Machine Learning : CIS 519

Rutuja Moharil

26 March 2019

Taking 1 Late day out of the 4 days

1 Probability decision boundary

Let the loss function be L(h(X),Y)
Which represents the loss if we have a false positive or false negative. The loss matrix that is given as :
$L(0,1) = 10$
$L(1,0) = 5$
$L(0,0) = 0$
$L(1,1) = 0$
The generalization error of the instance that we are trying to classify is given by :
$er[h] = E_x, y[L(Y, h(x))]$
Now lets assume that $\eta(x)$ is the conditional probability of the label being 1 given an instance x .
$\eta(x) = P(Y = 1|X = x)$
The total error can be given by :
$er[h] = E_X[\eta(x) * L(0,1) + (1 - \eta(x)) * L(1,0)]$
The bayes error which wold be associated by this distribution will be :
$er_D = argmin[10 * \eta(x), 5(1 - \eta(x))]$
Thus we can draw the following conclusions :

1. $h(x) = 1$
   $10\eta(x) > 5(1 - \eta(x))$
   $\eta(x) > \frac{1}{3}$

2. $h(x) = 0$ and $\eta(x) < \frac{1}{3}$
   We get the following classifier
   $h(x) = sign(\eta(x) - \frac{1}{3})$
   Here -1 would depict 0 so we get the decision boundary $\theta = \frac{1}{3}$

2 Double counting the evidence

Given an example of attribute values (x_1 ,x_2):Y = T , if

$$\frac{\log P(Y=T)}{\log P(Y=F)} + \frac{\log P(X_1=x_1|Y=T)}{\log P(X_1=x_1|Y=F)} + \frac{\log P(X_2=x_2|Y=T)}{\log P(X_2=x_2|Y=F)} > 0 \text{else} Y = F$$

1. If only X_1 is the attribute considered , then the error would be :

$$\begin{aligned} e_{X1} &= P(X=F,Y=T) + P(X_1=T,Y=F) \\ &= P(Y=T)P(X_1=F|Y=T) + P(X_1=T|Y=F) \\ &= 0.5*0.2 + 0.5*0.3 = 0.25 \\ Similarly, if only X_2 is considered, the error rate will be: \\ e_{X2} &= P(X=F,Y=T) + P(X_2=T,Y=F) \\ &= P(Y=T)P(X_2=F|Y=T) + P(Y=F)P(X_2=T|Y=F) \\ &= 0.5*0.5 + 0.5*0.1 = 0.3 \end{aligned}$$

2. The error when using both attributes X_1 and X_2

$$\begin{aligned} E &= P(X_1=T,X_2=T,Y=F) + P(X_1=T,X_2=F,Y=F) + \\ &P(X_1=F,X_2=T,Y=F) + P(X_1=F,X_2=F,Y=T) \\ &\text{Probabilities factorise to the form} P(X_1,X_2,Y) = P(Y)P(X_1|Y)P(X_2|Y) \\ &= 0.5*0.3*0.1 + 0.5*0.3*0.9 + 0.5*0.7*0.1 + 0.5*0.5*0.1 = 0.235 \end{aligned}$$

3. So in this case we create a new attribute X_3 copy of X_2. So Naive bayes considers attribute X_3 for which it assumes that it is conditionally independent of X_1 and X_2 given Y .
   So predictions will change and so double counting the evidence X_2 will make a difference here :

$$\begin{aligned} X_1 = T \ \& \ X_2 = T &\implies Y = T \\ X_1 = T \ \& \ X_2 = F &\implies Y = T \\ X_1 = F \ \& \ X_2 = T &\implies Y = F :: \\ &\text{This prediction has changed} \\ X_1 = F \ \& \ X_2 = F &\implies Y = F \end{aligned}$$

   Although we did not create any new information in the model by creating X_3 . So P(X_1, X_2, X_3, Y)= P(X_1, X_2, Y) as $X_3 = X_2$

Expected error is similar to the one we derived above only with different values of Y .

$$E = P(X_1 = T, X_2 = T, Y = F) + P(X_1 = T, X_2 = F, Y = F) +$$
$$P(X_1 = F, X_2 = T, Y = T) + P(X_1 = F, X_2 = F, Y = T)$$
Again factorising the probabilities to the form
$$P(X_1, X_2, Y) = P(Y)P(X_1|Y)P(X_2|Y)$$
$$= 0.3$$

Basically by duplicating the attribute we did not add any new information to the system as such . Hence there is no way that the performance would improve. Also the error cannot be zero or go to zero simply by duplicating the attribute X_2.

