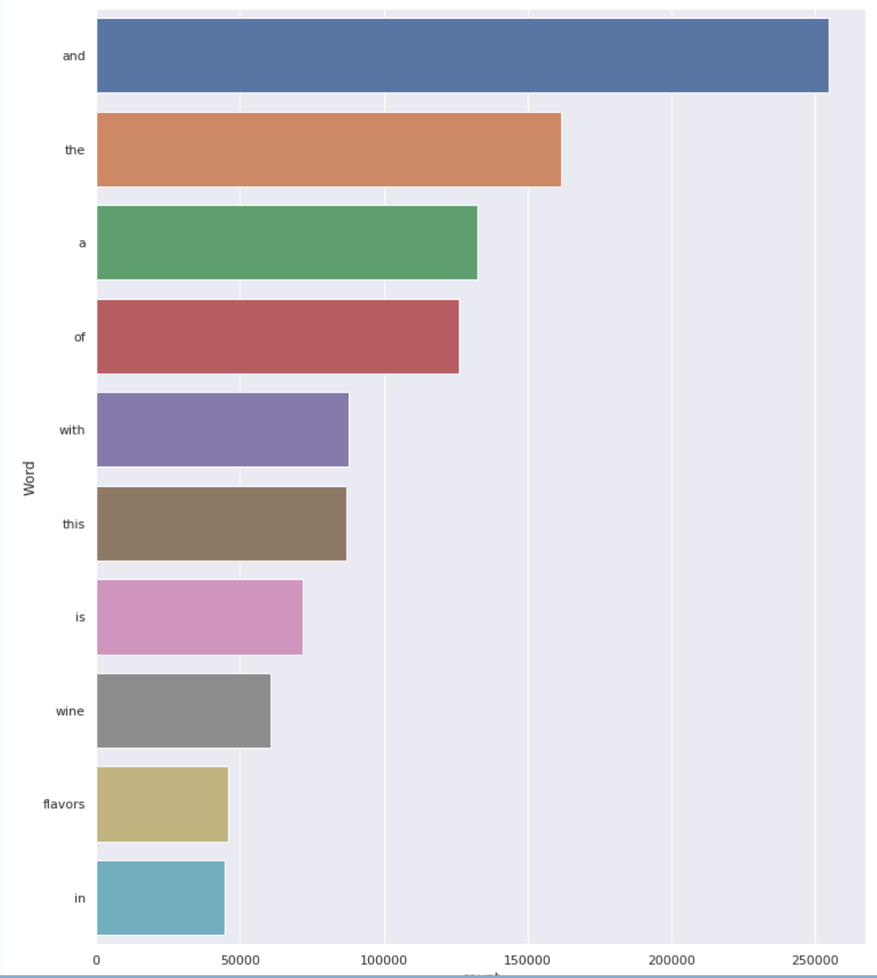
We will be using the wine description as the feature to predict who the reviewer is. In order to mine the key information from the unstructured reviews, we first need to perform cleaning of the text block. For this case study, we keep only alphabets, make all words lower cases, and remove the common stop words like "the", "on", "are", etc.. By doing so, we can now focus on the important words in the review.

Next, we use the cleaned text to create the features for our modeling. In this study, we used two different vectoring methods: TF-IDF term weighting and unigram Common Vectorizer. With the TF-IDF, commonly used words would be given lower weight and unique words would get higher score/weight. For the Common Vecorizer, we first select words that occur in at least 1,000 reviews, then created the dichotomous indicator that if a given text contain a word, the feature/variable would be set to 1.

