IME-672A

Group-9

What's Cooking

Made By-

1.)Akash Gangwar(170067)
2.)Aditya Rai(170053)
3.)Aman Raj Singh(170087)
4.)Anshika Verma(170131)
5.)Akash Kumar(170069)
6.)Akshita Jha(170076)

Mentored By-

Faiz Hamid

Index

1.)Problem Statement

The training dataset consists of different recipes distinguished by Recipe_id. Each recipe has a list of ingredients and the cuisine which it belongs to. The objective is to develop a model from the given dataset which classifies a recipe's cuisine given its ingredients.

2.)Data Understanding

The dataset is in JSON format. The dependent variable is cuisine and independent variable is ingredients.

- Train data consists of 39774 dishes as dish-ids, ingredients and the cuisine they belong to.
- Test data consists of 9944 dishes as dish-ids and ingredients.
- There are 20 different unique cuisines present in the data
- There are total 428275 number of ingredients
- Among which 6703 are unique

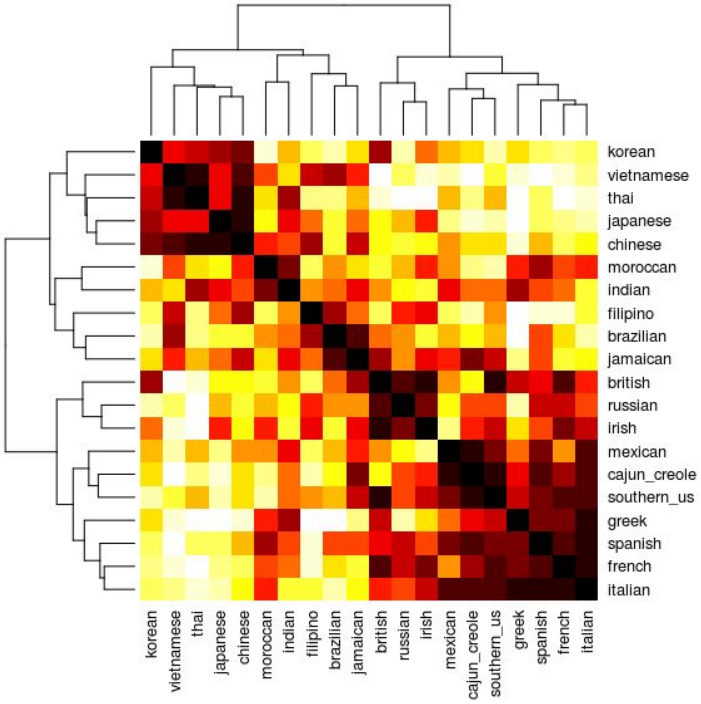
The dataset does not have many attributes and in our dataset the main attribute is a list of set of ingredients.Each set represents a dish.The data is basically categorical data and data types is list of strings

2.1)Data Visualization:

To gain an insight of data we created various plots whose drive links are given below:-

1.)Plot of Cuisines with the count of no Ingredients present in a given cuisine.
2.)Plot of distribution of recipes in each cuisine.
3.)Histogram of Recipes length with respect to Ingredients.
4.)Plot of most common 20 Ingredients.
5.) Plot of number of unique ingredients used in each cuisine
6.)Features Importance plots of Ingredients based on TF-IDF measure
7.)Plot of some important ingredient with all cuisines

To check the similarity and correlation we performed word embedding using Glove vectors .We performed training on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase of the interesting linear substructures of the word vector space.Correlation heap map obtained looks like this-



Similarly we also used a t-SNE to reduce the data set obtained from Glove to two attributes and plotted the similarity of different cuisines which can be obtained by Clicking Here.

