FIT5047 Intelligent Systems
Lab Assignment 2

Jaimon Thyparambil Thomas
Student ID: 29566428
Email: jthy0001@student.monash.edu

Nikhil Suresh
Student ID: 24325880
Email: nsur13@student.monash.edu

Question 1:

1. Use the WEKA visualization tool to analyze the data, and report briefly on the types of the different variables and on the variables that appear to be important.

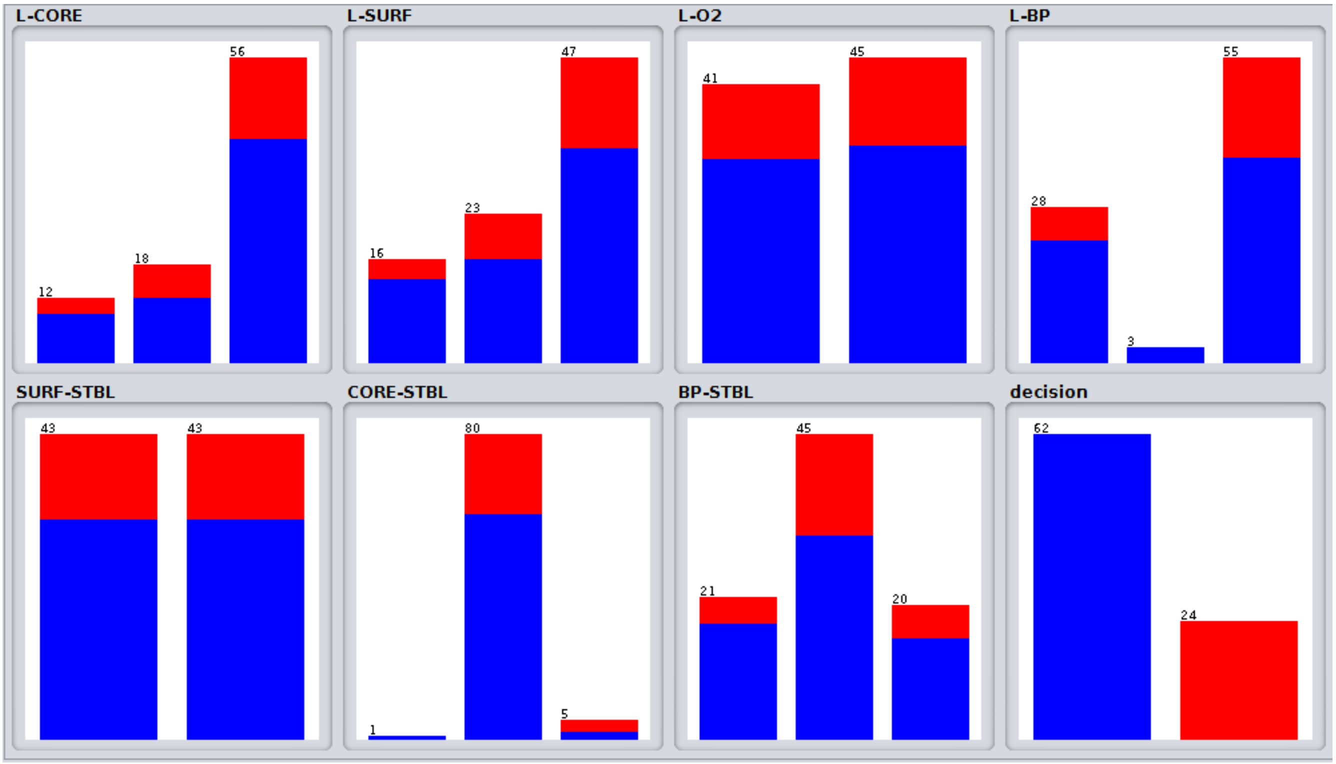


Figure 1. *Visualization of the variables in the dataset.*

To determine whether a variable is important, we want it to be the case that a case can be made that, after observing a variable, we can get a better estimate of the probability that a different **decision** was applied to a person than the base rate.

