What’s Cooking?

The Naive Baes: Nora Myer, Emily Engle, Frank Meszaros, Serena Davis

Recipe Classification

- Classifying recipes by cuisine
- Based on ingredients in recipe
- 39,774 classified recipes
  - Broken up into training and test
- Thought it would be fun and interesting text classification problem


```
{
    "id": 10259,
    "cuisine": "greek",
    "ingredients": [
    "romaine lettuce",
    "black olives",
    "grape tomatoes",
    "garlic",
    "pepper",
    "purple onion",
    "seasoning",
    "garbanzo beans",
    "feta cheese crumbles"
    ]
}
```

More about the data

- Number of cuisines: 20
- Number of ingredients: 6714
- Top 10 ingredients per cuisine
- Mean, median, standard deviation of cuisines and ingredients
- More in *data_analysis.txt*
- Provides context for representing data

Cuisine Type	Number of occurrences
Italian	7838
Mexican	6438
Southern US	4320
Indian	3003
Chinese	2673

Ingredient Type	Number of occurrences
Salt	18049
Olive oil	7972
Onions	7972
Water	7457
Garlic	7380

Text Classification

- Classifying documents into categories based on contained text
  - e.g. email spam identification, news article topic classification, movie review sentiment analysis
- Cuisine classification based on ingredients can be thought of as text classification
- Different because there is no need to consider grammar, parts of speech, word order
- Similar because of drastic variation in relative importance of words
  - “Salt” is the “The” of this dataset
- We attempted to address these similarities and differences with our feature representations

Representing the Data

Feature Vector

- Bag of words model
- Vector is array of 1s, 0s
- Feature mapped to unique ingredient
- Simple, clear text classification
- Longest step was pre-processing
    - 6,700 ingredients


```
[0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, . . . 0, 1, 0]
```

TF-IDF

- Term frequency - inverse document frequency
- Frequency score weighted by document frequency
- Gives weight to better classifying ingredients such as turmeric
    - Gives salt and water less weight

TF-IDF Score

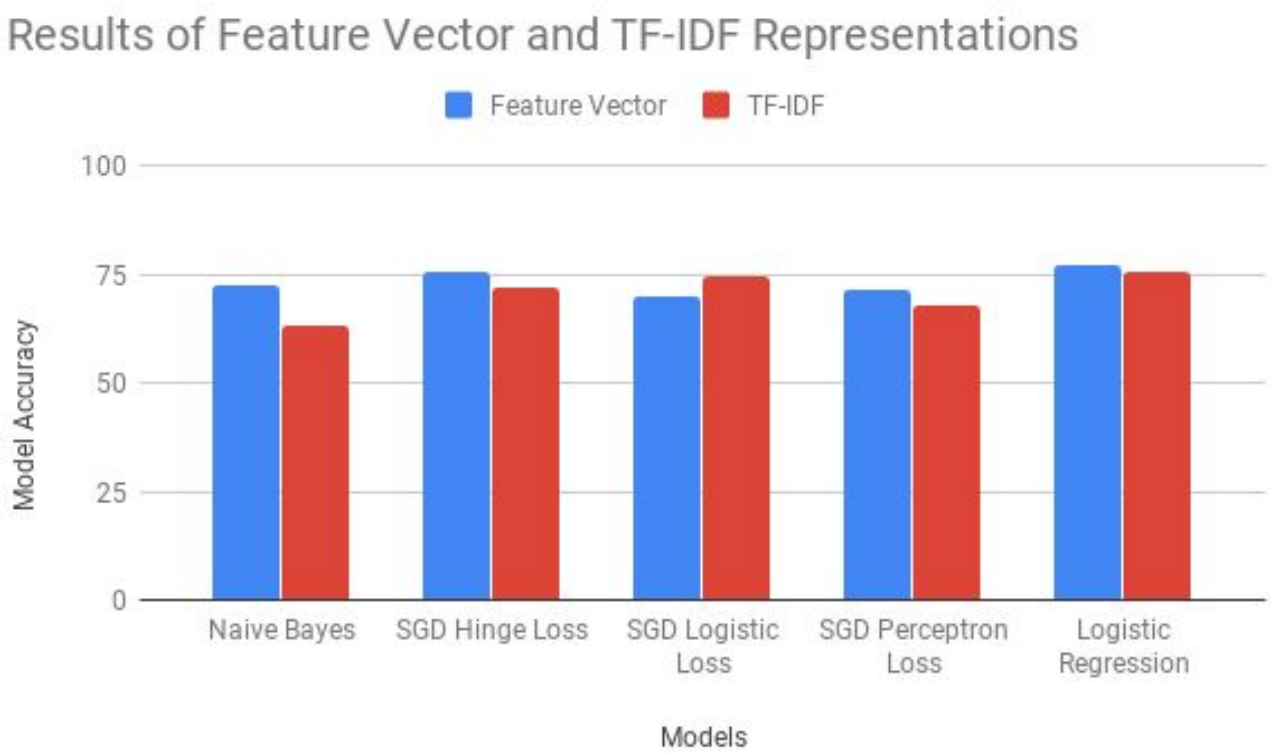
$$TF - IDF\ Score = TF_{x,y} * IDF = TF_{x,y} * \log\frac{N}{df} \ldots\ldots (1)$$