3.)Data Preprocessing

3.1)Data Cleaning:

The given dataset contains some quirks and needs to be cleaned.We took the following steps in Data Cleaning-

a).Checked the presence of missing values which came out to be NIL.

b).The incoming data can be loosely structured, multilingual, textual or might have poor spelling for example:

- Some ingredients may contain special symbols that are not relevant like @,# etc and are needed to be removed.
- Since text case doesn't change the meaning of the word so its better to convert them into single case.
- Punctuation elements like " " , - _ are not useful and shall be removed .For ex, Chilli Flakes and chilli-flakes are same.
- Stopwords are those words which add no value .They don't describe any sentiment. Examples are 'i','me','myself','they','them' and many more. Hence, these words should be removed if present.

- Stemming means bringing a word back to its roots. It is generally used for words which are similar but only differ by tenses.To treat word and word derivatives as same we stemmed our dataset.

c).Checked and removed duplicate data which resulted in numerosity reduction.
d).Set the ingredient data to unique for each row.

Ideally, we have done a Data cleaning such that it returned a more relevant version of these ingredients.

3.2)Data Reduction and Transformation :

a.) Combined the ingredients of a row to form a text document which gave a data set of about 40,000 documents with a class label attached to it.

b.) Removed the Duplicate rows present in Data which resulted in a little of numerosity Reduction.

Data Reduced: 39774 to 39256

c.)Document Term Matrix:

- From the text data (set of ingredients), we have created document term matrix (DTM).
- 'tm' package in R creates DTM•From the corpus of ingredients, the method sets '1' when that particular ingredient appears for a dish and '0' otherwise.
- Sparse terms from DTM has been removed where frequency of those terms for whole training dataset are less than 3

d.)TF-IDF Matrix:

- This is a technique to quantify a word in documents, The weights to each word are computed which signifies the importance of the word in the document and corpus with respect to other documents.
- From the DTM matrix we get the term frequency of each word in a document and IDF is the inverse of document frequency providing informativeness for term t.
- It gives more weight to those ingredients which occurs more often in one class of documents given it is less present in other classes documents
- These tf-idf value is assigned to each term in document and a matrix is formed
- Reduced parse number of columns from 6703 ingredients to 3010.

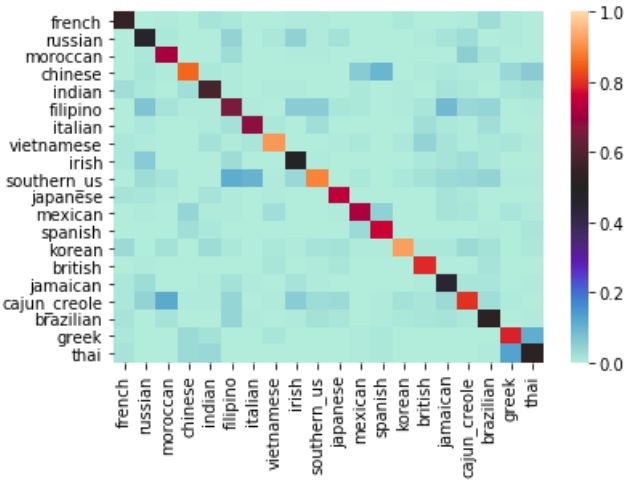
The TF-IDF matrix will be the input to most of our training data except in Naive Bayes where DTM is the training data.

4.)Data Mining Algorithms Applied

For training the Models and to check its performance we split the training data into two parts,80% of the train data was used for training and rest 20% was used for it's validation.With this division of the dataset the following DM algorithms were applied:

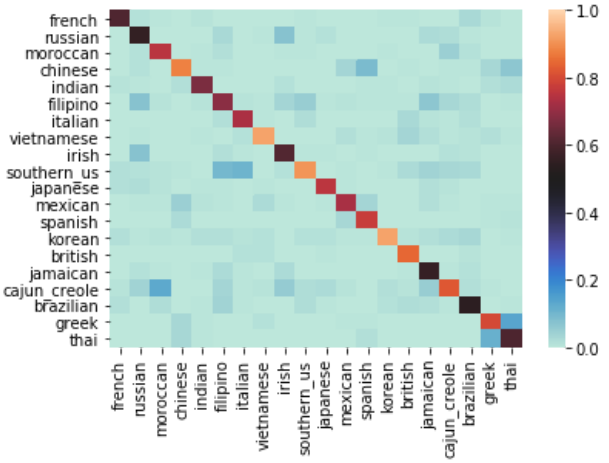
4.1.)Artificial Neural Networks: We build a four layered ANN with an output layer having 20 nodes each representing a unique class.The input layer consist of 3010 input dimensions and weights were assigned using 'he_normal' kernel initializer for further layers.There are 2 hidden layers each with 1000 node units and the activation function used for them is "relu".The activation function used for the output layer containing 20 nodes is "softmax" .In training the input is feed in batch of 512 with maximum iterations set at 20 for the training data with validation split of 0.1. The Confusion Matrix(Plotted on Heat Map) and other evaluation Metrics which are given below

Metrics	Measure(%)
Accuracy	80.20
Precision	76
Recall	71
F1-Score	73
Support	9944

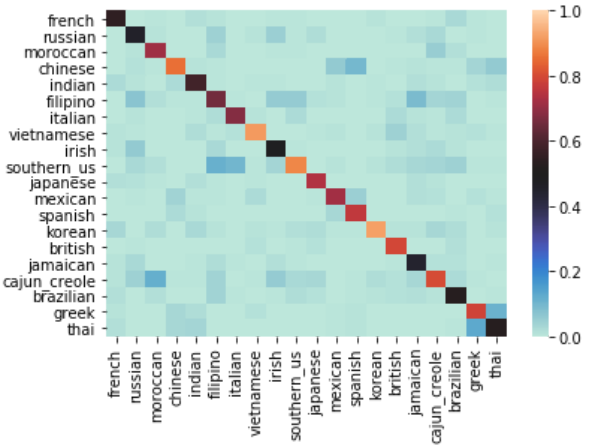


4.2.)Support Vector Machine: We built two SVM models using the LINEAR and RBF Kernels separately.Similar to ANN we trained our Model on 80% of the training data and evaluated the Model on the remaining data . Here also we evaluated our both classifiers based on Confusion Matrix and other metrics .Here due to space constraints we will be showing the Confusion matrix of rbf kernel only

Metrics	RBF(%)	Linear(%)
Accuracy	82	79.05
Precision	78	78
Recall	73	69
F1-Score	76	72
Support	9944	9944



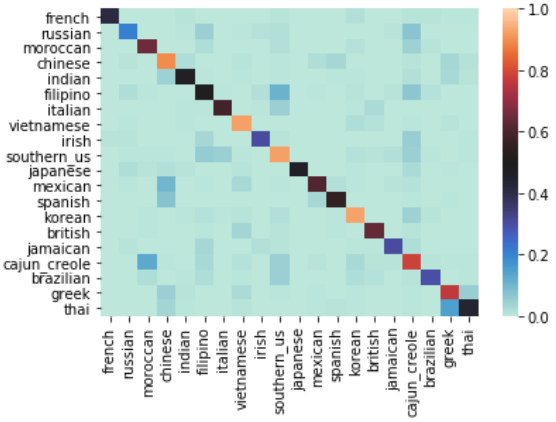
4.3.)Logistic Regression: Similar to ANN and SVM here also we trained the Logistic Regression with 'C=20' model on 80% of training and evaluated the performance of model on the remaining 20% . The confusion matrix heat-map and other measures are given below.