L-CORE: This variable has three levels. All three levels seem to have similar ratios of blue (discharged, S) to red (remain, A) classifications. The third level of this variable seems to indicate a higher level of a discharge. This variable can probably be used for predictions, but won't be the strongest predictor.

L-SURF: This variable also has three levels. The ratios are once against fairly similar, so this would not be expected to be a major predictor.

L-O2: Both the levels in this variable are similar, and likely only allow for a small improvement in classification accuracy.

L-BP: The first two levels of this variable seem to indicate a higher probability of a discharge decision. However, it is worth noting that the second level only has a few data points in it, so it is probably not very reliable as a predictor on its own.

SURF-STBL: This variable provides no information about whether a person was discharged or not, as people under each level were equally likely to belong to the other group.

CORE-STBL: This variable seems like quite a strong predictor of the decision to discharge or not. The first level consists of only one observation and is likely not very useful. The second level indicates an increased probability of being discharged, while the third indicates an increased probability of not being discharged.

BP-STBL: Finally, the first and last levels of this variable indicate a high probability of being discharged.

2. Run J48, Naive Bayes and k-NN to learn models. Perform cross-validation and report results.

NOTE: All algorithms are run with a seed of 1.

a) J48:

All parameters were left at their defaults, other than binarySplits. For the first trial, we enabled binarySplits.

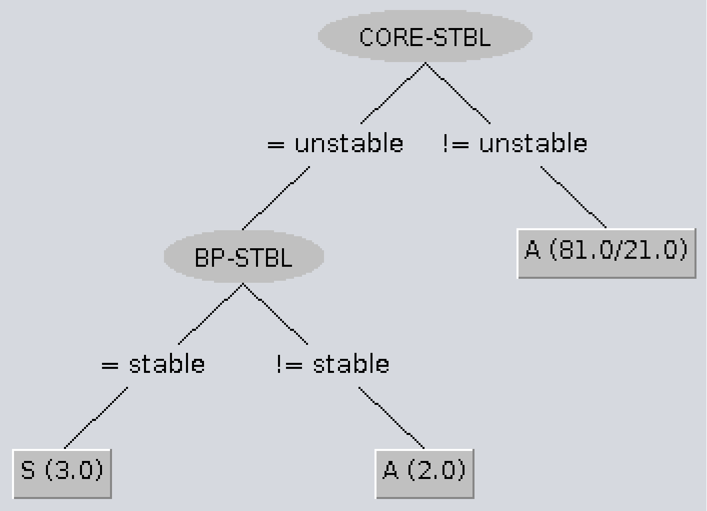


Figure 1. *First run of the J48 model.*

For the second trial, we enabled binarySplits.

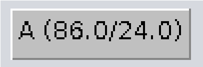


Figure 2. *Second run of the J48 model.*

For the second run, with binarySplits disabled, the tree has been pruned back so far that is simply consists of a single node guessing the base rate.

i) Examine the decision tree and indicate which are the main variables.

We opted to use the tree from Figure 1, as it actually had some branches in it. As expected, the main variables were CORE-STBL and BP-STBL. CORE-STBL was more important than BP-STBL. BP-STBL contained very few observations in it after selecting for cases where CORE-STBL was ‘unstable’.

ii) What is the accuracy of the decision tree? Explain the results in the confusion matrix.

		Classification:		**Correct:**
		A	S	
True class:	A	61	1	61
	S	24	0	0
				61/86 = 70.93%

Of patients in A, the final model successfully identified 61 of the 62 patients that were in this category from the predictors. Of the patients in S, none were classified correctly. Overall, 61 out of 86 patients were classified correctly, for an accuracy of 70.93%.

b) Naive Bayes.

i) Explain the meaning of probability distributions in the output with reference to BP-STBL.

