Sivaranjani Prabasankar A20436206

Homework 2

1. (20 points) Classification: We are going to make a decision about whether an animal is useful (P) or useless (N) in our experiments. We measure their age in days, whether fat or not, and the size of their soles of the feet.

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	3.2	Yes	100	P
4.6	4.2	1.4	0.2	Yes	40	N
4.3	5.0	1.5	4.2	No	160	N
5	1.3	1.4	2.2	Yes	90	P
5	1.2	4.7	1.4	No	40	N
4.4	3.2	4.5	1.5	No	80	P
4.9	3.1	4.9	1.5	No	100	P
2.5	1.3	4	1.3	Yes	110	P
4.5	2.8	4.6	1.5	Yes	120	P
4.3	3.3	4.9	2.5	Yes	30	N
1.8	2.7	5.0	1.9	Yes	20	N
2.1	3	5.0	2.1	No	40	N
4.3	2.9	5.0	1.8	No	30	N
4.5	3	4.9	2.2	No	50	N
4.3	**3.6**	**1.5**	**1.8**	**Yes**	**70**	**?**

a). [5 points] Do we need normalization and discretization (data type transformation) to use KNN classifier? Why (use your own text/description)?

Yes, we need normalization as range of Age (in days) is between 20-160 whereas for other attributes its around 1 – 5. These difference in attribute scale might affect the results. Hence, we require normalization for Age attribute.

Similarly, we need discretization for Fat column. KNN use distance measure to find the label. Hence all attributes should be in numerical except the label. Fat is a categorical attribute, so we need transformation for it.

b). [5 points] If your answer is Yes in part 1), please apply normalization (to new scale [1,5]) and discretization.

Original Data:

Sivaranjani Prabasankar A20436206

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	3.2	Yes	100	P
4.6	4.2	1.4	0.2	Yes	40	N
4.3	5.0	1.5	4.2	No	160	N
5	1.3	1.4	2.2	Yes	90	P
5	1.2	4.7	1.4	No	40	N
4.4	3.2	4.5	1.5	No	80	P
4.9	3.1	4.9	1.5	No	100	P
2.5	1.3	4	1.3	Yes	110	P
4.5	2.8	4.6	1.5	Yes	120	P
4.3	3.3	4.9	2.5	Yes	30	N
1.8	2.7	5.0	1.9	Yes	20	N
2.1	3	5.0	2.1	No	40	N
4.3	2.9	5.0	1.8	No	30	N
4.5	3	4.9	2.2	No	50	N
4.3	**3.6**	**1.5**	**1.8**	**Yes**	**70**	**?**

Discretized Data:

Attribute: Fat Values: Yes → 1, No → 0

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	3.2	1	100	P
4.6	4.2	1.4	0.2	1	40	N
4.3	5.0	1.5	4.2	0	160	N
5	1.3	1.4	2.2	1	90	P
5	1.2	4.7	1.4	0	40	N
4.4	3.2	4.5	1.5	0	80	P
4.9	3.1	4.9	1.5	0	100	P
2.5	1.3	4	1.3	1	110	P
4.5	2.8	4.6	1.5	1	120	P
4.3	3.3	4.9	2.5	1	30	N
1.8	2.7	5.0	1.9	1	20	N
2.1	3	5.0	2.1	0	40	N
4.3	2.9	5.0	1.8	0	30	N
4.5	3	4.9	2.2	0	50	N
4.3	**3.6**	**1.5**	**1.8**	**1**	**70**	**?**

Normalization of Data (Age): Min Max Normalization

New Min: 1, New Max: 5

Old Min: 20, Old Max: 160

New value = {(Old value – Old min) * (New Max – New Min) / (Old Max – Old Min)} + (New Min)

= {(Old value –20) * (5– 1) / (160– 20)} + (1)

= {(Old value –20) * (4) / (140)} + (1) = {(Old value –20) / (35)} + (1)