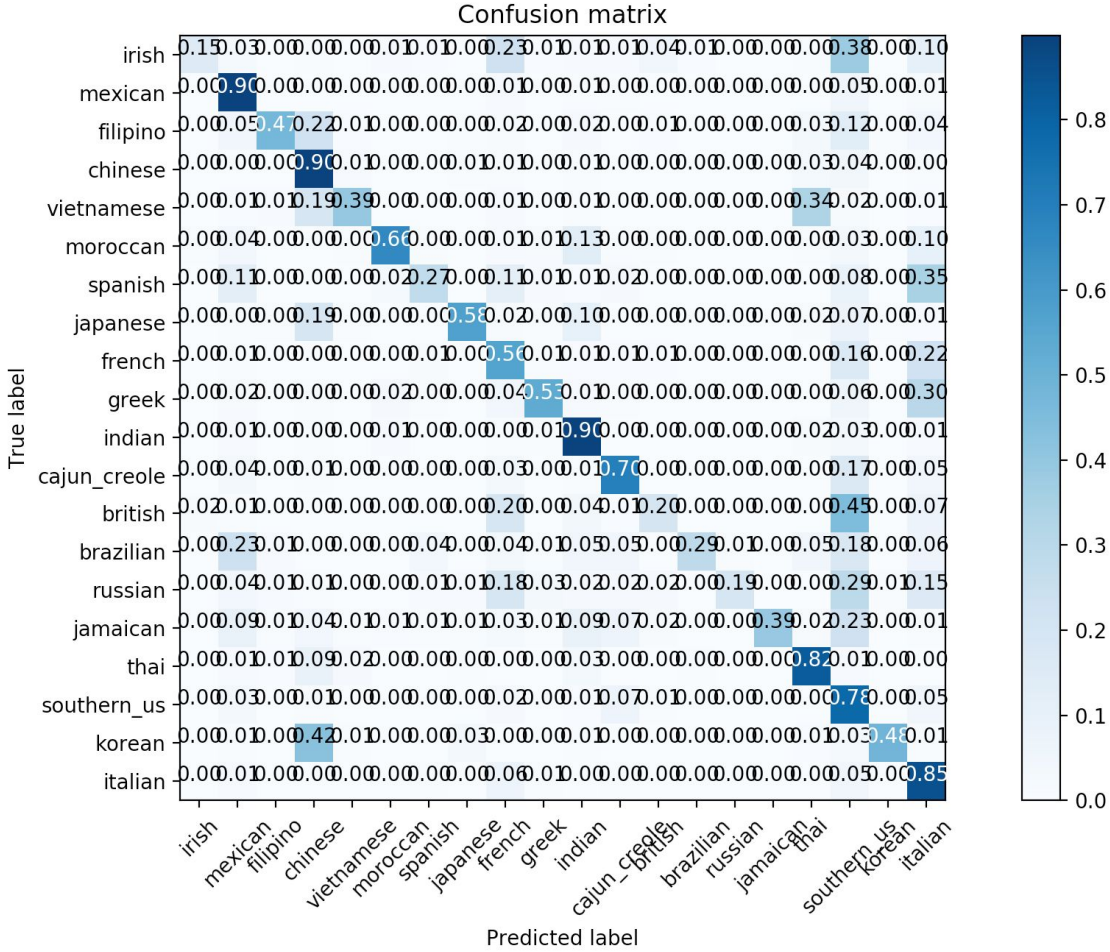
, where $TF_{x,y}$ is the frequency of keyphrase X in the article Y,
N is the total number of documents in the corpus.
df is the number of documents containing keyphrase X

Classification

Classifiers Used

- Naive Bayes
  - Assumes independence
  - Good baseline model
- Linear Classifier (Logistic Regression)
  - Measures the relationship between the categorical dependent variable and one or more independent variables
- SGD
  - Hinge
  - Log
  - Perceptron


Results of Feature Vector and TF-IDF Representations
Feature Vector
TF-IDF
100
75
50
25
0
Model Accuracy
Naive Bayes
SGD Hinge Loss
SGD Logistic Loss
SGD Perceptron Loss
Logistic Regression
Models


Confusion matrix
True label
Predicted label
irish
mexican
filipino
chinese
vietnamese
moroccan
spanish
japanese
french
greek
indian
cajun_creole
british
brazilian
russian
jamaican
thai
southern_us
korean
italian
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

Results Analysis

- Baseline -- Naive Bayes with feature vectors:
  - 72.37% Accuracy
- Best -- Logistic Regression with feature vectors:
  - 76.95% Accuracy

- Performance improvement with logistic regression was expected because of correlated features
  - Recipe ingredients are highly dependent on one another
- Surprised to find that TF-IDF did not offer performance improvement for most models.

Next Steps

- Text cleaning
  - Reduce noise in the form of punctuation, suffix variations, etc.
  - Ex. “sliced tomatoes”, “tomato slices”, and “chopped tomatoes”
- Ensemble models
- Use word embeddings to capture semantic similarities among ingredients
- Augment dataset with more/better examples

Thanks! Questions?