Class Project: What's Cooking

FNU Anirudh and Venkata Prudhvi Raj Indana

December 10, 2015

INTRODUCTION

The goal of this project is to study the characterstics of dishes in training set and learn traits associated to particular cuisine and classify cuisine based on ingredients in test set.

In the report, we describe the data we use and how we process it. Then we explore techniques for classification and also look at unsupervised learning techniques for more information. JSON files for training and test dataset has been provided.

1. DATASET

Our Main dataset is train data set which we use to train our model and then implement our model to classify cuisines in test dataset. First let us analyze dimension of training dataset. Both training and test datasets are in JSON format and were read using library jsonlite in R.

```
## [1] 39774     3
```

There are 39774 rows and 3 columns in our training dataset.

Let us now analyze the column names in our dataset

```
## [1] "id"          "cuisine"     "ingredients"
```

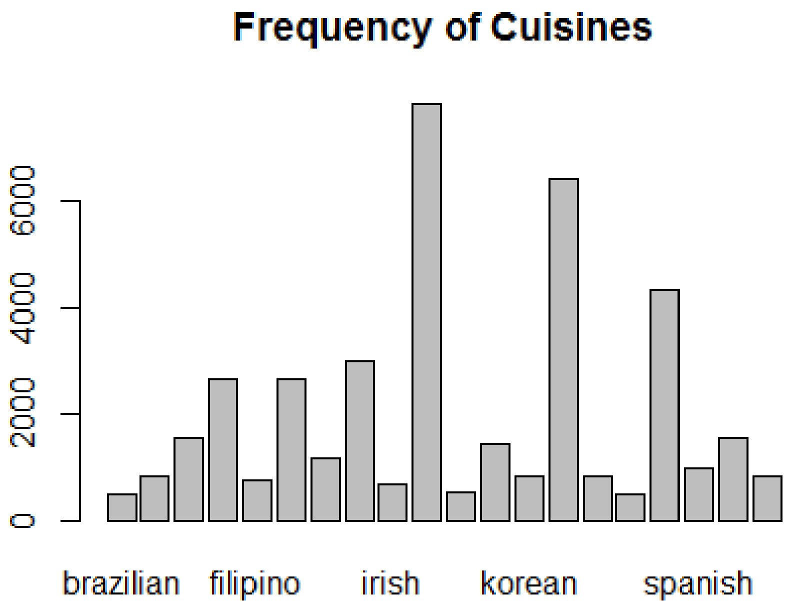
From this we see that there are 3 colummns in training data namely id, cuisine and ingredients.

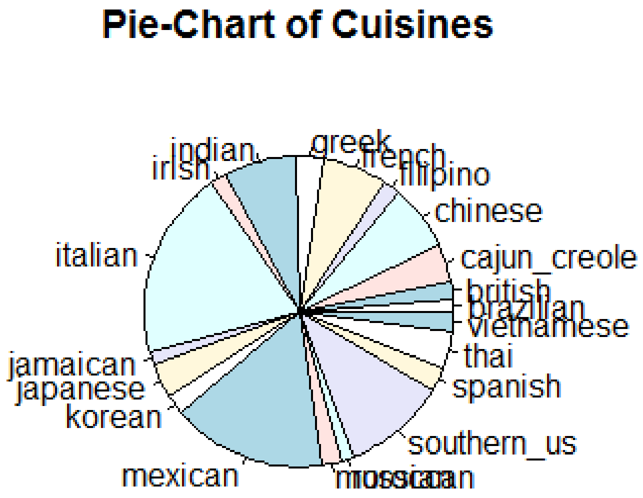
Let us print first row to see how our dataset looks.

```
##      id cuisine
## 1 10259   greek
##
ingredients
## 1 romaine lettuce, black olives, grape tomatoes, garlic, pepper, purple on
ion, seasoning, garbanzo beans, feta cheese crumbles
```

From above we see that all the ingredient that are associated with a dish are stored in single vector hence in order to classify we will have to split each ingredient and check it's occurence in other dishes to understand the pattern.