Sivaranjani Prabasankar A20436206

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	3.2	1	= {(100–20) / (35)} + (1)	P
4.6	4.2	1.4	0.2	1	= {(40–20) / (35)} + (1)	N
4.3	5.0	1.5	4.2	0	= {(160–20) / (35)} + (1)	N
5	1.3	1.4	2.2	1	= {(90–20) / (35)} + (1)	P
5	1.2	4.7	1.4	0	= {(40–20) / (35)} + (1)	N
4.4	3.2	4.5	1.5	0	= {(80–20) / (35)} + (1)	P
4.9	3.1	4.9	1.5	0	= {(100–20) / (35)} + (1)	P
2.5	1.3	4	1.3	1	= {(110–20) / (35)} + (1)	P
4.5	2.8	4.6	1.5	1	= {(120–20) / (35)} + (1)	P
4.3	3.3	4.9	2.5	1	= {(30–20) / (35)} + (1)	N
1.8	2.7	5.0	1.9	1	= {(20–20) / (35)} + (1)	N
2.1	3	5.0	2.1	0	= {(40–20) / (35)} + (1)	N
4.3	2.9	5.0	1.8	0	= {(30–20) / (35)} + (1)	N
4.5	3	4.9	2.2	0	= {(50–20) / (35)} + (1)	N
4.3	**3.6**	**1.5**	**1.8**	**1**	**= {(70–20) / (35)} + (1)**	**?**

Normalization of Data (Right Front): Min Max Normalization

New Min: 1, New Max: 5

Old Min: 0.2, Old Max: 4.2

New value = {(Old value – Old min) * (New Max – New Min) / (Old Max – Old Min)} + (New Min)

= {(Old value –0.2) * (5– 1) / (4.2– 0.2)} + (1)

= {(Old value –0.2) * (4) / (4)} + (1) = {(Old value –0.2)} + (1)

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	= {(3.2–0.2)} + (1)	1	3.29	P
4.6	4.2	1.4	= {(0.2-0.2)} + (1)	1	1.57	N
4.3	5.0	1.5	= {(4.2-0.2)} + (1)	0	5	N
5	1.3	1.4	= {(2.2-0.2)} + (1)	1	3	P
5	1.2	4.7	= {(1.4-0.2)} + (1)	0	1.57	N
4.4	3.2	4.5	= {(1.5-0.2)} + (1)	0	2.71	P
4.9	3.1	4.9	= {(1.5-0.2)} + (1)	0	3.29	P
2.5	1.3	4	= {(1.3-0.2)} + (1)	1	3.57	P
4.5	2.8	4.6	= {(1.5-0.2)} + (1)	1	3.86	P
4.3	3.3	4.9	= {(2.5-0.2)} + (1)	1	1.29	N
1.8	2.7	5.0	= {(1.9-0.2)} + (1)	1	1	N
2.1	3	5.0	= {(2.1-0.2)} + (1)	0	1.57	N
4.3	2.9	5.0	= {(1.8-0.2)} + (1)	0	1.29	N
4.5	3	4.9	= {(2.2-0.2)} + (1)	0	1.86	N
4.3	**3.6**	**1.5**	**= {(1.8-0.2)} + (1)**	**1**	**2.43**	**?**

Sivaranjani Prabasankar A20436206

Normalization of Data (Fat): **Min Max Normalization**

New Min: 1, New Max: 5

Old Min: 0, Old Max: 1

New value = {(Old value – Old min) * (New Max – New Min) / (Old Max – Old Min)} + (New Min)

= {(Old value –0) * (5– 1) / (1– 0)} + (1)

= {(Old value) * (4) / (1)} + (1) = {Old value * 4} + (1)

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	4	5	3.29	P
4.6	4.2	1.4	1	5	1.57	N
4.3	5.0	1.5	5	1	5	N
5	1.3	1.4	3	5	3	P
5	1.2	4.7	2.2	1	1.57	N
4.4	3.2	4.5	2.3	1	2.71	P
4.9	3.1	4.9	2.3	1	3.29	P
2.5	1.3	4	2.1	5	3.57	P
4.5	2.8	4.6	2.3	5	3.86	P
4.3	3.3	4.9	3.3	5	1.29	N
1.8	2.7	5.0	2.7	5	1	N
2.1	3	5.0	2.9	1	1.57	N
4.3	2.9	5.0	2.6	1	1.29	N
4.5	3	4.9	3	1	1.86	N
4.3	**3.6**	**1.5**	**2.6**	**5**	**2.43**	**?**

