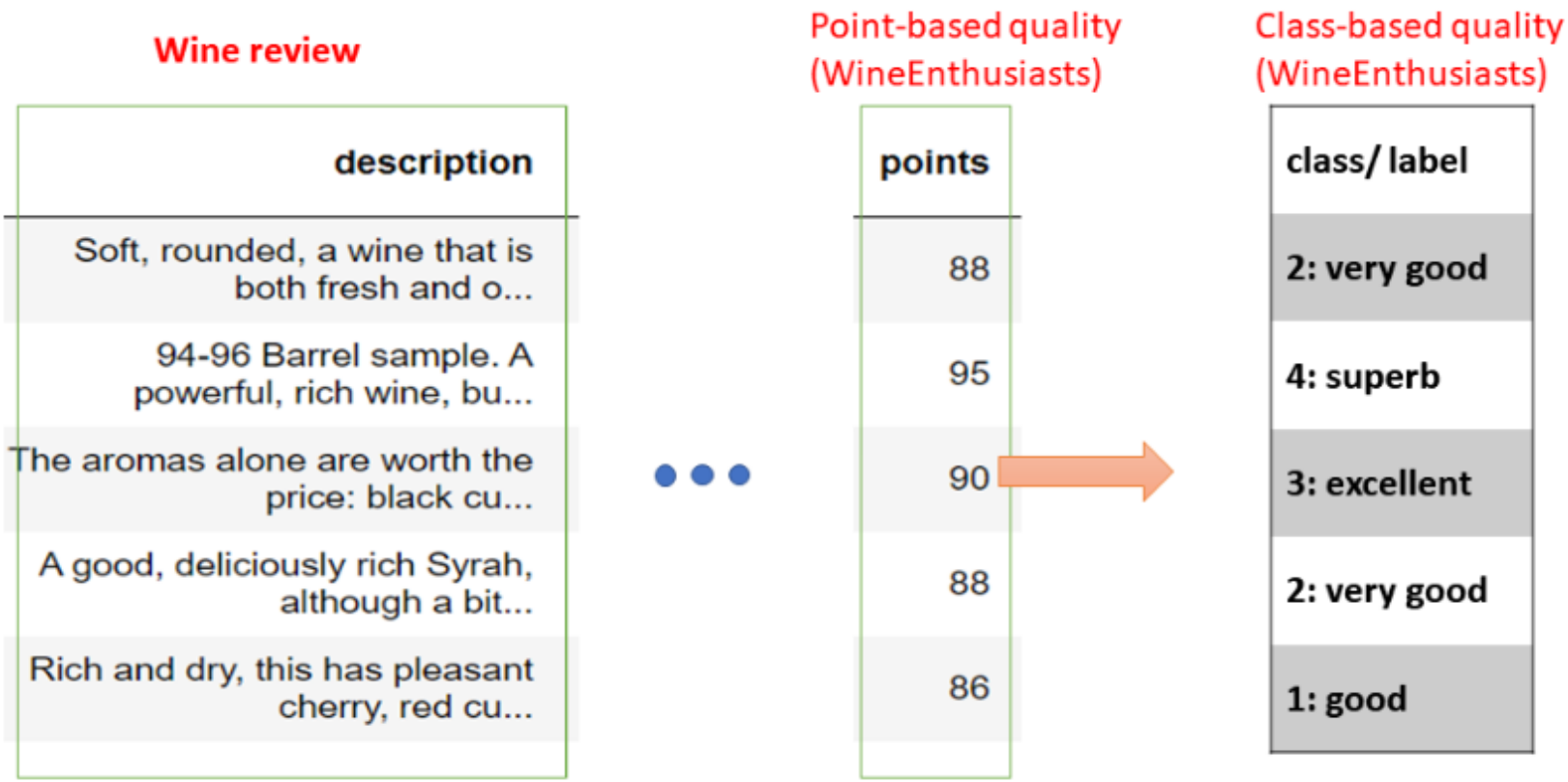
Final project - LING 539

Hanh Nguyen

University of Arizona

Project's objective

Build a classifier to predict the quality of wine based on the wine review in the dataset [1].



