Machine Learning Engineer Nanodegree
Capstone Project Report

Goutham Nekkalapu
goutham.nekkalapu@gmail.com

Categorizing Cuisine Using Recipe Ingredients

I. Definition:

Background:

Data classification and useful data analysis has been a challenging task ever the since data started accumulating in such vast range. Useful analysis and predictions of this data help us to make a lot of productive decisions which helps the companies prosper and expand their market share.

Recipe search and recommendation websites such as Yummly are growing in popularity. Yummly is available both as website and as a mobile app that provides recipe recommendations personalized to the individual's tastes, semantic recipe search, a digital recipe box, shopping list and one-hour grocery delivery. It also lets its users to search items like ingredient, diet, allergy, taste, price etc. When users contribute new recipes to these websites, they are faced with a large number of fields that require manual input of ingredient lists, cooking steps, cuisine types, and descriptions, among other data. This might discourage the users to enter their data. They wanted to solve this problem by predicting some of the fields based on inputs that user has already entered, thus making it easier for the user to share information. Also, these predictions can be used further to refine their services and products.

Yummly has partnered with Kaggle to host a competition to predict the cuisine of a recipe given a list of ingredients. Such classification can help food databases and recipe recommender systems autonomously categorize new recipes based on their ingredients. Yummly shared their labelled dataset which has about 40,000 recipes. Each recipe comprises a list of ingredients, a unique identifier, and a cuisine label.

Problem Statement:

The problem at hand is : "Can we design a model to predict to which cuisine a dish belongs to, given the list of ingredients(recipe) used ?"

I tried to solve this problem in this project. First, I tried to understand the data, did some pre-processing, built a few predictive models, analysed and compared their performances to choose the best one. I have executed this project in python using scikit-learn machine learning library, Numpy, Pandas. In the subsequent sections, I will explain you about various stages of this project.

Metrics:

Here we are solving a classification problem and a recipe can be classified into many cuisines (equal to the number of cuisines available in data set which is 20). While choosing a classifier, not only its accuracy is considered but also its performance against false negatives, false positives. When there is an uneven distribution of different labels(classes) in the data set, the classifier job becomes challenging and it tends to misclassify certain inputs. The latter two mentioned before (false positives and false negatives) gives us a measure of misclassification of a recipe into a different cuisine.

So, precision, recall are used along with accuracy to measure the performance of the model. These take into the account the true positives ,true negatives,false positives and false negatives. Also F1-score which is a weighted average of the precision and recall where it reaches its best value at 1 and worst score at 0 is also used. It expresses a balance between Precision and Recall.

Accuracy = true positives / (total population)
Precision = true positives / (true positives + false positives)
Recall = true positives / (true positives + false negatives)
F1 score = 2 * (Precision * Recall) / (Precision + Recall)

Also confusion matrix was used to interpret and understand the performance of the models. A confusion matrix is a table that is often used to

Describe the performance of classification model on a set of test data for which the true values are known.

II. Analysis

Data Exploration:

The data is available to us in the JSON format. A sample data format can be seen below:

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

To get an insight into the data I have done some analysis on the data like total cuisines available in the dataset, no of recipes each cuisine has, top 10 ingredients used in each cuisine, the most and least common ingredients used in all the cuisines present in the data set.

The below table-1 represents the no of recipes each cuisine has in the data set. We can see that Italian has the highest number followed by mexican and brazilian cuisine has the least. The top 10 ingredients of each cuisine are also examined. For example the top 10 ingredients of ‘indian cuisine’ are :

[(u'salt', 1934), (u'onions', 1195), (u'garam masala', 862), (u'water', 820), (u'ground turmeric', 728), (u'garlic', 726), (u'cumin seed', 697), (u'ground cumin', 683), (u'vegetable oil', 593), (u'oil', 546)]

Also the most common ingredients present across all the cuisines are represented in table-2:

cuisine	no of recipes
Italian	7838
mexican	6438
southern_us	4320
indian	3003
chinese	2673
french	2646
cajun_creole	1546
thai	1539
japanese	1423
greek	1175
spanish	989
korean	830
vietnamese	825
moroccan	821
british	804
filipino	755
irish	667
jamaican	526
russian	489
brazilian	467