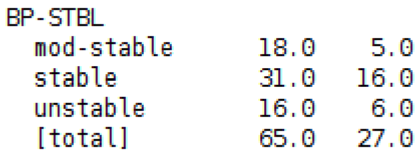
In Naive Bayes, it is typically assumed that every variable is independent of every other variable. However, the Naive Bayes algorithm implemented in WEKA engages in probability smoothing to deal with cases where a particular variable combination has 0 instances belonging to the outcome class.

The probability distribution is typically calculated as:

$$\frac{Probability(C) \times Probability(X|C)}{Probability(X)} \qquad \text{(Equation 1)}$$

However, when calculating the probability of X given C, Weka adds one to the numerator, and adds one to the denominator *per number of levels in variable X*.

So for example, looking at the output of BP-STBL:



```
BP-STBL
  mod-stable      18.0    5.0
  stable          31.0   16.0
  unstable        16.0    6.0
  [total]         65.0   27.0
```

the rightmost column is the ‘smoothed’ number of counts of the number of people with that particular level of BP-STBL that were classified as S. So for example, it says that ‘5’ people who were ‘mod-stable’ were classified as S, which was in fact 4 people. Similarly, the total of 27 is the actual total plus 3 (one for each level of BP-STBL).

ii) Calculate, by hand, the hypothetical probability of S.

<u>Probability of S, where α is a normalizing constant:</u>

P(L-CORE = mid/S) = 15/24

P(L-SURF = low/S) =7/24

P(L-O2 = good/S) =13/24

P(L-BP = high/S) =6/24

P(SURF-STBL = stable/S) =12/24

P(CORE-STBL = stable/S) =21/24

P(BP-STBL = mod-stable/S) = 4/24

P(S) = 24/86

P(S/ L-CORE = mid, L-SURF = low, L-O2 = good, L-BP = high ,SURF-STBL = stable, CORE-STBL = stable, BP-STBL = mod-stable)

= α* P(L-CORE = mid/S) * P(L-SURF = low/S) * P(L-O2 = good/S) * P(L-BP = high/S) * P(SURF-STBL = stable/S) * P(CORE-STBL = stable/S) * P(BP-STBL = mod-stable/S) * P(S)

= α * 15/24 * 7/24 * 13/24 * 6/24* 12/24 * 21/24 * 4/24 * 24/86

= α *198132480/394436542464

= α * 0.00050

Probability of A, where α is a normalizing constant:

P(L-CORE = mid/A) = 41/62

P(L-SURF = low/A) =16/62

P(L-O2 = good/A) =32/62

P(L-BP = high/A) =22/62

P(SURF-STBL = stable/A) =31/62

P(CORE-STBL = stable/A) =59/62

P(BP-STBL = mod-stable/A) = 17/62

P(A) = 62/86

P(A/ L-CORE = mid ,L-SURF = low, L-O2 = good, L-BP = high ,SURF-STBL = stable ,CORE-STBL = stable, BP-STBL = mod-stable)

= α* P(L-CORE = mid/A) * P(L-SURF = low/A) * P(L-O2 = good/A) * P(L-BP = high/A) * P(SURF-STBL = stable/A) * P(CORE-STBL = stable/A) * P(BP-STBL = mod-stable/A) * P(A)

= α* 41/62 * 16/62 * 32/62 * 22/62 * 31/62 * 59/62 * 17/62 * 62/86

= α * 890288605184/302858856133888

= α * 0.00293

iii) What is the probability that a person with these attributes will remain in hospital and that s/he will be discharged?

The probability of both outcomes must sum to one. We can get the probability of outcome A, ignoring α, by 0.00293 / (0.0005+ 0.00293) = 0.8542. This means there is an 85.42% chance of A under the Naive Bayes model. By extension, the probability of S would be 100 – 85.42 = 14.58%. The Naive Bayes classifier would thus choose outcome A.

iv)

		Classification:		**Correct:**
		A	S	
True class:	A	58	4	58
	S	22	2	2
				60/86 = 69.77%

The Naive Bayes classifier successfully identifies 58 cases of people who were in A, and 2 cases of people who were in S. The overall accuracy is thus 60 correct predictions of out 82. However, it should be noted that unlike the other two algorithms, it successfully identified some cases of S.

c) KNN.