Let us now Analyze how many unique cuisines are there in our dataset and how they are distributed and how many times does each appear in our training data.





```
##
##    brazilian      british cajun_creole      chinese     filipino
##          467          804         1546         2673          755
##       french        greek       indian        irish      italian
##         2646         1175         3003          667         7838 
##     jamaican     japanese       korean      mexican     moroccan 
##          526         1423          830         6438          821 
##      russian  southern_us      spanish         thai   vietnamese 
##          489         4320          989         1539          825

## [1] 7838
```

We find that there are 20 different or unique cuisines in our dataset. From barplot we see that there are couple of cuisines which have high frequency, To get better understanding we plot Pie Chart and find out that Italian and Mexican cuisines have high frequency compared to other dishes.We print all the cuisines in our dataset and number of times each cuisine repeats and find that there are 7838 Italian dishes out of 39,774 dishes. Our data is categorical data.

2. TECHNIQUES

We have used following classification algorithms in our project to classify cuisines:-

2.1 Naive Bayes

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach to model a predictive modeling problem probabilistically.

Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.

The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class.

To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.

Naive bases is often described using categorical data because it is easy to describe and calculate using ratios. A more useful version of the algorithm for our purposes supports numeric attributes and assumes the values of each numerical attribute are normally distributed (fall somewhere on a bell curve). Again, this is a strong assumption, but still gives robust results.

Reason for Using Naive Bayes

- Naive Bayes can handle missing data
- We can use Naive Bayes for Categorical data by calculating frequency of observation.
- Probabilities for each attribute are calculated independently from the training dataset. We can use a search algorithm to explore the combination of the probabilities of

different attributes together and evaluate their performance at predicting the output variable. We can use probabilities to do feature selection.

- Naive Bayes can give increased performance and focus on the elements of the problem that are more difficult to model by identifying and separating out segments that are easily handled by a simple probabilistic apporach.
- One of the Benefit of Naive Bayes is that we can re-calculate probabilities as data changes.
- An interesting point about Naive Bayes is that even when the independence assumption is violated and there are clear known relationships between attributes, it works anyway.

2.2 K- Nearest Neighbour

The purpose of the k Nearest Neighbours (kNN) algorithm is to use a database in which the data points are separated into several separate classes to predict the classification of a new sample point.

we consider each of the characteristics in our training set as a different dimension in some space, and take the value an observation has for this characteristic to be its coordinate in that dimension, so getting a set of points in space. We can then consider the similarity of two points to be the distance between them in this space under some appropriate metric.

The algorithm can be summarised as:

- A positive integer k is specified, along with a new sample
- We select the k entries in our database which are closest to the new sample
- We find the most common classification of these entries
- This is the classification we give to the new sample

Reason for Using KNN Algorithm

- Cost of Learning process is zero.
- No assumptions about the characterstics of the concept to learn have to be done.
- Complex concepts can be learned by local approximation using simple procedures.

2.3 Neural Network

Neural networks or artificial neural networks (ANNs) have received a lot of attention for their abilities to 'learn' relationships among variables. They represent an innovative technique for model fitting that doesn't rely on conventional assumptions necessary for standard models and they can also quite effectively handle multivariate response data. As a part of this project we want to train on multivariate data from whats cooking and compare it with other bench mark algorithams.

Reasons for using Neural Networks

- Ability to implicitly detect complex nonlinear relationships between dependent and independent variables.

- Ability to detect all possible interactions between predictor variables.
- Availability of multiple training algorithms.

2.4 Random Forest

Random Forests grows many classification trees. To classify a new object from an input vector, put the input vector down each of the trees in the forest. 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).

Each tree is grown as follows:

- If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data. This sample will be the training set for growing the tree.
- If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.
- Each tree is grown to the largest extent possible. There is no pruning.

Reasons for using Random Forest

- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It offers an experimental method for detecting variable interactions.

3.RESULTS

3.1 Naive Bayes

We run the Naive Bayes Algorithm after removing redundant features or ingredients from our training dataset.We will first try to predict our accuracy on training dataset and then try to predict cuisine in test dataset.