Table-1

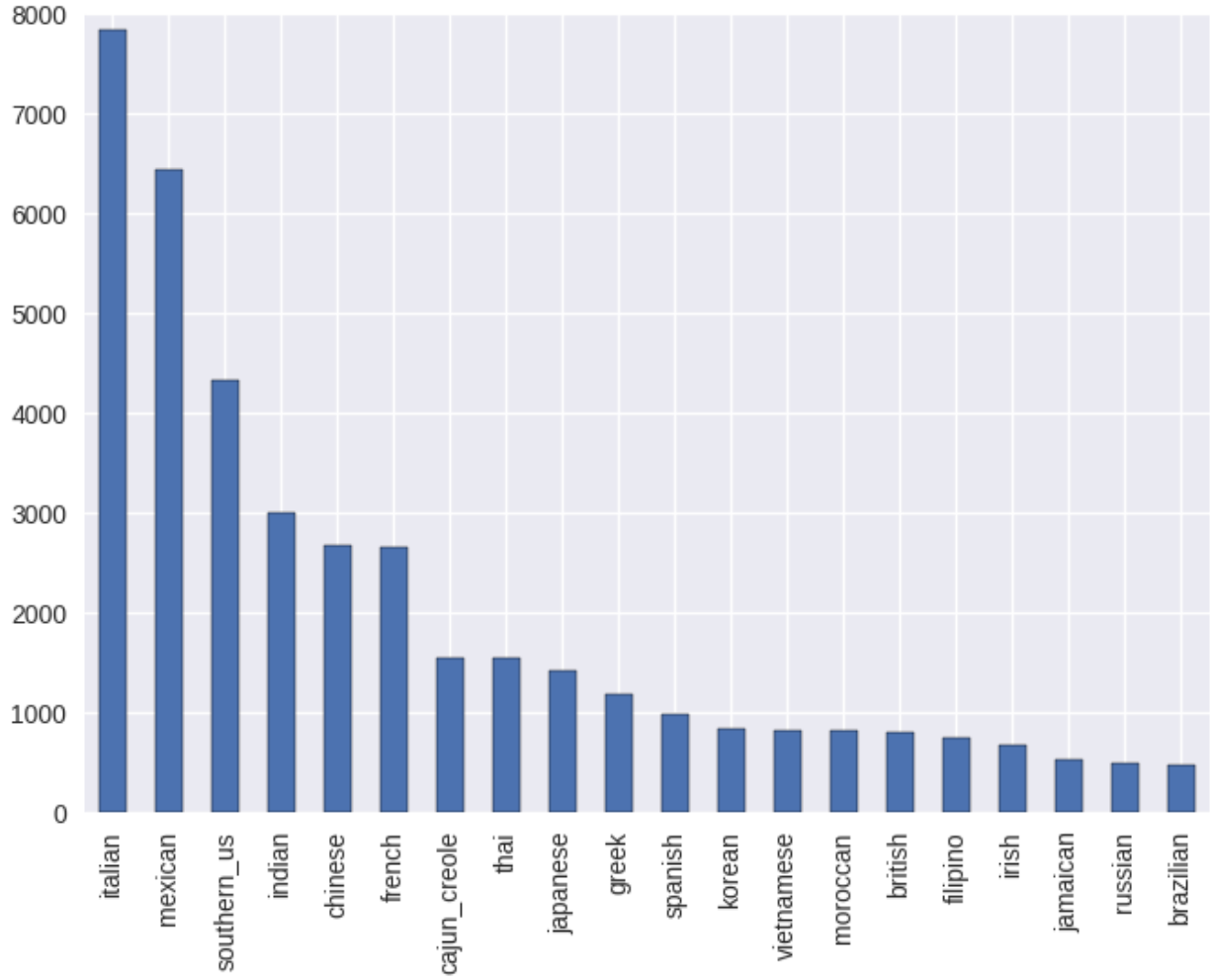
Ingredient	Occurrences in recipes
salt	18049
onions	7972
olive oil	7972
water	7457
garlic	7380
sugar	6434
garlic cloves	6237
butter	4848
ground black pepper	4785
all-purpose flour	4632
pepper	4438
vegetable oil	4385
eggs	3388
soy sauce	3296
kosher salt	3113
green onions	3078
tomatoes	3058
large eggs	2948
carrots	2814
unsalted butter	2782
ground cumin	2747
extra-virgin olive oil	2747
black pepper	2627
milk	2263
chili powder	2036