Hypothetical patient (ID = H):

L-CORE = mid (3 levels)

L-SURF = low (3 levels)

L-O2 = good (2 levels)

L-BP = high (3 levels)

SURF-STBL = stable (2 levels)

CORE-STBL = stable (3 levels)

BP-STBL = mod-stable (3 levels)

i) Select three ‘similar’ points and calculate Jaccard coefficients. Similar was not defined, so three points that shared the same CORE-STBL level were selected and manually iterated over until points that were not too different were found:

Patient ID*	**H**	**47**	**39**	**10**	**Union**
L-CORE	mid	mid	low	mid	{low, mid}
L-SURF	low	mid	low	low	{low, mid}
L-O2	good	excellent	good	excellent	{good, excellent}
L-BP	high	mid	mid	mid	{mid, high}
SURF-STBL	stable	unstable	stable	unstable	{stable, unstable}
CORE-STBL	stable	stable	stable	stable	{stable}
BP-STBL	mod-stable	stable	stable	mod-stable	{stable, mod-stable}
Decision	-	A	S	S	-
Matches**	-	2	4	4	-

*Patient ID refers to the instance for that point in the data file, where H is the hypothesized patient.

** Matches is the number of elements in the intersection between the hypothesized patient and the selected cases.

The number of elements in the denominator of the Jaccard coefficient is the union of all variable levels across the two points.

The similarity of patient 47 is thus:

Unions of all variables = {mid} , {low, mid}, {good, excellent}, {high, mid}, {stable, unstable}, {stable}, {stable, mod-stable}.

Denominator = 12.

Similarity = 2/12 = 0.16

The similarity of patient 39 is thus:

Unions of all variables = {mid, low} , {low}, {good}, {high, mid}, {stable}, {stable}, {stable, mod-stable}.

Denominator = 10.

Similarity = 4/10 = 0.40

The similarity of patient 10 is thus:

Unions of all variables = {mid} , {low}, {good, excellent}, {high, mid}, {stable, unstable}, {stable}, {mod-stable}.

Denominator = 10.

Similarity = 4/10 = 0.40

The final classification for the hypothetical patient is:

Decision A has a summed Jaccard coefficient of 0.16.

Decision S has a summed Jaccard coefficient of 0.8.

Under weighted KNN, the patient is classified as S.

ii) We run the KNN classifier using 1-NN and 5-NN, with all other parameters set to defaults.

1-NN classifier:

		Classification:		**Correct:**
		A	S	
True class:	A	53	9	53
	S	24	0	0
				53/86 = 61.63%

5-NN classifier:

		Classification:		**Correct:**
		A	S	
True class:	A	61	1	61
	S	24	0	0
				61/86 = 70.93%

Using 1-NN (61.63% accuracy) means that the KNN algorithm has the most flexibility (simply being classified as whatever the next variable is, regardless of weighting), implying high variance and low bias. The 5-NN classifier (70.93% accuracy) becomes closer to simply guessing at whatever the base rate is as the the selection for K approaches the sample size. The 1-NN classifier makes many mistakes because it is too flexible and has overfit the data. The 5-NN classifier is less flexible and more biased towards guessing the average result (A), and thus achieves better accuracy on this dataset.

3. Compare all three algorithms.

Algorithm	**Accuracy**	**Kappa**
J48 (split tree)	70.93%	-0.0228
Naive Bayes	69.77%	0.0244
K-NN (K = 5)	70.93%	-0.0228