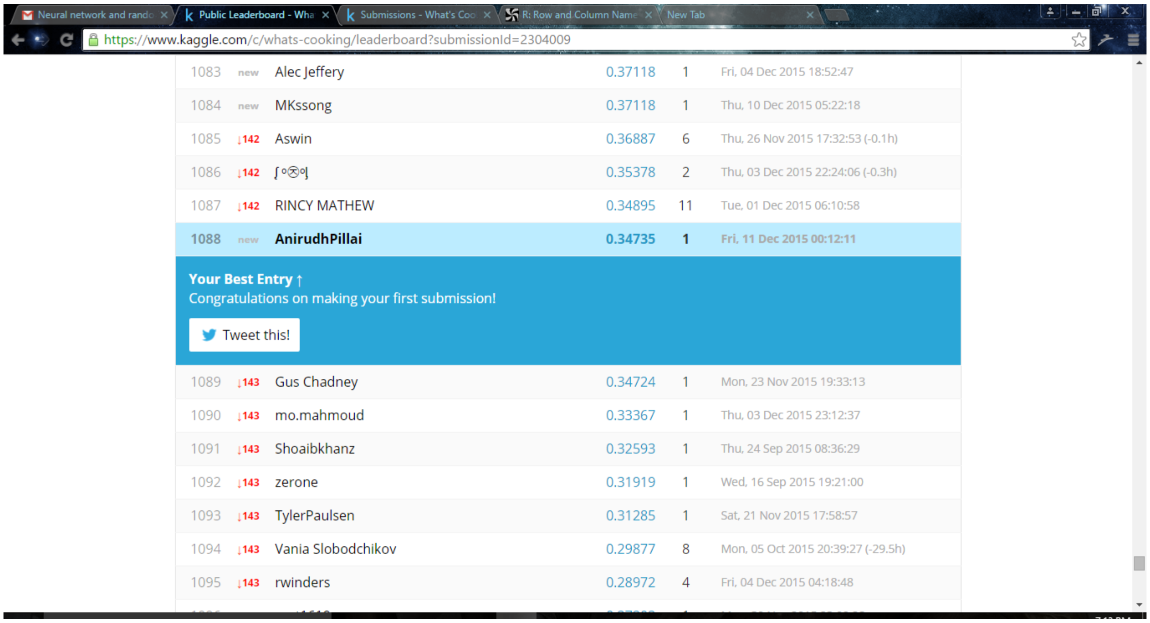
```
## [1] "Naive Bayes Predictions"

## t
##    brazilian      british cajun_creole      chinese     filipino
##          203         4367         1755         1170          461
##       french        greek       indian        irish      italian
##         1427         1300         2382         3666         4598
##     jamaican     japanese       korean      mexican     moroccan
##          379          631         4850         4227         1642
##      russian  southern_us      spanish         thai   vietnamese
##         1555          663          931          972         2595

## [1] "Accuracy of Naive Bayes is ="
## [1] 40.46362
```

We get only 40% accuracy on Implementing Naive Bayes. Let us now predict cuisine in our test data set.

```
## t
##     brazilian      british cajun_creole      chinese     filipino 
##            40          612          452          263          125 
##        french        greek       indian        irish      italian 
##           203          266          798         1162         1169 
##      jamaican     japanese       korean      mexican     moroccan 
##            83          152         1588          888          231 
##       russian  southern_us      spanish         thai   vietnamese 
##           608           87          207          189          821 
```



1083	new	Alec Jeffery	0.37118	1	Fri, 04 Dec 2015 18:52:47
1084	new	MKssong	0.37118	1	Thu, 10 Dec 2015 05:22:18
1085	↓142	Aswin	0.36887	6	Thu, 26 Nov 2015 17:32:53 (-0.1h)
1086	↓142	ʃ ᵒ㉨ᵒʅ	0.35378	2	Thu, 03 Dec 2015 22:24:06 (-0.3h)
1087	↓142	RINCY MATHEW	0.34895	11	Tue, 01 Dec 2015 06:10:58
1088	new	**AnirudhPillai**	**0.34735**	**1**	**Fri, 11 Dec 2015 00:12:11**

Your Best Entry ↑
Congratulations on making your first submission!

Tweet this!