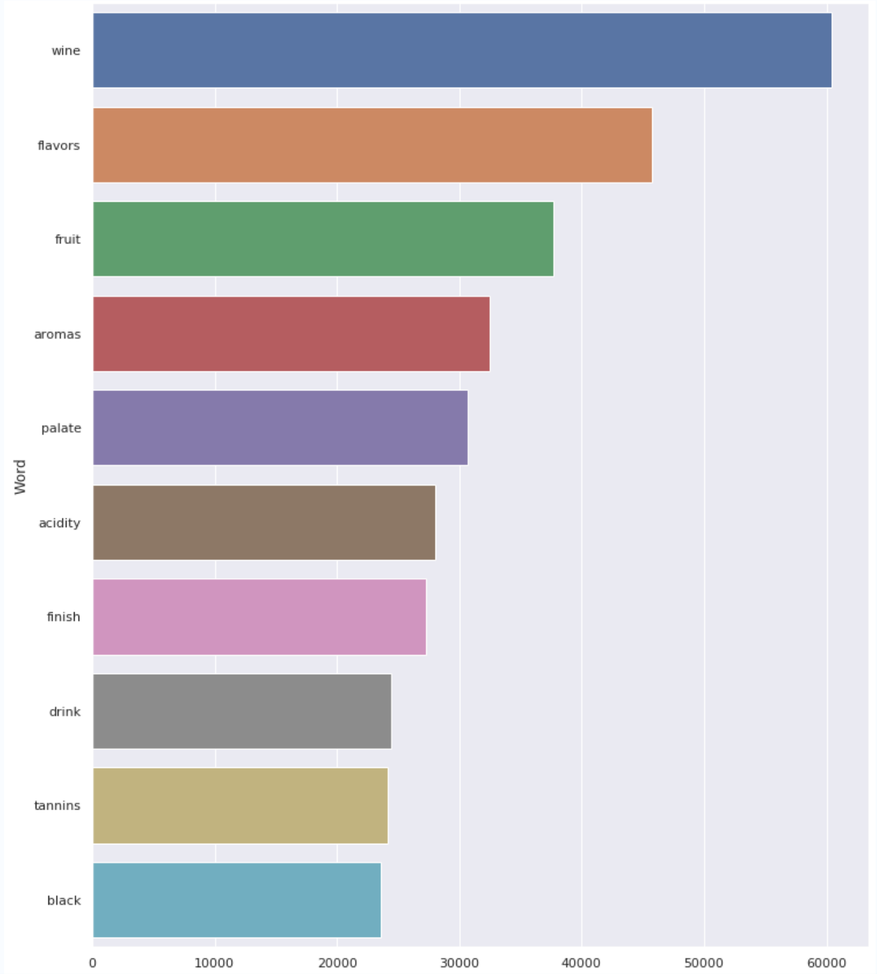
Last, we create training and testing data set at a 70-30 ratio split where we would use the training data to train the model and testing data to perform evaluation and validation.

DATA

TOP WORDS BEFORE CLEANING



TOP WORDS AFTER CLEANING



MODEL TRAINING

MODEL 1: MULTINOMIAL NAIVE BAYSE CLASSIFIER

Naive Bayes Classifier is a family of probabilistic algorithms base on applying the Bayes' theorem with the naive assumption. Naive Bayes are mostly used in NLP problems where one can predict the tag of a text. The output of the tag/target is assigned to the one with the highest calculated probability of a given text. Here, we are using the Multinomial Naive Bayes, with multinomial distribution, since this works well for the data that can be turned into counts. This is suitable for classification with discrete features (e.g. word counts for text classification).

Stella Liu 2020

MODEL TRAINING

MODEL 2: RANDOM FOREST CLASSIFIER

For our second model training, we used the Random Forest ensemble tree-base learning algorithm where it used aggregated votes from different decision trees to decide the final class/reviewer of a review. We use this approach because this can handle lots of input features and produce highly accurate classification results. With Random Forest, we can also view and understand what are the top features for the model.

When setting up our model, we set the model to create 500 bootstrap sampled trees with max number of features in each tree allowed to be square root of the total available features.

MODEL TESTING

MODEL 1: MULTINOMIAL NAIVE BAYSE CLASSIFIER

After training the model, we use our testing data set (30% of the total available data from top 10 reviewers) to evaluate our model performance.