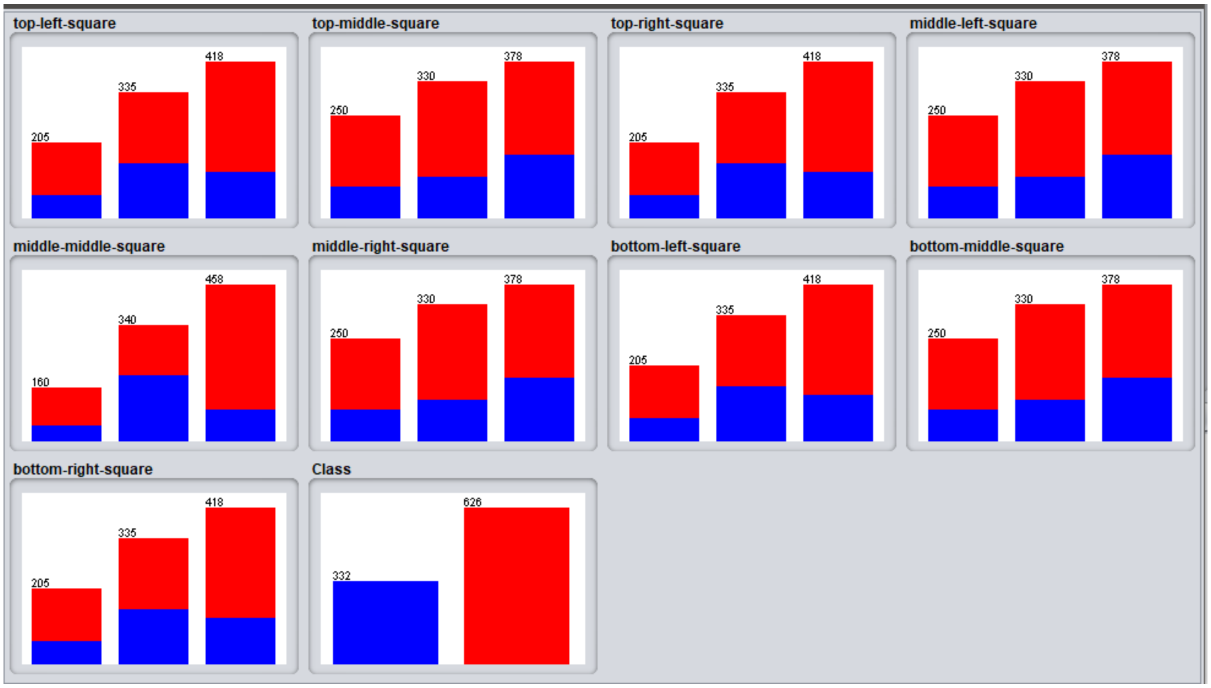
J48 and K-NN achieve the highest accuracies by a small margin (one more correctly classified case than Naive Bayes), but the Naive Bayes algorithm successfully classifies a few cases of patients who are in class S (kappa = 0.0244), whereas J48 and K-NN accomplish the higher accuracy by simply guessing the most frequent case (A) more often. This is reflected by their lower kappa statistics (they are worse than simply picking (A) every time). Thus the Naive Bayes algorithm arguably performs the best.

Question 2:

Before you run the classifiers, use the weka visualization tool to analyze the data. (2 + 2 = 4 marks) (a) Which attributes seem to be the most predictive of winning or losing? (hint: if you were the “x” player, where would you put your first cross and why?)

Interpretation:

First column represents ‘blank’.
Second column represents ‘o’.
Third column represents ‘x’.
Red – colour represents ‘x’ won.
Blue – colour represents ‘o’ won.



Figure

The attributes seem most predictive of winning are:”
1) The side that takes the middle-middle-square gets an advantage.
2) Taking any of the corner squares (top-left, top-right, bottom-left, bottom-right).
3) Playing as ‘x’, as ‘x’ wins far more games than ‘o’.

(b) What can you infer about the advantage (or otherwise) of being the first player?