Table-2

While looking for the ingredients an attempt was made to find out the least used ingredients, which gave a list of some ingredients which appear only once in the entire dataset. If we train the prediction model with these included, they might force the model to overfit. Examples of such recipe items are :
saffron road vegetable broth, gluten-free broth, egg roll skins etc. These outliers have to removed during preprocessing stage.

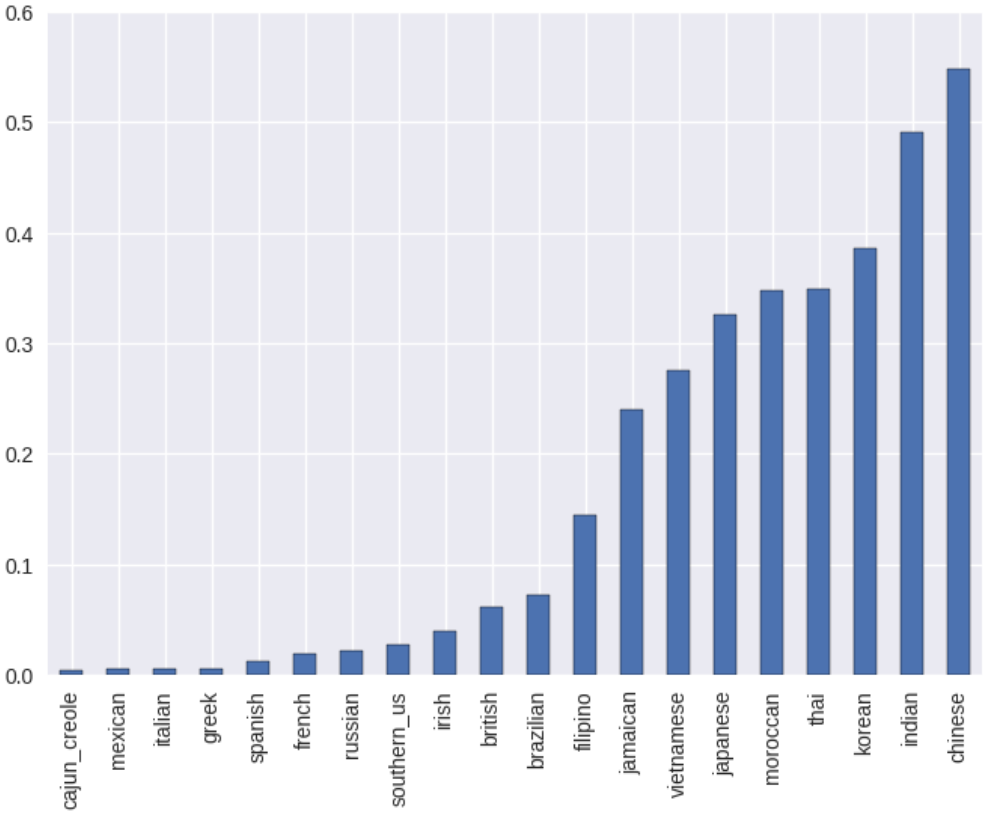
Exploratory Visualization:

The plot below shows the distribution of recipes and cuisines. The X-axis represents the type of cuisine and y-axis the no of recipes in each cuisine. From this plot we can visualize various categories present in the dataset.



In the data analysis section we have seen that the top 10 most used ingredients in the entire dataset as well as the top 10 ingredients used in specific cuisine. But to get more insight while predicting we can try to find the relevance of each ingredient to a particular cuisine. This can be done by calculating the frequency of this ingredient and dividing it with the no of recipes of that cuisine. For example we have seen that ‘garlic’ is one of the top 10 used ingredients in Indian cuisine. The visualization of relevance of this ingredient in each cuisine can seen below.