1089	↓143	Gus Chadney	0.34724	1	Mon, 23 Nov 2015 19:33:13
1090	↓143	mo.mahmoud	0.33367	1	Thu, 03 Dec 2015 23:12:37
1091	↓143	Shoaibkhanz	0.32593	1	Thu, 24 Sep 2015 08:36:29
1092	↓143	zerone	0.31919	1	Wed, 16 Sep 2015 19:21:00
1093	↓143	TylerPaulsen	0.31285	1	Sat, 21 Nov 2015 17:58:57
1094	↓143	Vania Slobodchikov	0.29877	8	Mon, 05 Oct 2015 20:39:27 (-29.5h)
1095	↓143	rwinders	0.28972	4	Fri, 04 Dec 2015 04:18:48

We get very low accuracy on our test dataset as seen above. We should try a different algorithm.

3.2 K-Nearest Neighbours

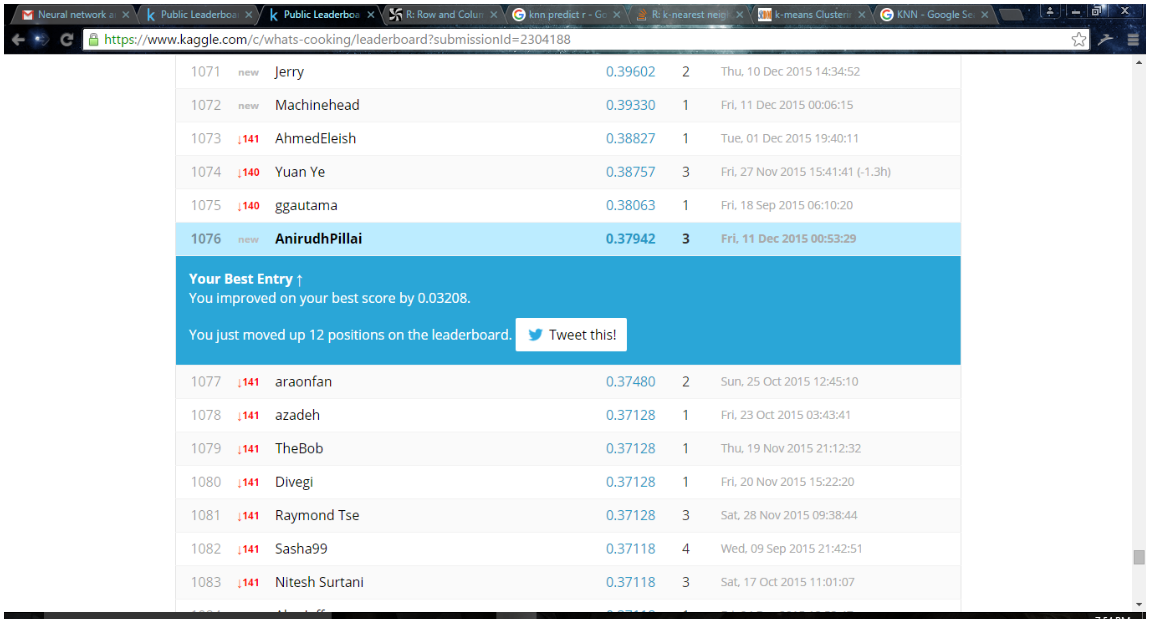
We try K-Nearest Neighbours Algorithm to predict cuisine in our training dataset. We split our original training dataset into training and test dataset. We have chosen first 30,000 dishes as training and remaining 9774 as test dataset, We apply KNN to predict cuisine on test dataset.

```
## [1] "The Accuracy of K- Nearest Neighbours is ="

## [1] 56.90608
```