Normalized and Discretized Data:

Left Back	Right Back	Left Front	Right Front	Fat	Age	Label
4.1	4.8	1.6	4	5	3.29	P
4.6	4.2	1.4	1	5	1.57	N
4.3	5.0	1.5	5	1	5	N
5	1.3	1.4	3	5	3	P
5	1.2	4.7	2.2	1	1.57	N
4.4	3.2	4.5	2.3	1	2.71	P
4.9	3.1	4.9	2.3	1	3.29	P
2.5	1.3	4	2.1	5	3.57	P
4.5	2.8	4.6	2.3	5	3.86	P
4.3	3.3	4.9	3.3	5	1.29	N
1.8	2.7	5.0	2.7	5	1	N
2.1	3	5.0	2.9	1	1.57	N
4.3	2.9	5.0	2.6	1	1.29	N
4.5	3	4.9	3	1	1.86	N
4.3	**3.6**	**1.5**	**2.6**	**5**	**2.43**	**?**

Sivaranjani Prabasankar A20436206

c). [10 points] Apply KNN Classifier to the new data table in part b). In other words, build your KNN classifier by the following requirements based on the knowledge in the table, and then predict which class/label the object (in red) belongs to:

- Distance measures: Manhattan distance
- K = 1, 3, 5

Manhattan Distance – KNN

Distance = | x1 – y1 | + | x2 – y2 | + | x3 – y3 | + | x4 – y4 | + | x5 – y5 | + | x6 – y6 |

No	Left Back	Right Back	Left Front	Right Front	Fat	Age	Label	Manhattan Distance	
1	4.1	4.8	1.6	4	5	3.29	P	0.2 + 1.2 + 0.1 + 1.4 + 0 + 0.86	3.76
2	4.6	4.2	1.4	1	5	1.57	N	0.3 + 0.6 + 0.1 + 1.6 + 0 + 0.86	3.46
3	4.3	5.0	1.5	5	1	5	N	0 + 1.4 + 0 + 2.4 + 4 + 2.57	10.37
4	5	1.3	1.4	3	5	3	P	0.7 + 2.3 + 0.1 + 0.4 + 0 + 0.57	4.07
5	5	1.2	4.7	2.2	1	1.57	N	0.7 + 2.3 + 3.2 + 0.4 + 4 + 0.86	11.46
6	4.4	3.2	4.5	2.3	1	2.71	P	0.1 + 0.4 + 3 + 0.3 + 4 + 0.28	8.08
7	4.9	3.1	4.9	2.3	1	3.29	P	0.6 + 0.5 + 3.4 + 0.3 + 4 + 0.86	9.66
8	2.5	1.3	4	2.1	5	3.57	P	1.8 + 1.3 + 2.5 + 0.5 + 0 + 1.14	7.24
9	4.5	2.8	4.6	2.3	5	3.86	P	0.2 + 0.8 + 3.1 + 0.3 + 0 + 1.43	5.83
10	4.3	3.3	4.9	3.3	5	1.29	N	0 + 0.3 + 3.4 + 0.7 + 0 + 1.14	5.54
11	1.8	2.7	5.0	2.7	5	1	N	2.5 + 2.9 + 3.5 + 0.1 + 0 + 1.43	10.43
12	2.1	3	5.0	2.9	1	1.57	N	2.2 + 0.6 + 3.5 + 0.3 + 4 + 0.86	11.46
13	4.3	2.9	5.0	2.6	1	1.29	N	0 + 0.7 + 3.5 + 0 + 4 + 1.14	9.34
14	4.5	3	4.9	3	1	1.86	N	0.2 + 0.6 + 3.6 + 0.4 + 4 + 0.57	9.37
15	**4.3**	**3.6**	**1.5**	**2.6**	**5**	**2.43**	**?**		