The x-axis represents type of cuisine and y-axis the relative frequency of ‘ginger’ to appear in recipes.



From above image we can see that Chinese cuisine has around 0.55 frequency, India has close to 0.5.By looking at above visualization we can say that ‘ginger’ as an ingredient is used in Chinese cuisine more than any other cuisine and less used in cajun_creole cuisine.

Algorithms and Techniques:

As this is a classification problem, I wanted to start with decision trees and try other models like random forests and logistic regression and find out which one performs better.

Decision trees:

It is a type of supervised learning algorithm that is mostly used for classification problems. A decision tree is a flowchart ike structure in which each internal node represents a "test" on an attribute (e.g. whether a coin flip

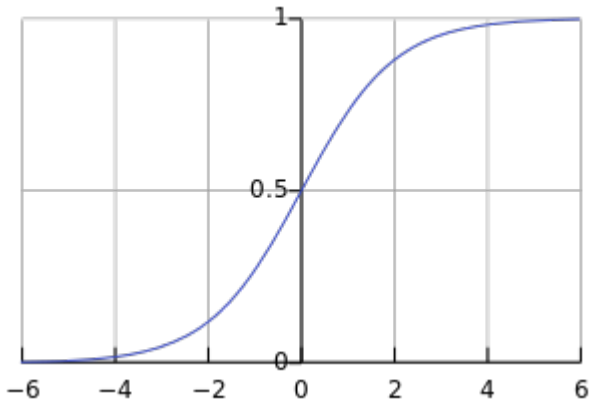
comes up heads or tails), each branch represents the outcome of the test and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represents classification rules. It works for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets. This is done based on most signal can't attributes/ independent variables to make as distinct groups as possible. One of the main advantage of decision tree is that Nonlinear relationships between parameters do not affect tree performance. and they are easy to visualize. The disadvantage is that they have very high bias.

Random Forests :

Random forests is an ensemble method of decision trees. It can also be thought of a form of nearest neighbor predictor. Ensembles are a divide and conquer approach used to improve the performance and accuracy of the model. The main principle of ensemble methods is to group multiple 'weak learners' to combine a 'strong learner'. Because, Random forest builds many trees using a subset of the available input variables and their values, it contains some underlying decision trees that omits the noise generating variables/ features. To classify a new object based on attributes,each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest). Random forest run-times are quite fast and they are able to deal with unbalanced and missing data.

Logistic Regression:

It is a classification algorithm. Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1. The function is f(x) = 1/1+e^(-x) and its plot is as shown below where the y-axis represents f(x)



Logistic regression can be seen as a special case of the generalized linear model. While other regression techniques take the input X and generate a value Y {f(x)}, logistic regression takes the input X to generate output Y which is the value ranged between 0 and 1. It predicts the probability of occurrence of an event by fitting data into a logistic function. The function can be described as logistic(P) = ln(P/(1-P)). Here 'P' is the probability of presence of the characteristic of interest.

Benchmark:

This problem was hosted as a competition in Kaggle and the best accuracy score achieved in the leader-board is '0.832' . Achieving a score near to that would be a pretty good one. Also achieving a small difference of training and validation is also a good measure as it indicates that the model is not over-fitted.

Preprocessing (discussed more in later section) is done on the dataset based on our observations from exploring the data. After that the data set is divided into training and testing data in the ratio of 80% and 20%. Further 10 fold cross-validation is performed on the training data-set to estimate performance of the model.

III. Methodology:

Data Preprocessing:

As seen earlier, the input data is in json format. The dataset contains ingredients of a recipe and their respective label. In-order to use this data set to train the machine learning models, we need to encode this data into a matrix so that the

models can understand. This is done using the ‘countvectorizer’ library of sklearn. By this we will build a matrix with 1s and 0s when ingredients are present. For converting the labels of each recipe ‘LabelEncoder’ library is used to.