We get 57% accuracy on Implementing K Nearest Neighbour. Let us now predict cuisine in our test data set.

```
## m2
##     brazilian      british cajun_creole      chinese     filipino
##           145           22          193          205           27
##        french        greek       indian        irish      italian
##           289           56          765           68         3241
##      jamaican     japanese       korean      mexican     moroccan
##            24          509           82         1789           18
##       russian  southern_us      spanish         thai   vietnamese
##             2         2399           29           57           24
```



https://www.kaggle.com/c/whats-cooking/leaderboard?submissionId=2304188

1071	new	Jerry	0.39602	2	Thu, 10 Dec 2015 14:34:52
1072	new	Machinehead	0.39330	1	Fri, 11 Dec 2015 00:06:15
1073	↓141	AhmedEleish	0.38827	1	Tue, 01 Dec 2015 19:40:11
1074	↓140	Yuan Ye	0.38757	3	Fri, 27 Nov 2015 15:41:41 (-1.3h)
1075	↓140	ggautama	0.38063	1	Fri, 18 Sep 2015 06:10:20
1076	new	**AnirudhPillai**	0.37942	3	Fri, 11 Dec 2015 00:53:29

Your Best Entry ↑
You improved on your best score by 0.03208.

You just moved up 12 positions on the leaderboard. Tweet this!

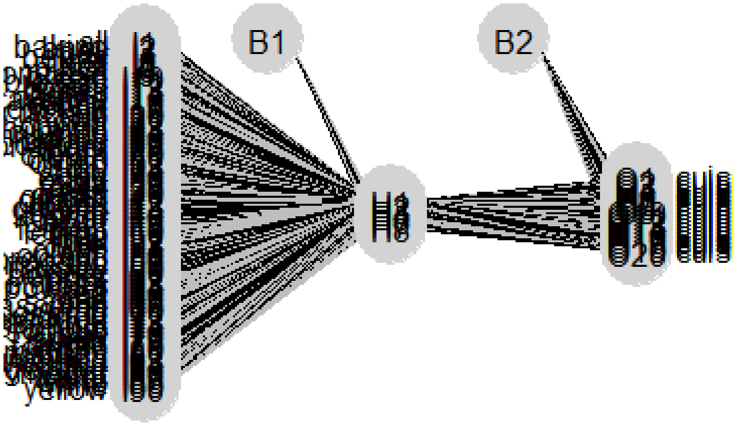
1077	↓141	araonfan	0.37480	2	Sun, 25 Oct 2015 12:45:10
1078	↓141	azadeh	0.37128	1	Fri, 23 Oct 2015 03:43:41
1079	↓141	TheBob	0.37128	1	Thu, 19 Nov 2015 21:12:32
1080	↓141	Divegi	0.37128	1	Fri, 20 Nov 2015 15:22:20
1081	↓141	Raymond Tse	0.37128	3	Sat, 28 Nov 2015 09:38:44
1082	↓141	Sasha99	0.37118	4	Wed, 09 Sep 2015 21:42:51
1083	↓141	Nitesh Surtani	0.37118	3	Sat, 17 Oct 2015 11:01:07

We get accuracy of 38% on test dataset which is slightly better than our naive bayes algorithm.

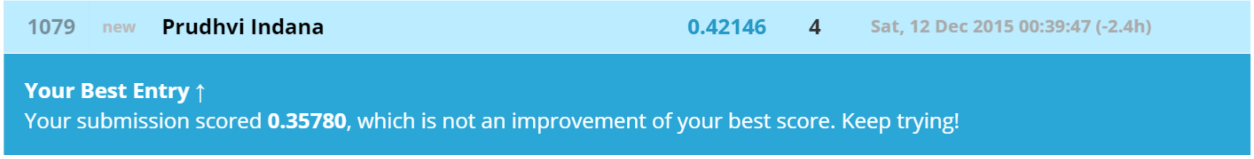
3.3 Neural Network

Learning and predicting multinomial data R has a package NNET. NNET uses Back propagation and resilient back propagation to train on the train dataset. We passed parameter values such that the max number of iterations are 1000 and number of weights each input has is 8.

