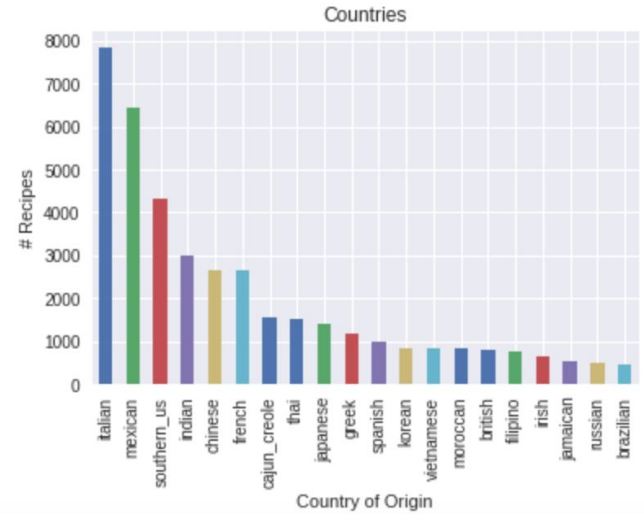
Recipe Identification: Multiclass Classification

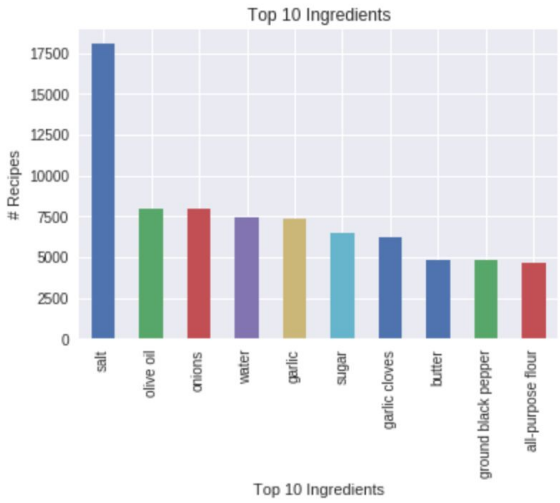
Project Milestone: Stanford University CS229

Course	CS229: Machine Learning (Fall 2018)
Project	Where is the Chef From?
Category	General Machine Learning, Multiclass classification, Food and Drinks
Team	Manish Pandit (manish7) - Code, Algorithms Annie Pitkin (apitkin) - Code, Research, Write Up Hermann Qiu (hq2128) - Code, Algorithms, Write Up

Motivation:

Goal: Categorize a recipe's country of origin based on the recipe's ingredients.
Data overview: 39,774 recipes with 6714 ingredients from 20 countries





Method:

Our dataset contains large set of unique recipes belonging to various cuisines and associated list of ingredients. The format of the dataset is json. To be able to model the text data effectively we extracted the ingredients as individual elements and mapped the cuisines and ingredients to dictionaries. Then we created an *(m, n) design* matrix from our training set, where *m* is the number of recipes and *n* is the number of ingredients. Our design matrix employs

sparse representation of each ingredients in the recipes, so the matrix only consists of zeros and ones. Also, we created a corresponding *(m, 1)* matrix with values ranging from 0-19 to represent the country of origin for each recipe in the dataset. We split the data into 37,785 examples for training and 1,989 examples for test - this is roughly a 95% / 5% split.

Preliminary Experiments:

To train this model we implemented a neural network using the Tensorflow and Kera packages in Colab using Google's TPU. We wanted to see a baseline model so we implemented a simple one layer neural network to get an output and understand how our data behaves.
The the input layer flattens the data. We have one hidden layers with 1024 neurons and ReLU as the activation function. We use a drop out rate of 20% to regularize the network and prevent overfitting. Finally, the output layer uses a softmax activation function and has the same number of neurons as number of cuisines (20). The optimizer is adam with default mini batch gradient descent. Each neuron in the output layer predicts the probability of that cuisine given a recipe. We initially ran 5 epochs for training. The neuron with the highest probability is our selection for the given recipe. The loss function is sparse cross entropy.

The initial results are very encouraging but perhaps not optimal. The accuracy of our model on training set is around 97% and accuracy on test set is around 78%. The model clearly overfits the training data.

Our initial error analysis revealed that there is an opportunity to cleanup the data; in particular remove words from names of the ingredients that don't any value to learning and makes ingredients appear distinct while it is not.

Next Steps:

Considering that our neural network based model has low bias and high variance, our focus going forward will be on reducing the variance. These are a few techniques we will employ to try to reduce the variance:

- **Data cleanup and modeling:**
    - Our initial analysis of the data reveals that there are number of options to reduce complexity in the data through clean up. There are indicative words associated with some of the ingredients that add no value during learning but add to the complexity. For example, currently basic ingredient like tomato appears as different ingredients because it is referred to as "chopped tomato", "diced tomato" or in plural as "tomatoes". We will look to homogenise the ingredients. This will reduce the dimension of the feature vectors and also aid learning.
    - We may look to create a dev set in addition to the training and test sets to facilitate hyper parameter tuning and evaluation.

- **Error Analysis:** We will look to identify and list the reasons why certain samples were mis-classified. We are hoping to learn how to define our design matrix better from this analysis.

- **Tune current neural network:** We will look to address our high variance problem through further tuning and applying various optimization techniques. Including:
    - Early stopping to reduce overfitting.
    - Adding L1 or L2 regularization to the loss.

- **Additional Multiclass classification algorithms:** In addition to current base model, we plan to evaluate additional machine learning algorithms for prediction and compare the performance of each. Additional algorithm we plan to try (but are not limited to):
    - Extended Support Vector Machines.
    - K-Nearest Neighbors.
    - Decision trees.
    - Naive Bayes classifier.
    - Random Forests.

Reference Articles:

1. *Multiclass classification. https://en.wikipedia.org/wiki/Multiclass_classification*
2. *Softmax function: https://en.wikipedia.org/wiki/Softmax_function*
3. *Naive Bayes Classifier: https://en.wikipedia.org/wiki/Naive_Bayes_classifier*
4. *Mohamed, Aly (2005). "Survey on multiclass classification methods" (PDF). Technical Report, Caltech.*