Implementation:

First the preprocessed dataset is split into two sets training data and test data in the ration of 80-20. The training data is used to train different predictive models chosen. These models are instantiated from sklearn library. A 10 fold cross validation is performed on the train data and cross-validation scores are calculated. The test data sets are fed to the machine learning models to predict the category of the cuisine. The accuracy score was calculated along with precision, recall, F1-score to measure the performance of each model.

Refinement:

By looking at the performance of the models (represented in results section), Logistic regression performs well among the others. The logistic regression model used in sklearn has the following default parameters :

logisticregression (*penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True,intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100, multi_class='ovr',verbose=0, warm_start=False, n_jobs=1*)

In Order to improve the performance tune I tried to fine tune some the parameters. The documentation of logistic regression implementation in SKlearn library suggests to use ‘newton-cg’ and ‘lbfg’ solvers for multiclass problems. Most of the other parameters correspond to the implementation of the model. But the parameter ‘C’ specifies ‘Inverse regularization strength’, the smaller the value the stronger the regularization. It helps us to reduce overfitting . The scores obtained for these variations are documented in below table.

C	Train score	Validation score	Test score	Precision	Recall	F1-score
0.1	0.7956	0.764	0.767	0.77	0.77	0.76
0.4	0.83	0.78	0.782	0.78	0.78	0.78
0.7	0.841	0.783	0.7857	0.78	0.79	0.78
1	0.85	0.783	0.785	0.78	0.79	0.78
1.3	0.855	0.783	0.785	0.78	0.79	0.78
1.6	0.858	0.783	0.785	0.78	0.79	0.78
1.9	0.861	0.782	0.784	0.78	0.78	0.78
2.2	0.864	0.782	0.7832	0.78	0.78	0.78
2.5	0.866	0.781	0.782	0.78	0.78	0.78

The above results hold true for both solvers ‘newton-fg’ and ‘lbfgs’ (Limited memory BFGS optimization algorithm). The score values represented are rounded until two decimals after zero. From this table we can see that for C=0.7 the difference between train and validation score is less compared to other cases, achieves an accuracy score of 0.785 with good precision and recall scores. These can be used as our final model parameters.

Analyzing Results and Performance :

The performance of various models was analysed based on the scores achieved by them. They are documented in the below table.

Model	Train score	Validation score	Test score
Decision Trees	0.9996857224	0.63	0.648
Random Forest	0.9997485779	0.753	0.752
Losgistic Regression	0.8513781074	0.78	0.785

Table -3

It can be observed that logistic regression performs better when compared to other models with an accuracy score of 78.5%. A 10 fold cross validation was performed in-order to detect any overfitting. In case of decision trees the train score is 0.999 while validation is 0.63 which indicates that this model overfits the data. When we move from decision trees to random forests the overfitting slightly decreases as it takes decision based on multiple trees. While coming to logistic regression the difference in train and validation scores are considerably lesser than other models, this lead me to choose logistic regression as my model

to solve the problem at hand.

For each model various metrics which we have discussed before are also calculated and presented below.

Model	Precison	Recall	F1-score
Decision Trees	0.65	0.65	0.65
Random Forest	0.76	0.75	0.74
Losgistic Regression	0.78	0.79	0.78

Table -4 : Various scores achieved on test data

From above table we can see that though the accuracy score of logistic regression (LG) model on test data is 78.5% , the precision score is 78% and we can try to improve these scores to improve the overall performance of the classification model.

For analysing the performance of the models in much depth, confusion matrix is used.A part of the confusion matrix generated when logistic regression model was used can be seen below:

	italian	mexican	southern_us	indian	chinese	french
italian	54	0	1	0	4	3
mexican	1	62	1	0	1	17
southern_us	3	1	221	0	0	16
indian	2	0	2	445	5	3
chinese	2	1	0	17	111	3
french	0	8	2	0	1	322
cajun_creole	0	0	0	0	0	12
thai	0	3	3	1	2	2
japanese	0	13	1	0	0	12
greek	0	4	9	2	0	65
spanish	0	2	0	1	2	1
korean	1	1	0	31	4	6
vietnamese	0	0	0	18	3	0
moroccan	0	0	4	0	4	12
british	1	0	1	1	0	4
filipino	0	2	1	0	2	16