This model run time was under 1 second with the accuracy of the class prediction at 91.26%.

Two tables at the right shows the distribution of our target class in testing data set (counts of observations) and the confusion matrix between the actual class and predicted class where the diagonal cells show when the model predicted the correct class/reviewer.

```
=== class distribution ===
0    7757
1    4504
2    3217
4    2871
3    2855
5    1857
7    1540
6    1526
8    1293
9    1239
```

```
=== Confusion Matrix ===
[[7740   17    0    0    0    0    0    0    0    0]
 [   3 4489    0    0   12    0    0    0    0    0]
 [   4    5 3206    1    1    0    0    0    0    0]
 [  88  123   11 2585   38    5    1    0    3    1]
 [  89  189    5   17 2568    2    0    0    1    0]
 [   8   30    6   23    8 1782    0    0    0    0]
 [ 168  309   12   65  129    5  834    0    3    1]
 [  53  129   37   24   85    1    1 1210    0    0]
 [  40   31   31   46   31    4    0    0 1110    0]
 [ 270  190    9   98   40    2    0    1    0  629]]
```

Stella Liu 2020

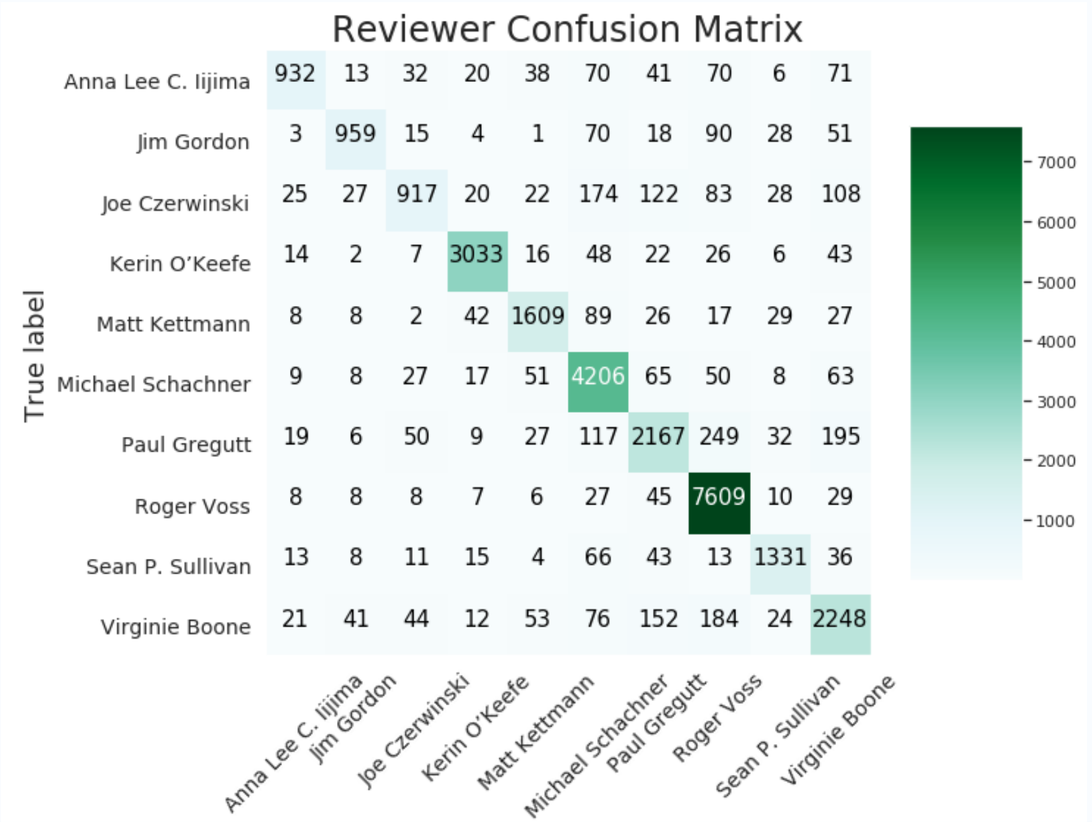
MODEL TESTING

MODEL 2: RANDOM FOREST CLASSIFIER

Same approach for the Random Forest model. After training the model, we use our testing data set (30% of the total available data from top 10 reviewers) to evaluate our model performance.