Solution

K = 1 → {R2} → {N}

When K = 1, using KNN – Manhattan distance measures the predicted label is N. **(i.e. The animal is not useful).**

K = 3 ➔ {R2, R1, R4} ➔ {N, P, P}
When K = 3, using KNN – Manhattan distance measures the predicted label is P. **(i.e. The animal is useful).**

K = 5 ➔ {R2, R1, R4, R9, R10} ➔ {N, P, P, P, N}
When K = 5, using KNN – Manhattan distance measures the predicted label is P. **(i.e. The animal is useful).**

2. (40 points) Use Naïve Bayes Classifier to classify the objects

We conducted a survey to collect people's daily diets and try to build a model to predict whether their diets result in healthy conditions or not. The final results could be Yes, No, Unsure

Breakfast	**Lunch**	**Dinner**	**Healthy?**
Ham	Carnivorous	Beef	Y
Milk	Carnivorous	Beef	N
Bread	Veggie	Pork	U
Bread	Veggie	Veggie	Y
Ham	Veggie	Veggie	Y
Bread	Carnivorous	Beef	N
Ham	Veggie	Pork	N
Milk	Veggie	Pork	U
Milk	Carnivorous	Veggie	U
Noddle	Carnivorous	Pork	?

1). [10 points] What is laplace smoothing? And why we need it in the Naïve Bayesian classifier?

Sometimes, categorical variable has a category in test data set, which was not trained in train data set. i.e. There are no examples contains the attribute value mentioned in test data. In this case, model will assign a 0 (zero) probability and will be unable to make a prediction.

Laplace smoothing is a solution, to smooth categorical data. A small-sample correction will be incorporated in every probability estimate. Consequently, no probability will be zero. This way is called as Laplace smoothing for regularizing Naive Bayes.

2). [20 points] Using the Naive Bayesian Classification Hint: you may need to use laplace smoothing if you do have zero-conditional probabilities. Use the setting in the slide to solve the problems in this case. Note, only apply laplace smoothing to the ones you have zero-conditional probabilities.

C1: Healthy = Yes,	C2: Healthy = No,	C3: Healthy = Unsure
E1: Breakfast= Noodle,	E2: Lunch = Carnivorous,	E3: Dinner = Pork
P(C1) = 3/9 = 0.33,	P(C2) =3/9 = 0.33,	P(C3) = 3/9 = 0.33
P(E1/C1) = 0/3 = 0,	P(E1/C2) = 0/3,	P(E1/C3) = 0/3

Using Laplace Smoothing for P(E1/C1) as its 0,

P (E1/C1) = P (Breakfast= Noodle / Healthy = Yes) = (nc+ m*p) / (n + m)

m = 9, nc = 0, n = 3, p = 1/t where t = 4, Hence p = 0.33

P (Breakfast= Noodle / Healthy = Yes) = (0+ 9*0.25) / (4 + 9) = 2.25/12 = 0.1875,

P (E1/C1) = 0.1875

Using Laplace Smoothing for P(E1/C2) as its 0,

P (E1/C2) = P (Breakfast= Noodle / Healthy = No) = (nc+ m*p) / (n + m)

m = 9, nc = 0, n = 3, p = 1/t where t = 4, Hence p = 0.33

P (Breakfast= Noodle / Healthy = No) = (0+ 9*0.25) / (4 + 9) = 2.25/12 =0.1875

P (E1/C2) = 0.1875

Using Laplace Smoothing for P(E1/C3) as its 0,

P (E1/C3) = P (Breakfast= Noodle / Healthy = Unsure) = (nc+ m*p) / (n + m)

m = 9, nc = 0, n = 3, p = 1/t where t = 4, Hence p = 0.33

P (Breakfast= Noodle / Healthy = Unsure) = (0+ 9*0.25) / (4 + 9) = 2.25/12 = 0.1875