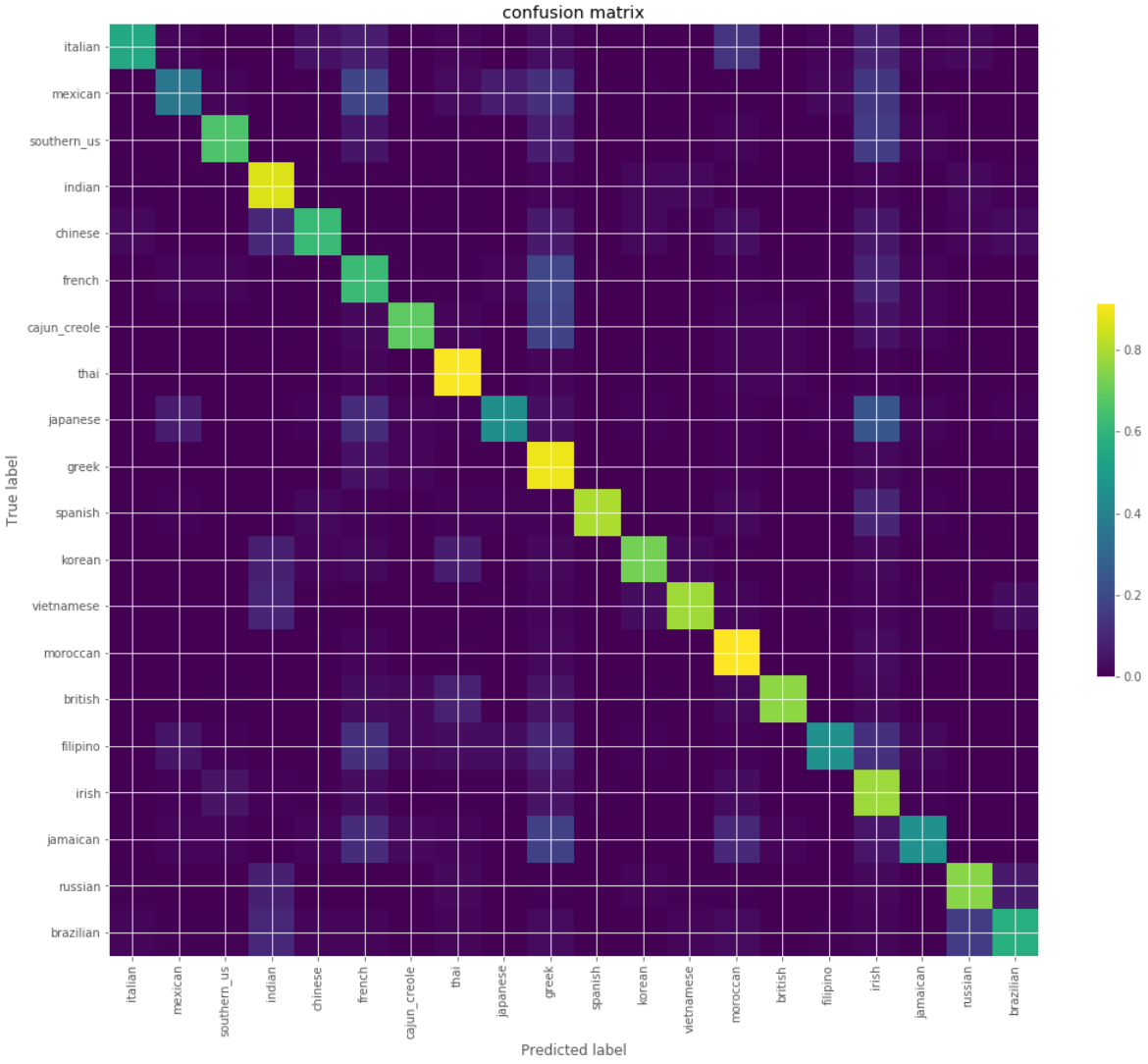
irish	2	8	35	2	4	25
jamaican	1	2	3	1	1	13
russian	3	0	1	25	0	0
brazilian	3	0	0	15	3	1