4. Here when we make a copy of X_2 (which is X_3), the naive baye over counts the evidence from attribute X_2 and the error increases.

5. Logistic regression does not use the same conditional independence and hence does not get affected by higly correlated features .So the coefficients obtained by training on X_1,X_2,X_3 would simply be just half of coefficient obtained by training on X_1, X_2 .

3 Reject Option

We have been given a loss matrix in this question and using this information we will have to make optimal choices .

1. Here we have been given :

$$P(y = 1|x) = 0.2$$
We can write the expected loss as
$$p(A|x) = 0.8L(0, a) + 0.2L(1, a)$$
If we choose the action a=0
$$p(a|x) = 0.2 * 10 = 2$$
If we choose the action a=1
$$p(a|x) = 0.8 * 10 = 8$$
If we choose the action a=rejection
$$p(a|x) = 0.8 * 3 + 0.2 * 3 = 3$$

Comparing the above values we can see that the best decision is to choose class 0 .

2. Here we have been given the following information:

$$P(y = 1|x) = 0.4$$
We can write the expected loss as
$$p(A|x) = 0.6L(0, a) + 0.4L(1, a)$$
If we choose the action a=0
$$p(a|x) = 0.4 * 10 = 4$$
If we choose the action a=1
$$p(a|x) = 0.6 * 10 = 6$$
If we choose the action a=rejection
$$p(a|x) = 0.8 * 3 + 0.2 * 3 = 3$$

Here the best decision is to choose rejection. This makes sense because $P(y = 0|x) = 1 - P(y = 1|x) = 0.6$ is not so high.

3. Here if we believe that class 1 or Class 0 is right we must choose the respective class else we choose the rejection. So at what particular value of θ threshold would correspond to the belief that the 1 or 0 class is right .

$$p(a|x) = P(y = 0|x)L(0, a) + P(y = 1|x)L(1, a)$$
If we choose $a = 0$
$$p(a|x) = 10p_1$$
If we choose $a = 1$
$$p(a|x) = 10(1 - p_1)$$
If we choose a= rejection
$$p(a|x) = 3p_1 + 3(1 - p_1) = 3$$
So for $a = 0$,the best choice should be
$$10p_1 < 10(1 - p_1) \implies p_1 \leq 0.5$$
$$10p_1 \leq 3 \implies p_1 \leq 0.3$$
So for $a = 1$,the best choice should be
$$10(1 - p_1) < 10(p_1) \implies p_1 \geq 0.5$$
$$10(1 - p_1) \leq 3 \implies p_1 \geq 0.7$$

Thus the thresholds we found are $\theta_0 = 0.3$ and $\theta_1 = 0.7$
In this complete question we tried to see which should be chosen among rejection ,0,1. The last part i.e the part 3 shows that we can choose a class only if it has probability

of more than 0.7 being true .

4. For the new loss matrix the thresholds can be found out by :

$$
\begin{aligned}
p(a|x) = P(y=0|x)L(0,a) + P(y=1|x)L(1,a) \\
\text{If we choose } a = 0 \\
p(a|x) = 10p_1 \\
\text{If we choose } a = 1 \\
p(a|x) = 5(1-p_1) \\
\text{If we choose a= rejection} \\
p(a|x) = 3p_1 + 3(1-p_1) = 3 \\
\text{So for } a = 0\text{,the best choice should be} \\
10p_1 < 5(1-p_1) \implies p_1 \leq 0.2 \\
10p_1 \leq 3 \implies p_1 \leq 0.3 \\
\text{So for } a = 1\text{,the best choice should be} \\
10(1-p_1) < 10(p_1) \implies p_1 \geq 0.5 \\
5(1-p_1) \leq 3 \implies p_1 \geq 0.4
\end{aligned}
$$

Thus the new thresholds are $\theta_0 = 0.3$ and $\theta_1 = 0.4$

4 Programming Part

4.1 Generalizing to unseen data

For this question which asks us to find ways to analyse and do extensive exploratory data analysis.
Observations about the data:

1. In this data set there are 81 columns or attributes,that may or not describe the dependency of virus infectation.

2. So on converting the csv file using pandas I noticed several NaN values in the datset. This means that there were several missing values .