P (E1/C3) = 0.1875

P(E2/C1) = 1/3,	P(E2/C2) = 2/3,	P(E2/C3) = 1/3
P(E3/C1) = 0/3,	P(E3/C2) = 1/3,	P(E3/C3) = 2/3

Using Laplace Smoothing for P(E3/C1) as its 0,

P (E3/C1) = P (Dinner = Pork / Healthy = Yes) = (nc+ m*p) / (n + m),

m = 9, nc = 0, n = 3, p = 1/t where t = 3, Hence p = 0.33

P (Dinner = Pork / Healthy = Yes) = (0+ 9*0.33) / (3 + 9) = 1.65/8 = 0.2475

P (E3/C1) =0.20625

P(C1) = 0.33	P(C2) = 0.33	P(C3) = 0.33
P (E1/C1) = 0.1875	P (E1/C2) = 0.1875	P (E1/C3) = 0.1875
P(E2/C1) = 0.33	P(E2/C2) = 0.66	P(E2/C3) = 0.33
P (E3/C1) = 0.2475	P(E3/C2) = 0.33	P(E3/C3) = 0.66

$$\mathbf{P\ (E \mid C1) = \prod^{m}{}_{j=1} P\ (E_j/C1)}$$

P (E | C1) = 0.1875* 0.33 * 0.2475 = 0.015

P (E | C2) = 0.1875* 0.66 * 0.33 = 0.041

P (E | C3) = 0.1875* 0.33 * 0.66 = 0.041

P(E) = P(C1) * P(E/C1) + P(C2) * P(E/C2) + P(C3) * P(E/C3)

= 0.33*0.015 +0.33* 0.041 + 0.33*0.041 = 0.03201

P (C1 | E) = (P(C1) * P (E | C1)) / P(E) = 0.33 * 0.015 / 0.03201 = 0.00495 / 0.03201 = 0.1546

P (C2 | E) = (P(C2) * P (E | C2)) / P(E) = 0.33 *0.041 / 0.03201 = 0.01353 / 0.03201 = 0.4223

P (C3 | E) = (P(C3) * P (E | C3)) / P(E) = 0.33 * 0.041 / 0.03201 = 0.01353 / 0.03201 = 0.4223

P (C1 | E) < P (C2 | E) = P (C3 | E)

Assuming the features are independent the Probability of getting Healthy as NO and UNSURE are higher than Yes. But Probability of getting Healthy as NO and UNSURE are equal. The reason for this ambiguity is due to less training data. This requires further analysis/ training to predict the label. Therefore the 10th row can be predicted as **UNSURE**.

3). [10 points] List the characteristics, and the advantages and disadvantages of the Naïve Bayes classification method?

Advantages

1) Easy to implement
2) Requires less training data

Disadvantages

1) Violation of Independent assumption

   In Naive Bayes there is an assumption that predictors are independent. Sometimes, the data we get might not be completely independent.

2) Zero Conditional probability

   Sometimes, categorical variable has a category in test data set, which was not observed in training data set. In this case, model will assign a 0 (zero) probability and will be unable to make a prediction.

3) Multi-Collinearity

   There could be correlation between attributes used to train data. In Naïve Bayes, there are no ways to find the correlation between features.

4) Handling Numeric or continuous features

   We may need to bin data and convert Quantitative to Categorical features carefully without losing any information.

5) Imbalanced classes

   It may result in skewed probabilities.

3. (9 points) Consider the following three short documents:

Doc #1

Glimpse is an indexing and query system that allows for search through a file system or document collection quickly. Glimpse is the default search engine in a larger information retrieval system. It has also been used as part of some web based search engines.

Doc #2

The main processes in an retrieval system are document indexing, query processing, query evaluation and relevance feedback. Among these, efficient updating of the index is critical in large scale systems.

Doc #3

Clusters are created from short snippets of documents retrieved by web search engines which are as good as clusters created from the full text of web documents.