[1] https://www.kaggle.com/zynicide/wine-reviews/data#winemag-data_first150k.csv

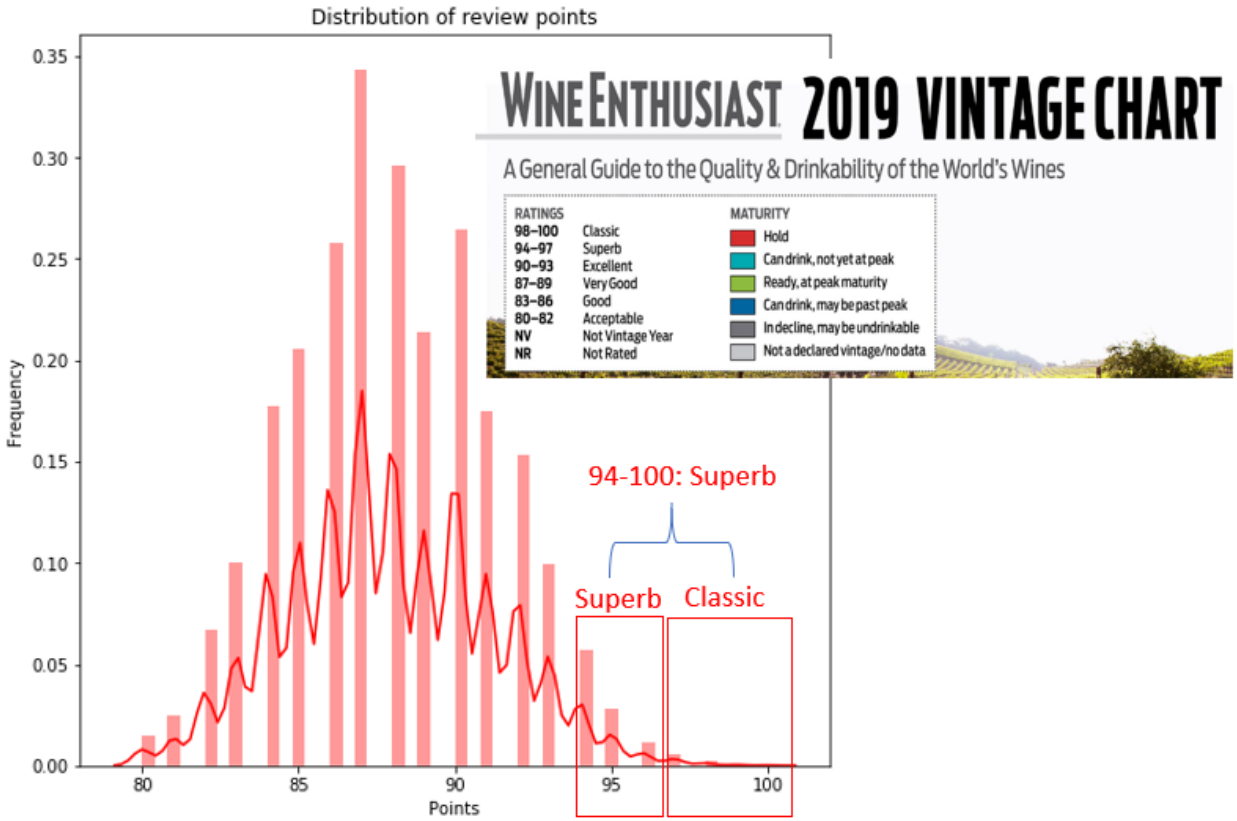
Project's objective

From points to class labels:

class 0: Acceptable (80-82 pts)
class 1: Good (83-86 pts)
class 2: Very Good (87-89 pts)
class 3: Excellent (90-93 pts)
class 4: Superb (94-100 pts)