Metrics	Measure(%)
Accuracy	79.41
Precision	76
Recall	69
F1-Score	72
Support	9944

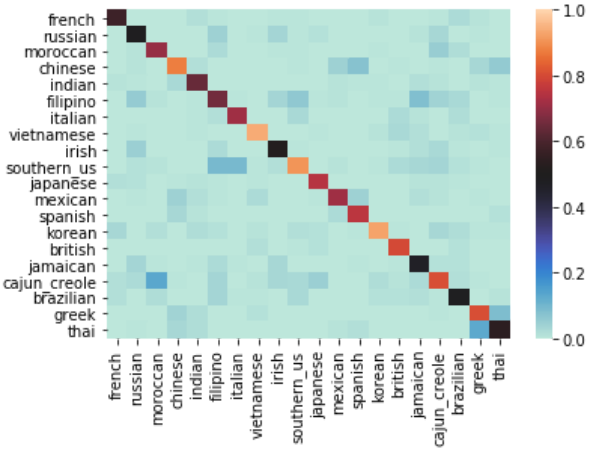
4.4.)Random Forest:A single decision tree tends to over-fit the model.Random Forest basically improves the accuracy of decision tree since the ensemble of weak learner predict far better than a single decision tree.For our given set of input values from tf-idf matrix.we created Random forest with 10 number of features and n_esimators set as 600.Here also the training will be done on 80% and validation on the remaining 20%. The confusion matrix and other measures were plotted and calculated for our split train data.

Metrics	Measure
Accuracy	75.4
Precision	81
Recall	59
F1-Score	65
Support	9944



4.5.)Naive Bayes: Unlike the other models here the input data was a Document Term matrix of training data.From the DTM we formed a probability matrix for each cuisine.The naive bayes classifier predicts the cuisine on the basis of maximum value of probability for our train data set.This classifier gave the accuracy of **82%** on training data

4.6.)Ensemble Classifier: In this classifier we ensemble the three classification models into one classifier.The three models that we ensembled were Logistic Regression,SVM(Linear Kernel) and Random forest.The training was done in the same way as above models by a split of 80-20 percent.The confusion matrix Heat-Map and other model evaluation metrics are given below.



Metrics	Measure(%)
Accuracy	80.6
Precision	78
Recall	70
F1-Score	74
Support	9944

5.)Performance Comparison

Models\Metric	Accuracy	Precision	Recall	F1-score	Support
ANN	80.20	76	71	73	9944
SVM(RBF)	82	78	73	76	9944
Logistic Regression	79.41	75	69	72	9944
SVM(Linear)	79.45	78	69	72	9944
Random Forest	75.4	81	59	65	9944
Ensemble	80.6	78	70	74	9944

If we compare all the metrics of the model we can eliminate Random forest due to its low Recall.Similarly at the same time SVM(Linear) and Logistic Regression has comparable lower Accuracy,Precision,Recall and F1-score as compared to other

models.So these three are weak Classifiers so we build a Ensemble of these three Models which has a second best Accuracy of all models trained.From all the metrics we can see that **SVM(RBF Kernel)** has the best of all the metrics so we thought it would be the best model to score our given test data.The second best that we thought was **Artificial Neural Network** as it also has comparable good Accuracy and precision as compared to other Models.So when final Submissions were mades on Kaggle scores were something like this-

6)Interpretation of Results

Models	Kaggle Score
ANN	80.752
SVM(RBF)	82.119
Ensemble	79.475
Naive Bayes	80.26
Logistic Regression	78.388
SVM(Linear Kernel)	78.207

So as predicted best accuracy was obtained obtained by SVM(RBF) on the Kaggle.The maximum accuracy which anyone could have gotten on this data on Kaggle was 83%.So, as compared to maximum SVM(RBF) is very good model.We could increased the accuracy by including the SVM(RBF) in Ensemble model with ANN but then there came a catch of time complexity.Both Ensemble and SVM have high time complexity in training so including both in one would have been exponentially increased it.So for a little increase in Time Complexity with exponential Time Complexity was not worth it for our case.

7.)Recommendations:

Since Accuracy could have been increased by ensembling a SVM(RBF) with ANN and other models but that would require a device of higher Computational power and would also have a high time Complexity.We were short on both of these.But if any professional organisation is well equipped with both of these resources should try the Ensemble of SVM(RBF) with ANN and other models.The Model will surely work better than all of them.