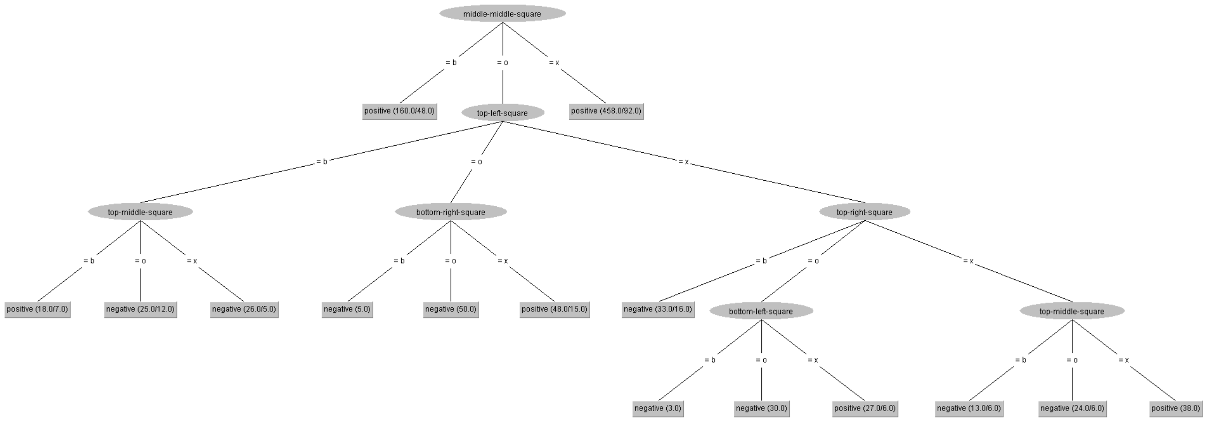
The advantage of being the first player is likely that, in tic-tac-toe, one has access to selecting the best squares. The *second* player is restricted to options that have not alreay been taken by the first player. In the case above, the main advantage of going first seems to be the ability to pick the middle-middle square.

2. Run J48 (=C4.5, decision tree), Na¨ıve Bayes and IBk (=k-NN) to learn a model that predicts whether the “x” player will win. Perform 10-fold cross validation, and analyze the results obtained by

these algorithms as follows. Note: When using IBk, click on the “Choose” bar to set the value of k (default is 1). Consider different values of k. (a) J48 (=C4.5) (2 + 3 + 14 + 3 = 22 marks)

i. Examine the decision tree and indicate the main variables.

When minobj = 20



The main variables are

1)Middle – middle square

2)Top – left square

3)top – middle , bottom –right

ii. Trace the decision tree for the following game. What would it predict?

The decision tree predicts a loss for ‘x’, (i.e, negative).

iii. What is the first split in the decision tree? Calculate (by hand) the Information Gain obtained from the first split in the tree. Show your calculations.

The first split in the decision tree is middle - middle square

Information gain

H(S) = (332-,626+) = -332/958 log$_2$(332/626) - 626/958 log$_2$(626/958)= 0.7182

H(b) = (48-,112+)= -48/160 log$_2$(48/160) - 112/160 log$_2$(112/160) = 0.8812

H(o) = (192-,148+)= -192/340 log$_2$(192/340) - 148/340 log$_2$(148/340) = 0.9878

H(x) = (92-,366+)= -92/458 log$_2$(92/458) - 366/458 log$_2$(366/458) = 0.7236

Information Gain = H(S) - (160/958 * H(b) + 340/958 * H(o) + 458/958 *H(x))

= 0.7182 - (160/958 * 0.8812 + 340/958 * 0.9878 + 458/958 * 0.7236)

= 0.7182 - 0.8436

= -0.1254

iv. What is the accuracy of the decision tree? Explain the results in the confusion matrix.

		Classification:		**Correct:**
		Loss*	Win*	
True class:	Loss	151	181	151
	Win	50	576	576
				727/958 = 75.89%

* Losses and wins are with respect to ‘x’.

Out of all 332 losses for ‘x’, the decision tree correctly identified 151. Of the 626 wins for ‘x’, the decision tree successfully predicts 576. This results in a total of 727 correct predictions out of 958 for a total accuracy of 75.89%.