3. Now according to some references on how to deal with missing values ,I found that this could be tackled by :

(a) Dropping missing values :
The criteria for dropping a certain column or attribute is if the missing values make upto more than 50% of the data.

```
US_citizen has 1 null values
Self_respond_sp has 1 null values
Interview_lang has 75 null values
Self_respond_fam has 75 null values
Self_respond2_fam has 75 null values
HH_country has 211 null values
HH_education has 210 null values
HH_marital has 78 null values
Income_household has 84 null values
Income_family has 77 null values
Income_ratio has 566 null values
Q1 has 70 null values
Q2 has 2736 null values
Q3 has 2736 null values
Q4 has 2736 null values
Q5 has 6997 null values
Q6 has 6997 null values
Q7 has 7805 null values
Q8 has 6997 null values
Q9 has 6997 null values
Q10 has 70 null values
Q11 has 5909 null values
Q12 has 73 null values
Q13 has 2736 null values
Q14 has 73 null values
Q15 has 1468 null values
Q16 has 70 null values
Q17 has 3517 null values
Q18 has 3519 null values
Q19 has 3519 null values
Q20 has 3519 null values
Q21 has 3519 null values
Q22 has 3520 null values
Q23 has 3228 null values
Q24 has 3520 null values
Q25 has 3521 null values
Q26 has 4901 null values
blood_pressure has 2133 null values
max_inflation_level has 1832 null values
pulse_status has 237 null values
abdomen_diameter has 2103 null values
weight_status has 5062 null values
weight has 65 null values
dentition_status has 335 null values
pregnancy_test has 6841 null values
fluoride_water_level has 4585 null values
fluoride_plasma_level has 6020 null values
urine_flow_rate has 2169 null values
virusB_infect_status has 5434 null values
virusC_infect_status has 6137 null values
virusD_infect_status has 4601 null values
glycohemoglobin has 2532 null values
last_alcohol has 7815 null values
recent_alcohol has 526 null values
hepatitis_A_test has 1246 null values
hepatitis_D_test has 1744 null values
hepatitis_B_test has 1741 null values
```

blood_pressure has 2133 null values
max_inflation_level has 1832 null values
pulse_status has 237 null values
abdomen_diameter has 2103 null values
weight_status has 5062 null values
weight has 65 null values
dentition_status has 335 null values
pregnancy_test has 6841 null values
fluoride_water_level has 4585 null values
fluoride_plasma_level has 6020 null value
urine_flow_rate has 2169 null values
virusB_infect_status has 5434 null values
virusC_infect_status has 6137 null values
virusD_infect_status has 4601 null values
glycohemoglobin has 2532 null values
last_alcohol has 7815 null values
recent_alcohol has 526 null values
hepatitis_A_test has 1246 null values
hepatitis_D_test has 1744 null values
hepatitis_B_test has 1741 null values
calcium_level has 2641 null values
glucose_level has 2607 null values
iron_level has 2626 null values
virusE_infect_status has 1744 null values

(b) Filling the missing values or Imputing
Generally imputing methods are done by filling the Nan or missing values with mean,mode or median of that particular attribute .
However in this case we have several categorical attributes (i,e Yes no questions , or ratings) such as the answers to the questionare with questions 1 to 27 , Virus C,D,E morbidity etc.
So imputing with mean or mode does not make sense here . In the answer options of categorical questions for this data set ,I noticed options like not sure or indeterminate, or can't say.
Instead we can replace the missing values with the answer options like Not sure ,indeterminate etc . This way we can tackle with categorical variables by filling in the values that do not significantly change our data .
Also the labels are more biased towards 0 .Basically there are more 0 values than 1 in the label data .

Observation : For this data set the columns of blood pressure,abdomen diameter,max inflation level, etc upto column for Q27 is empty for ages 0 to 5 .
This just shows that for ages 0 to 5 these data have not been recorded so it's a possibilty that it is not missing at random and instead missing for some reason.
Another observation I saw , the class labels are consistent,i.e for age 0 the label is 0. Similar observations for age 1,2,3,4,5 have the labels as 0 as well.
The strategy I used to fill in the values in these columns (corresponding to age 0 to 5) was to search the average ranges of normal values in these attributes and then generated the values randomly using random.randint within a range of values. This way we ensure that we do not assign simply constant values
.

Columns that were dropped

1. The columns patient id,Status,Unnamed:0,ReleaseCycle were removed or dropped because they are not needed for analysis.

2. Race1 : Race 1 is dropped because we already have 'Race2' column which specifies the information that is needed and so race 2 column is sufficient.

3. Q5,Q6,Q7,Q8: So the dataset given to us has information related to the person being interviewed. So the questions related to the children not eating or having improper food does not relate to how that person would be infected by the virus.