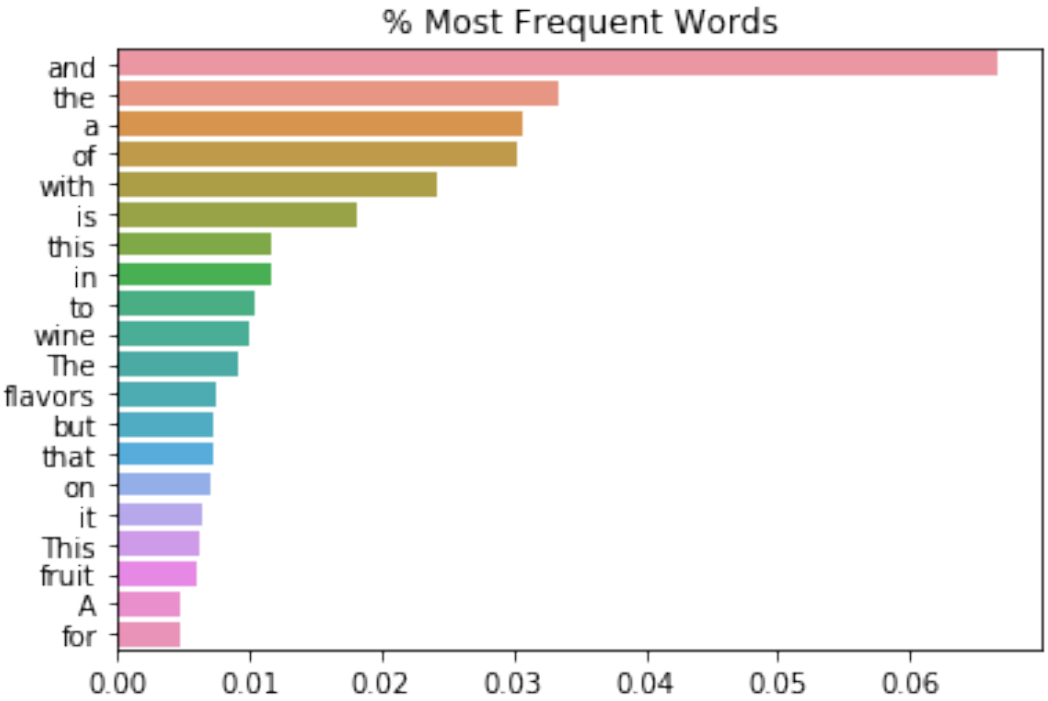
Based on The Official 2019 Wine Vintage Chart (see `https://www.winemag.com/2019/01/02/wine-vintage-chart-2019/`) and the distribution of points in the dataset.

Preprocessing text description



- Wine categories are classified based on scales and distribution of rating points
- Using the cardinalities of classes to determine corresponding weights of classes

Preprocessing text description



Top 20 words having highest occurence in the wine reviews. A lot of stopwords like "this", "is" etc. not useful information

→ Use word_tokenize in NLTK to remove stopwords and non-alpha characters, lower cases.

Training the classifier

- Dataset (150,930 reviews): train set (95%) and development set (5%)
- Tokenize reviews and build a vocabulary of words (CountVectorizer) from tokens in the train set (see **TextToFeatures** function). Tuning parameters: n-grams (uni vs bi-gram)
- Initialize the classifier: Logistic Regression, multi-class, 5-fold CV, solver = "sag" (for large dataset),
  class weight: $\{0:7, 1:1, 2:1, 3:1, 4:7\}$ (weights $\approx$ inverses of cardinalities)
- Train the classifier on the train data.
- Make prediction on the development set.

Evaluating the model

Unigram vs Bigram

	Uni-gram	Bi-gram
Confusion matrix	[[253 70 6 0 0] [124 1717 416 64 5] [14 361 1731 359 39] [5 46 363 1481 141] [0 0 8 85 259]]	[[268 59 2 0 0] [65 2004 228 27 2] [3 207 2082 208 4] [1 25 222 1750 38] [0 0 1 93 258]]
Misclassification errors	precision recall f1-score support 0 0.639 0.769 0.698 329 1 0.783 0.738 0.760 2326 2 0.686 0.691 0.689 2504 3 0.745 0.727 0.736 2036 4 0.583 0.736 0.651 352	precision recall f1-score support 0 0.795 0.815 0.805 329 1 0.873 0.862 0.867 2326 2 0.821 0.831 0.826 2504 3 0.842 0.860 0.851 2036 4 0.854 0.733 0.789 352
Accuracy	**72.1%**	**84.3%**
Processing time	1 hour	3 hours

Limitations

- This dataset includes only positive wine reviews (at least 80 points). This classifier is NOT applicable to all types of wines.
- The Bag of Word approach may not be sophisticated enough to capture subtle differences between neighboring classes.
- This classifier may take hours to train and predict (dataset size is large).
- This classifier does not take into account other features besides the reviews.

Future directions

.

- May use tf-idf to model similarly like CountVectorizer.
- Combine text description with other columns of the dataframe (like "Prices", "Winery" etc) to predict better.
- The reviews are mostly positive in the dataset. Word relationships are therefore important. Word embedding and deep learning methods (RNNs) are probably more suitable (and are probably slower to run).
- May build a model to identify the variety, winery, location of a wine, using text description in the dataset.

THANK YOU!