Naïve Bayes

i. Calculate (by hand) the predicted probability of a win for the following game. Show your calculations.

```
P(Positive) = 626/958
P(TL=x|Positive)=295/626
P(TM =x |Positive)=225/626
P(TR = b|Positive)= 142/626
P(ML = 0|Positive)= 229/626
P(MM =0|Positive)= 148/626
P(MR = x |Positive)= 225/626
P(BL =b |Positive)= 142/626
P(BM=0|Positive)= 229/626
P(BR=b |Positive)= 142/626

P(Positive|TL = x, TM =x, TR = b, ML = 0, MM =0, MR = x, BL =b, BM=0,
BR=b)

= α * P(TL=x|Positive) * P(TM =x |Positive) * P(TR = b|Positive) * P(ML =
0|Positive) * P(MM =0|Positive) * P(MR = x |Positive) * P(BL =b |Positive)
* P(BM=0|Positive) * P(BR=b |Positive) * P(Positive)

= α * 295/626 * 225/626 * 142/626 * 229/626 * 148/626 * 225/626 * 142/626
* 229/626 * 142/626 * 626/958
= α * 0.00001469

P(Negative) = 332/958
P(TL=x| Negative)=123/332
P(TM =x | Negative)=153/332
P(TR = b| Negative)= 63/332
P(ML = 0| Negative)= 101/332
P(MM =0| Negative)= 192/332
P(MR = x | Negative)= 153/332
P(BL =b | Negative)= 63/332
P(BM=0| Negative)= 101/332
P(BR=b | Negative)= 63/332
```

P(Negative|TL = x, TM =x, TR = b, ML = 0, MM =0, MR = x, BL =b, BM=0, BR=b)

= α * P(TL=x| Negative) * P(TM =x | Negative) * P(TR = b| Negative) * P(ML = 0| Negative) * P(MM =0| Negative) * P(MR = x | Negative) * P(BL =b | Negative) * P(BM=0| Negative) * P(BR=b | Negative) * P(Negative)

= α * 123/332 * 153/332 * 63/332 * 101/332 * 192/332 * 153/332 * 63/332 * 101/332 * 63/332 * 332/958

= α * 9.97 * 10^{-6}

We know that
α * 0.00001469 + α * 9.97*10^-6 = 1
∴ α = 1/0.00002466

∴ P(Positive|TL = x, TM =x, TR = b, ML = 0, MM =0, MR = x, BL =b, BM=0, BR=b)
= 0.00001469/0.00002466
= 0.5957 = 59.57%

ii. What is the probability that a player with this configuration will win? What would the Na¨ıve Bayes classifier predict for this game?

The probability that the player with these moves will win is 59.57%. The Bayes classifier predicts that 'x' will win this game.

iii. What is the accuracy of the Na¨ıve Bayes classifier? Explain the results in the confusion matrix.

		Classification:		**Correct:**
		Loss*	Win*	
True class:	Loss	142	190	142
	Win	101	525	515
				667/958 = 69.62%

* Losses and wins are with respect to 'x'.

Out of all 332 losses for 'x', the decision tree correctly identified 142. Of the 626 wins for 'x', the decision tree successfully predicts 525. This results in a total of 667 correct predictions out of 958 for a total accuracy of 75.89%.

(c) k-NN (6 + 2 = 8 marks)

i. Find three instances in the dataset that are similar to the following game, and use the Jaccard coefficient to calculate (by hand) the predicted outcome for this game. Show your calculations.

i) Select three ‘similar’ points and calculate Jaccard coefficients. Similar was not defined, so three points that shared the same middle-middle and top-left square were selected.

Game ID*	H	68	56	28	Union
Left Top	x	x	x	x	{x}
Middle Top	x	x	x	x	{x}
Right Top	b	x	x	x	{b, x}
Left Middle	o	b	b	o	{b, o}
Middle Middle	o	o	o	o	{o}
Right Middle	x	b	x	b	{x, b}
Left Bottom	b	b	b	x	{x, b}
Middle Bottom	o	o	o	o	{o}
Right Middle	b	b	o	b	{b, o}
Decision	-	Win	Win	Win	-

*Game ID refers to the instance for that point in the data file, where H is the hypothesized game.