First remove stop words, and punctuation, and apply Porter's stemming algorithm to the three documents (Note: You can use the online stemming tools below for this purpose). List the final UNIQUE tokens/terms in the table below (make sure terms are listed row byrow)

Online Stemming Tool: http://9ol.es/porter_js_demo.html
Stop words list: http://www.ranks.nl/stopwords

Sivaranjani Prabasankar A20436206

Process	Doc #1	Doc #2	Doc #3
Original Doc	Glimpse is an indexing and query system that allows for search through a file system or document collection quickly. Glimpse is the default search engine in a larger information retrieval system. It has also been used as part of some web based search engines.	The main processes in an retrieval system are document indexing, query processing, query evaluation and relevance feedback. Among these, efficient updating of the index is critical in large scale systems.	Clusters are created from short snippets of documents retrieved by web search engines which are as good as clusters created from the full text of web documents.
After removing stop words	Glimpse indexing query system that allows search through file system document collection quickly Glimpse default search engine larger information retrieval system also used part some web based search engines	main processes retrieval system document indexing query processing query evaluation relevance feedback Among these efficient updating index critical large scale systems	Clusters created short snippets documents retrieved web search engines which good clusters created full text web documents.
Stemmed Doc	Glimps index queri system that allow search through file system document collect quickli Glimps default search engin larger inform retriev system also us part some web base search engin **Total No. of Words: 29**	main process retriev system document index queri process queri evalu relev feedback Among these effici updat index critic larg scale system **Total No. of Words: 21**	Cluster creat short snippet document retriev web search engin which good cluster creat full text web document **Total No. of Words: 17**

Sivaranjani Prabasankar A20436206

Removing duplicate words (UNIQUE Tokens)	Glimps index queri system that allow search through file document collect quickli default engin larger inform retriev also us part some web base **Total No. of Words: 23**	main process retriev system document index queri evalu relev feedback Among these effici updat critic larg scale **Total No. of Words: 17**	Cluster creat short snippet document retriev web search engin which good full text **Total No. of Words: 13**

4. (31 points) Use Normalized TF.IDF weighting to process the following documents

	Term1	Term2	Term3	Term4	Term5	Term6	Term7	Term8
DOC1	0	3	1	0	0	2	1	0
DOC2	5	0	0	0	3	0	0	2
DOC3	3	0	4	3	4	0	0	5
DOC4	1	8	0	3	0	1	4	0
DOC5	0	1	0	0	0	5	4	2
DOC6	2	0	2	0	0	4	0	1
DOC7	2	5	0	3	0	1	4	2
DOC8	3	3	0	2	0	0	1	3
DOC9	0	0	3	3	3	0	0	0
DOC10	1	0	5	0	2	4	0	2

1). [16 points] You should produce a new data table similar to the table above, but fill in the table with your normalized TF.IDF weights. Note: you can complete the calculations by hand or using Excel; or, you can complete the calculations by Excel; or, you can complete the calculations by compiling your own programs; You do not need to submit the Excel or program coding, the only thing you need to present is the new doc-term table with normalized TF-IDF weights

Sivaranjani Prabasankar A20436206

$$W_{ik} = tf_{ik}. Log_2(N/n_k)$$

$$W_{ik} = tf_{ik}. Log_2(N/n_k) / \sqrt{\Sigma^{t}_{k=1}} (tf_{ik})^2 [Log_2(N/n_k)]^2$$

```
        Term1  Term2  Term3  Term4  Term5  Term6  Term7  Term8
     ----------------------------------------------
DOC1     0      3      1      0      0      2      1      0
DOC2     5      0      0      0      3      0      0      2
DOC3     3      0      4      3      4      0      0      5
DOC4     1      8      0      3      0      1      4      0
DOC5     0      1      0      0      0      5      4      2
DOC6     2      0      2      0      0      4      0      1
DOC7     2      5      0      3      0      1      4      2
DOC8     3      3      0      2      0      0      1      3
DOC9     0      0      3      3      3      0      0      0
DOC10    1      0      5      0      2      4      0      2
      ---------------------------------------------
```