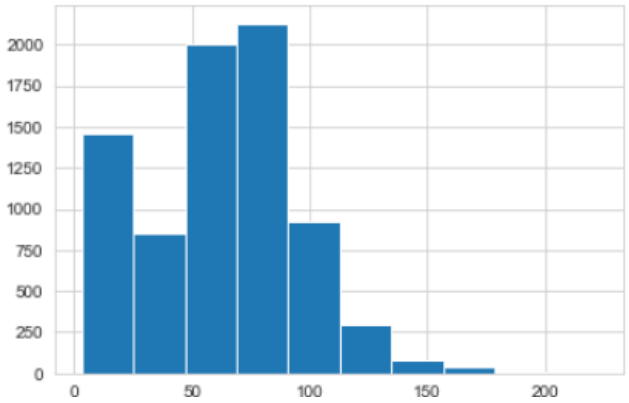
4. fluoride plasma level,fluoride water level,Q9,Q11,Q13,Q19: More than 50percent nanvalues that is why it is dropped.

5. Interviewlang,Selfrespondfam,Incomefamily, Selfrespond2fam,HHeducation,HHcountry, HHmarital,HHgender,Selfrespondsp,Selfrespond2sp.These values are not necessary and not indicative of virus infectation. HHage

Things that worked

1. So first I calculated the number of Nan in the columns, for categorical data we have been given an option of not sure . So I imputed the values for these categorical columns as not sure option .

2. For the numerical data like weight : I used the logic of substituing with the mean values. Following is the histogram that gives the distribution of weights from which we can see that it makes sense to replace the weight values with the mean which is equal to 62.537 .
   From the histogram below we see that common value of the weight is around the mean itself . So we can safely assume that this imputation method logic is right.

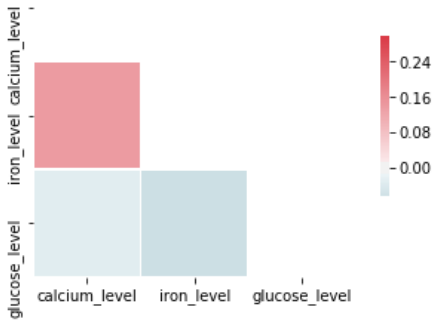


3. For numerical data where mean mode median could not be used , I imputed with a value of -10 . Which just suggests that the values were missing . This does not hinder the dataset or the prediction.

Things that did not work

1. As mentioned earlier Age 0 to 5 the values of BP, glycoheamoglobin,Iron,calcium were completely missing . Also the labels for these age is 0 . I thought that imputing the healthy values would give better accuracy results.
   However this approach did not really increase the accuracy. So this approach did not really turn out to be what I had expected.

2. Also there is another observation I made , at most places Iron ,calcium and glucose values were missing simultaneously i.e The three values were missing together .
   Initially,I thought that this was because they had some interdepency on each other and hence we could derive some co-relation between them . However I found the co-relation plot and from the plot it was observed that there wasn't any co-relation and the three values missing together was at random.



Classifiers
I used the following classifiers and noted an accuracy level which is tabulated as follows : 1) Decision trees 2) SVM 3)Adaboost 4) Bagging

Classifier	Accuracy
Random forest	0.8694
SVM(linear)	
SVM(rbf)	
Adaboost	0.8696
Bagging	0.8751
Decision tree	0.8443

As expected , ensemble learning methods give the highest accuracy . Here our best classifiers are Adaboost (DT-1) and Bagging.
As mentioned in the question every misclassification means that someone either dies or suffers pain during the unnecessary treatment. So it's very important to check for precision and recall values of the best classifiers .Because recall is the measure of total relevant results correctly classified by the algorithm, it is more important for us to since higher recall indicates lower false negatives.
Precision and recall for Adaboost 0.722, 0.732 and for Bagging 0.8484,0.8268. So we prefer a higher recall and thus would choose Adaboost over bagging in this case t reduce the cases of people identified as not having virus when in fact they have been infected with it .

4.2 Naive bayes

So in this case where we were supposed to implement Multinomial Naive bayes we get a test accuracy of and the values of precision and recall are 0.7053 and 0.6608 respectively.Following are the images of top 10 or most frequent 10 words for the training and test set for first 10 authors.