The number of elements in the denominator of the Jaccard coefficient is the union of all variable levels across the two points.

The similarity of game 68 is thus:

Unions of all variables = {x} , {x}, {x, b}, {o, b}, {o}, {x, b}, {b}, {o}, {b}.

Denominator = 12

Similarity = 6/12 = 0.50

The similarity of game 56 is thus:

Unions of all variables = {x} , {x}, {x, b}, {o, b}, {o}, {x}, {b}, {o}, {b, o}.

Denominator = 12

Similarity = 6/10 = 0.50

The similarity of patient 28 is thus:

Unions of all variables = {x} , {x}, {x, b}, {o}, {o}, {x, b}, {b, x}, {o}, {b}.

Denominator = 12.

Similarity = 6/10 = 0.50

The final classification for the hypothetical patient is:

Decision ‘Win’ has a summed Jaccard coefficient of 1.5.

Decision ‘Lose’ has a summed Jaccard coefficient of 0 (all three similar cases were ‘wins’).

Under weighted KNN, the game is predicted to be a win.

ii. What is the accuracy of the k-NN classifier? Explain the results in the confusion matrix.

K = 1

		Classification:		**Correct:**
		Loss*	Win*	
True class:	Loss	323	9	323
	Win	1	625	625
* Losses and wins are with respect to ‘x’.				948/958 = 98.96%

K = 2

		Classification:		**Correct:**
		Loss*	Win*	
True class:	Loss	323	9	323
	Win	1	625	625
* Losses and wins are with respect to ‘x’.				948/958 = 98.96%

K = 5

		Classification:		**Correct:**
		Loss*	Win*	
True class:	Loss	323	9	323
	Win	1	625	625
* Losses and wins are with respect to ‘x’.				948/958 = 98.96%

(a) (5.5 + 2.5 = 8 marks) Draw a table to compare the performance of J48, Na¨ıve Bayes and IBk using the summary measures produced by weka. Which algorithm does better? Explain in terms of weka’s summary measures. Can you speculate why?

Algorithm	**Accuracy**	**Kappa**
J48 (split tree)	75.89%	0.4132
Naive Bayes	69.62%	0.2843
K-NN (K = 1)	98.96%	0.9768

The K-NN algorithm achieves much better performance than the other two algorithms. A reason for this might be that not only is a win ‘correlated’ with the positions that have been selected by each player, they are actually defined by the final position of all the points. That is, the same combination of variables will always have the same outcome variable. As the sample size is quite large and the order the moves were made does not influence the calculation as to who won when reviewing the board state, the 'nearest neighbour' is very likely to be an identical board state, which guarantees a correct prediction.

Question 3: Regression (1 + 2 + 5 = 8 marks) Consider the dataset abs.arff available on moodle. This dataset contains continuous-valued economic attributes of a country, with the target variable being the unemployment rate. Additional documentation regarding these attributes appears in the arff file.

1. Perform a linear regression (under functions in weka) to learn a linear model of the unemployment rate as a function of the other variables. You can use the default parameters given in weka. What is the resultant regression function?

   Unemployment-Rate = -0.0014 * All-Ords-Index + -0.2452 * Housing-Loan-Interest-Rate + 13.7286

2. Using the resultant regression function, calculate by hand the Absolute Error for the year 1986.

```
Unemployment rate = -0.0014 * 1779.1 + -0.2452 * 15.5 + 13.7286
                  = 7.43726
```

2. Calculate the Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) obtained by the regression function (you can use the excel spreadsheet provided on moodle). How is MAE different from RMSE? (do these functions emphasize different aspects of performance?)

   MAE = 1.0607
   RMSE = 1.3327

   As RMSE works by *squaring* each residual, the penalty for large deviations from the predicted value of the regression line grows exponentially with distance.