Term Frequency (TF)								
	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
Doc 1	0	0.90309	0.30103	0	0	0.443697	0.30103	0
Doc 2	0.77451	0	0	0	1.19382	0	0	0.309804
Doc 3	0.464706	0	1.20412	0.90309	1.59176	0	0	0.77451
Doc 4	0.154902	2.40824	0	0.90309	0	0.221849	1.20412	0
Doc 5	0	0.30103	0	0	0	1.109244	1.20412	0.309804
Doc 6	0.309804	0	0.60206	0	0	0.887395	0	0.154902
Doc 7	0.309804	1.50515	0	0.90309	0	0.221849	1.20412	0.309804
Doc 8	0.464706	0.90309	0	0.60206	0	0	0.30103	0.464706
Doc 9	0	0	0.90309	0.90309	1.19382	0	0	0
Doc 10	0.154902	0	1.50515	0	0.79588	0.887395	0	0.309804

Inverse Document Frequency (IDF)	
Doc 1	1.092555313
Doc 2	1.456382557
Doc 3	2.36959774
Doc 4	2.852771928
Doc 5	1.693199384
Doc 6	1.126906918
Doc 7	2.184519792
Doc 8	1.304059019
Doc 9	1.748241775
Doc 10	1.950985502

Sivaranjani Prabasankar A20436206

Term Weighing (TF * IDF)								
	Term 1	**Term 2**	**Term 3**	**Term 4**	**Term 5**	**Term 6**	**Term 7**	**Term 8**
Doc 1	0	0.826585	0.275528	0	0	0.40611	0.275528	0
Doc 2	0.531804	0	0	0	0.819716	0	0	0.212722
Doc 3	0.196112	0	0.508154	0.381115	0.671743	0	0	0.326853
Doc 4	0.054299	0.844175	0	0.316566	0	0.077766	0.422088	0
Doc 5	0	0.177788	0	0	0	0.655117	0.711151	0.18297
Doc 6	0.274915	0	0.534259	0	0	0.787461	0	0.137458
Doc 7	0.141818	0.689007	0	0.413404	0	0.101555	0.551206	0.141818
Doc 8	0.356353	0.692522	0	0.461682	0	0	0.230841	0.356353
Doc 9	0	0	0.51657	0.51657	0.682869	0	0	0
Doc 10	0.079397	0	0.771482	0	0.407937	0.454844	0	0.158794

2). [15 points] Assume we have a query, and we already know the relevant documents to this query are: DOC1, DOC6, DOC9. Our IR system produced a top-10 list of retrieved results as: DOC1, DOC4, DOC3, DOC6, DOC5, DOC9, DOC2, DOC8, DOC7, DOC10. We are going to return the top-N relevant documents, let's choose N = 1, 3, 5. Calculate the precision and recall values at N = 1, 3 and 5. Note: you should show the confusion matrix which includes tp, fp, tn, fn for N = 1, 3, 5

	N = 1	**N = 3**	**N = 5**
Documents Required	DOC1, DOC6, DOC9		
Documents Retrieved	DOC1	DOC1, DOC4, DOC3	DOC1, DOC4, DOC3, DOC6, DOC5
Doc 1	True Positive	True Positive	True Positive
Doc 2	True Negative	True Negative	True Negative
Doc 3	True Negative	False Positive	False Positive
Doc 4	True Negative	False Positive	False Positive
Doc 5	True Negative	True Negative	False Positive
Doc 6	False Negative	False Negative	True Positive
Doc 7	True Negative	True Negative	True Negative
Doc 8	True Negative	True Negative	True Negative
Doc 9	False Negative	False Negative	False Negative
Doc 10	True Negative	True Negative	True Negative
Precision = tp / (tp+fp)	= 1/ (1+ 0) = 1	= 1 / (1+2) = 0.33	= 2 / (2+3) = 0.4
Recall = tp / (tp + fn)	= 1/ (1+ 2) = 0.33	= 1 / (1+2) = 0.33	= 2 / (2+1) = 0.67