Author 1 : Train data :

really just like subaru ve bunny way onions short old	

Test data :

like just really time good know think people day things	

Author 2 : Train

just really im like dont good people time things want	
like just really time good know think people day things	

Author3 : Train

just really like people im time dont good want year	

Test :

like just really time good know think people day things	

Author 4 : Train

just people quixtar time business really like good ll know	

Test :

like just really time good know think people day things	

Author 5 : Train

just people like time good quixtar really business know urllink	

Test:

like just really time good know think people day things	

Author 6:

just people like really time good quixtar business urllink know	

just people like really time good quixtar business urllink know	

Author 7 :

just like people really time good quixtar know got business	
like just really time good know think people day things	

Author 8 :

just like people time really good know think quixtar want	
like just really time good know think people day things	

Author 9 :

just like people time really good know think quixtar want	
Like just really time good know think people day things	

Author 10 :

like just really time people good know love want think	
like just really time good know think people day things	

We were asked to experiment with the parameters: stop words,ngram , max features. Following is the tabulated values of the accuracy corresponding to the parameter changes.

Parameters	Accuracy
stop_words='english',max_features=4000 ,ngram_range=(1,1))	0.6781
stop_words='english',max_features=6000 ,ngram_range=(1,2))	0.66089
stop_words='english',max_features=4000 ,ngram_range=(1,2))	0.65217
stop_words='english',max_features=4000 ,ngram_range=(2,2))	0.4433
stop_words='english',max_features=6000 ,ngram_range=(2,2))	0.4782
stop_words=values, max_features=6 000, ngram_range=(2,2))	0.5565
stop_words=values, max_features=6000,n gram_range=(1,2))	0.64347
stop_words=values, max_features=4000,n gram_range=(1,2))	0.62608
stop_words=values,max_features=4000,ng ram_range=(1,2)) values =['the','an','in','of','a','to']	0.60869

Observations: Here we use multinomial Naive Bayes classifier since it is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts.
Vectorizer tuning :

1. Stop words :
   Removing common and uncommon words is extremely useful in text calssification. Since it is obvious that common words like 'the', 'a', or 'and' are commonly used in English language formation so these words don't really help in giving out helpful information regarding the type or similarity of the documents in a sense, they carry less semantic weight.
   Whereas,there may be words that only appear one of twice in the entire corpus and we simply don't have enough data about these words to learn a meaningful output.Thus giving stopwords = english gives a higher accuracy.

2. Max features:
   This parameter chooses the maximum features upto which the most frequent words are calculated.The CountVectorizer will choose the words/features = number of max features that occur most frequently to be in its' vocabulary and drop everything else. The more the max features the more number of words it compares and could prove similarity of documents better.

3. ngram range :
   An ngram represents a string of n words in a particular row. E.g. the sentence 'This is Homework' contains the 2 grams which are 'This is ' and 'is Homework'.There are 3 grams in this sentence overall. Setting the parameter ngramrange=(a,b) where a,b represent the minimum and maximum ngram size respectively.
   I found that including Inclusion of 2grams as features did not really boost the performance as I had expected.Maybe because these 2 gramm occurences aren't frequent enough.
   For example during search we type things like 'data analyst' or 'machine learning job' so in this case a 2 gram would be 'machine learning'(which is likely to get more hits) and 'learning job'. So ngram range (1,2) might work in those cases but definitely not in our text documents.

TF IDF observations :
In this portion we have been asked to experiment with same three parameters as above. However TF-IDF is slightly different than the countvectorizer function.

Parameter	SVM	SVM:RBF	COSINE	GNB	MLG
stop_words= 'english',max_features=1000,ngram_range=(1,1))	0.686	0.65	0.686	0.408	0.664
stop_words= 'english',max_features=4000,ngram_range=(1,1))	0.7478	0.713	0.7478	0.530	0.756
stop_words= 'english',max_features=1000,ngram_range=(1,2))	0.6695	0.652	0.6695	0.434	0.660
stop_words= 'english',max_features=4000,ngram_range=(1,2))	0.7478	0.721	0.7478	0.504	0.756
stop_words= None, max_features=4000,ngram_range=(1,2))	0.7565	0.686	0.7565	0.434	0.765
stop_words= None, max_features=1000,ngram_range=(2,2))	0.573	0.567	0.573	0.312	0.608

Conclusions drawn for classifiers ,following are the results of the accuracies obtained

1. A Naive Bayes model treats the features as independent, whereas SVM looks at the interactions between them to a certain degree for non-linear kernels(like gaussian or rbf). In this case SVM(rbf) performs better than guassian naive bayes (gnb) model. Since the TF-IDF method gives us the words that are important or not , it can be said that linear SVM can predict with higher accuracy due to the linear nature of the data .