This model run time was a little over 3 minutes with the accuracy of the class prediction at 87.27%.

The table at the right shows the confusion matrix of the test data output between the actual class and predicted class where the diagonal cells are when the model predicted the correct class/reviewer.



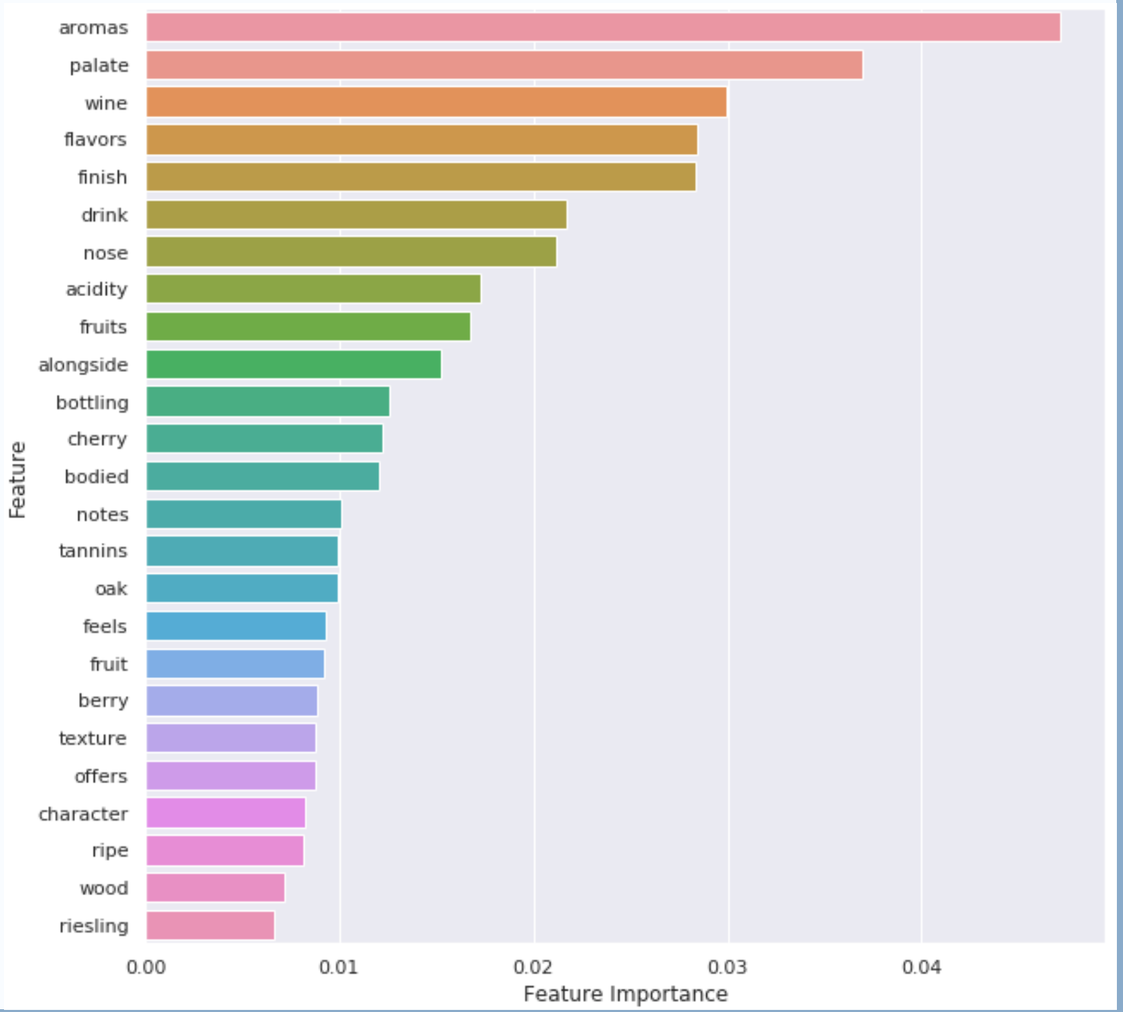
Stella Liu 2020

MODEL TESTING

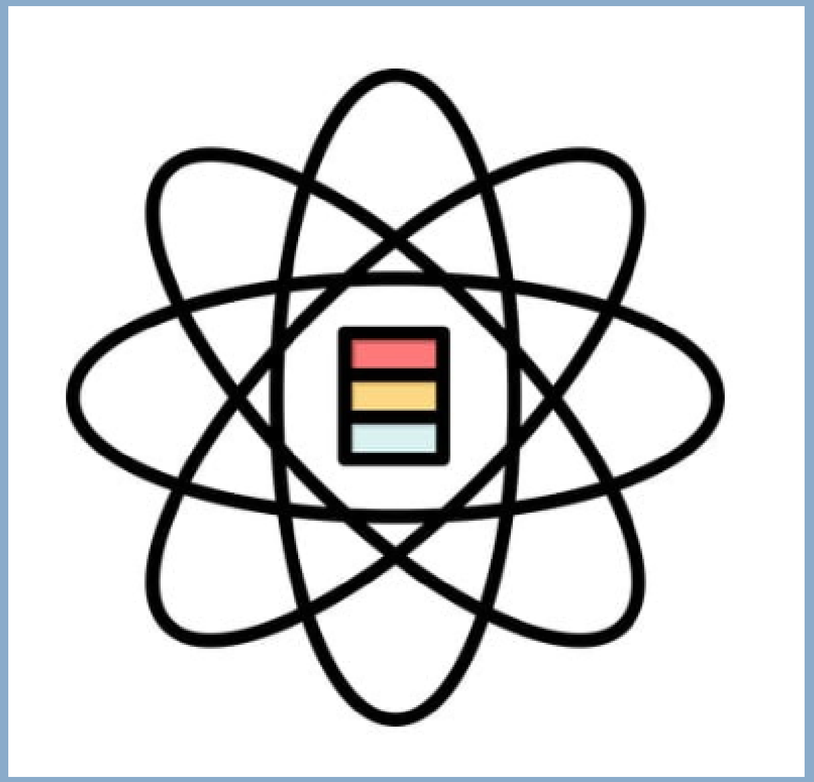
MODEL 2: RANDOM FOREST CLASSIFIER

With the Random Forest, we also utilize the feature selection output to show us what are some of the key features used in the model to help us classify who the reviewer is for a given review.

On the right is the plot of the top 25 features for our model. This is also a good way to verify whether the key words/ features make sense. Notice that some of the key words are: aromas, palate, finish, acidity, bottling, etc.. This list looks like a good starting point.



Stella Liu 2020



LEARNING

In this case study, we used two different ways of creating our model features (TF-IDF weights and Common Vector words) and train our data using two different modeling methods (Multinomial Naive Bayse Classifier and Random Forest Classifier) as well. For the different approaches, each has their unique strength. For both, we utilized our unseen testing data set to evaluate our models.

As a Data Scientists, I stay curious and utilize different learning each time in order to identify what the appropriate approach should be when solving a problem. Here, both models had good accuracy results given that we have 10 levels of class for the target/reviewer.

FOOD FOR THOUGHT

Information is the oil of the 21st century, and analytics is the combustion engine.

PETER SONDERGAARD

THANK YOU!

LINKEDIN

linkedin.com/in/yu-jen-stella-liu/

GITHUB

github.com/eewwazcp/stella-nlp-learning

EMAIL

stellaliu0607@gmail.com