Once NNET is trained we can find graph resultant NNET, We tried to improve the visibility but considering the fact that there are lots of input and output features increased the number of weights there by making the graph look clumsy.



```
## Prediction_nn
##    brazilian      british cajun_creole      chinese     filipino 
##           31            4          547          137           18 
##       french        greek       indian        irish      italian 
##          305          482          583           44         2214 
##     jamaican     japanese       korean      mexican     moroccan 
##          840          105           41         2291          534 
##      russian  southern_us      spanish         thai   vietnamese 
##           14         1253          235          240           26 
```



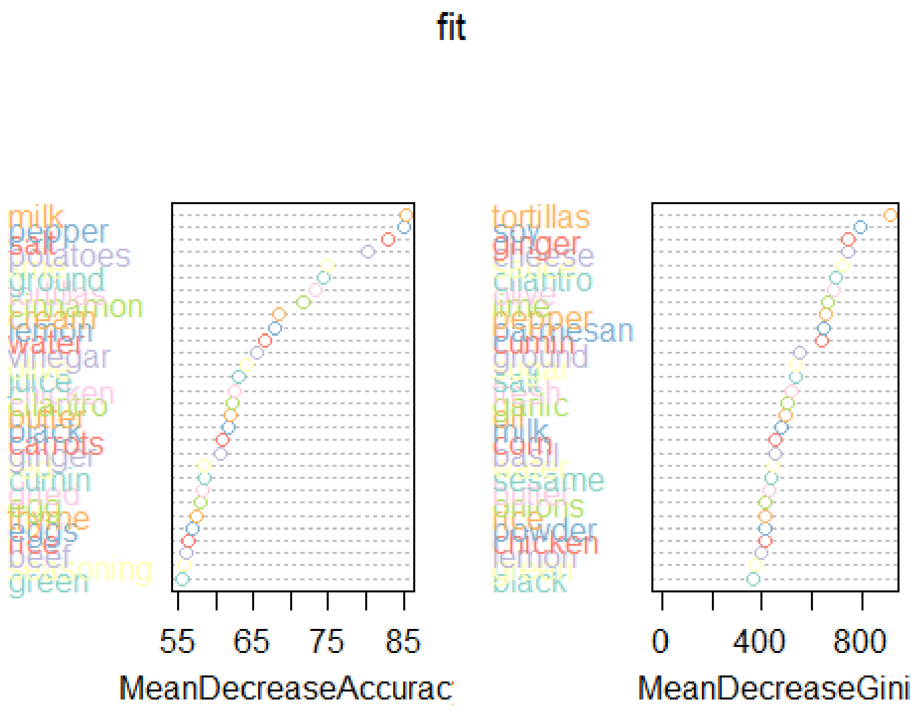
1079 new Prudhvi Indana 0.42146 4 Sat, 12 Dec 2015 00:39:47 (-2.4h)

Your Best Entry ↑
Your submission scored **0.35780**, which is not an improvement of your best score. Keep trying!

We got close to 36% accuracy for our neural network.

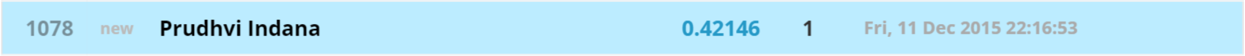
3.4 Random forest.

Since Random forest algorithm has ensemble learning and 'bagging', we choose to implement this algorithm to serve as a base line algorithm both to test and compare Kaggle score obtained with other algorithms(KNN, Neural network). Training Random forest on whole data produced better results than any of our previous algorithms.



```
table(Prediction)

## Prediction
##    brazilian      british cajun_creole      chinese     filipino
##           61            8          326          330            5
##       french        greek       indian        irish      italian
##          277           62          724           30         3189
##     jamaican     japanese       korean      mexican     moroccan
##          120          192           55         2978          172
##      russian  southern_us      spanish         thai   vietnamese
##            5         1172           39          149           50
```



1078 new **Prudhvi Indana** 0.42146 1 Fri, 11 Dec 2015 22:16:53

We get accuracy of 42% on our test dataset which is highest we have scored.

4. CONCLUSION

Although our algorithms did not achieve top ranks in Kaggle tire, they performed fairly well and ensemble these algorithms might be done to improve the performance. Applying these algorithms gave us enough confidence to work on real data set implementing algorithms learnt in class.

Given some more time we could try to improve our mode by implementing bootstrapping, stacking and applying Kaggle top algorithms like XGboost.

LIST OF REFERENCES

- https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm
- http://www.cs.upc.edu/~bejar/apren/docum/trans/03d-algind-knn-eng.pdf
- http://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
- http://www.rdatamining.com/
- http://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/