2. We can also notice that SVM rbf performs worst among SVM linear and cosine similarity . Since it is a high dimensional data and hidh dimensional data is generally linearly separable.This is because if a data is linearly separable then mapping to a higher dimension and using SVM rbf doesn't perform better than linear SVM.

3. Looking at multinomial logistic regression we notice that this is the best classifier for text classification in this case.Since for multinomial regression , there is no need for the independent variables to be statistically independent from each other unlike naive bayes classifier.

4. SVM linear and Multinomial logistic regression are pretty close in terms of accuracies.Multi nomial regression is slightly better because it cosiders the correlation between features unlike SVM.

TF IDF in short zeroes out the word that would occur in every single document (TF) while assigning a larger count to the word that appears in very few documents (IDF term).
With the TFIDF Vectorizer the value increases proportionally to count, but is offset by the frequency of the word in the corpus the IDF (inverse document frequency part).
This helps to adjust for the fact that some words appear more frequently.
However Countvectorizer just simply gives out the frequency of the words occurring in the text.

5 LDA

LDA is an unsupervised clustering algorithm.It is an example of topic model and is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.
The steps to be taken for data pre-processing are :

1. Tokenization: Splitting the text into sentences and then sentences into words. Lowercasing the words and removing punctuation is done then.

2. Remove all stop words

3. Lemmatization and stemming : Words like includes ,including, included are reduced to their roots i.e include . So we remove -ing,-ed etc .words in third person are changed to first person and verbs in past and future tenses are changed into present

After the pre-processing , I used the Countvetcorizer and thus created the feature space and passed it to the sklearn.decomposition package.

Author	Topic 1	Topic 2
1	0.510	0.462
2	0.015	0.964
3	0.200	0.782
4	0.951	0.024
5	0.134	0.801
6	0.420	0.529
7	0.440	0.514
8	0.217	0.742
9	0.510	0.429
10	0.410	0.567

Author	Topic 1	Topic 2	Topic 3
1	0.498	0.410	0.090
2	0.007	0.964	0.007
3	0.022	0.600	0.366
4	0.965	0.002	0.002
5	0.120	0.421	0.430
6	0.411	0.342	0.254
7	0.034	0.372	0.500
8	0.200	0.741	0.010
9	0.514	0.400	0.060
10	0.400	0.200	0.392

The most frequent words in the three topics for every author are :

Author 1	Topic 1: like trying weekend room profile thing mannequin bunch computer plastic Topic 2: plastic room bunch computer thing mannequin trying profile week end like Topic 3: bunch computer weekend thing profile plastic trying like manneq uin room
Author2	Topic 1: eh ah stay absolutely friend fun depressed guess wonderful guys Topic 2: just really want mom day things like feel oh night Topic 3: jenn got eh guys absolutely home really piano actually minutes
Author3	Topic 1: ideas cottage actually laughter got lesson long looking friends ve Topic 2: nt need time ve ll like having really goes talked Topic 3: driving going haha nice test canada just really lesson g2

Author 4	Topic 1: business money urllink bww ve people know make tapes upline Topic 2: tm money retail income profit net ftc legal people product Topic 3: doing ibo rich wo read taught person seminars wrong money
Author 5	Topic 1: best going beta ago hl2 doom hours like try knows Topic 2: nt game people like lot really did kinda doom think Topic 3: nt physics cs dad seams game times reads nice say
Author 6	Topic 1: license days yay going signing right Topic 2: right signing going days license yay Topic 3: signing yay license going right days
Author 7	Topic 1: lol bed coming ohmigosh huge old summer somebody like gon Topic 2: totally somebody really shoot yeah ohmigosh hey kinda bored lik e Topic 3: day like summer hate huge bed yay yeah coming washington
Author 8	Topic 1: brain things attention practice better best remembering teach z emansky ones Topic 2: nt people help suicide life group young ca join need Topic 3: addiction think tuesday remember nt traumatic make suffer cigar ette feel
Author 9	Topic 1: really nt love eric hate money man Topic 2: money love man really hate eric nt Topic 3: eric love man really money hate nt
Author 10	Topic 1: got tourney cool home nt today game cities hey fun Topic 2: school summer game did watched love got nt home hung Topic 3: just games cities summer weekend tourney busy hey later love

Collaborators :
Gauri Pradhan
G.Sai
Siddharth Singh
Tien Pham