Confusion matrix table

Here the column represents the predicted cuisines whereas the rows represent the actual cuisine to which they belong. For example in the first column 'Italian' we can see that 54 recipes are predicted correctly, where are 3 southern_us recipes are also categorized as Italian which is incorrect. So with the help of this confusion matrix we can identify for which cuisines the model good/bad job and use to find the similarity in cuisines. Instead of looking at the numbers this matrix can also be visualized; this visualization is shown below. Using the colour coding mentioned on the right side of visualization, we can identify which and to what degree other cuisines are mis classified (we are identifying true positives and false positives here). For example the indian cuisine has a yellow colour, which says that more than 80% of Indian recipes are classified correctly, while some recipes of of korean cuisine, vietnamese are misclassified as Indian.

Evidently, the most difficult cuisine to classify is Russian, the easiest cuisine to classify is Mexican. Also we can identify that french food is the most similar to Italian food in terms of ingredients and Japanese food is reportedly similar to Chinese food which is true as they share most of the ingredients.

OBJ



Confusion matrix Visualization

Comparison with benchmark:

From the discussion in this section, we have seen that logistic regression achieves an accuracy score of 0.785. On comparison to the maximum value achieved in the kaggle competition which is 0.83, it is ‘0.047’ less but still close by. However the logistic regression model has achieved a very less difference between train and validation scores compared to other models, which indicates less overfitting. Also this model has achieved almost same Precision, Recall, F1-

score values. Based on this I would say that this model does a good job at solving classifying the cuisines based on ingredients of the recipes.

Reflections :

In this project a real world problem was chosen to be solved using machine learning techniques. The process can be summarized as below:

1. A problem was identified.
2. Relevant labelled data was found.
3. Metrics were decided to measure the performance of the classifier.
4. Data Exploration was carried to get insights into the data.
5. Data was preprocessed
6. Different models were tried, fine tuning of the model was done.
7. Validation techniques were used to evaluate the models.
8. The results were analysed using confusion matrix to get insight into how the model performs for different types of input.

I found the steps 6,8 to be challenging as I have to analyse performance of classifier and try to understand how the model works while classifying certain cuisines and how good they are at classifying cuisines that have similar ingredients.

Conclusion and future scope:

The logistic regression model (with C=0.7) performs with an accuracy rate of 78.5% for classification of cuisines, which is a good performance although there is still room for improvement. While looking at the ingredients it was observed that certain ingredients have description for which adjectives, specific type, company/brand names are used; for example consider eggs, brown eggs, 4 large eggs. These are different, still the ingredient is ‘egg’ and it's highly unlikely that using different type of ingredient would change the cuisine. For the future scope, if we have a more advanced pre-processing to convert all these into single ingredient, this might increase performance of the classifier.

It is also worth in investigating why certain cuisines are intrinsically more difficult to accurately classify. As seen in confusion matrix analysis that Russian

recipes are difficult to classify. This can be challenging because they can be having only common ingredients which are found in all cuisines.

The confusion matrix further reveals that certain cuisines are often misclassified as another, thus the model considers such cuisine pairs very ‘similar’. This can be because ‘similar’ cuisine pairs have relatively more common ingredients than cuisine pairs that are considered ‘dissimilar’. For example many instances of misclassification occurred among asian cuisines were often because the cuisines share many common ingredients. Effort can be focussed on solving this ‘similarity’ issue which improves the overall accuracy.

References :

1. http://scikitlearn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

2. http://scikit-learn.org/stable/modules/cross_validation.html

3. http://scikit-learn.org/stable/modules/model_evaluation.html

4. http://www.scipy-lectures.org/intro/numpy/array_object.html

5. https://en.wikipedia.org/wiki/Yummly

6. https://en.wikipedia.org/wiki/Decision_tree

7.http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

8. https://www.kaggle.com/c/whats-cooking

9. http://scikitlearn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html