Edge Hill University

The Department of Computer Science

CIS3156/CIS3132 Intelligent Systems

Level 6

INDIVIDUAL PORTFOLIO

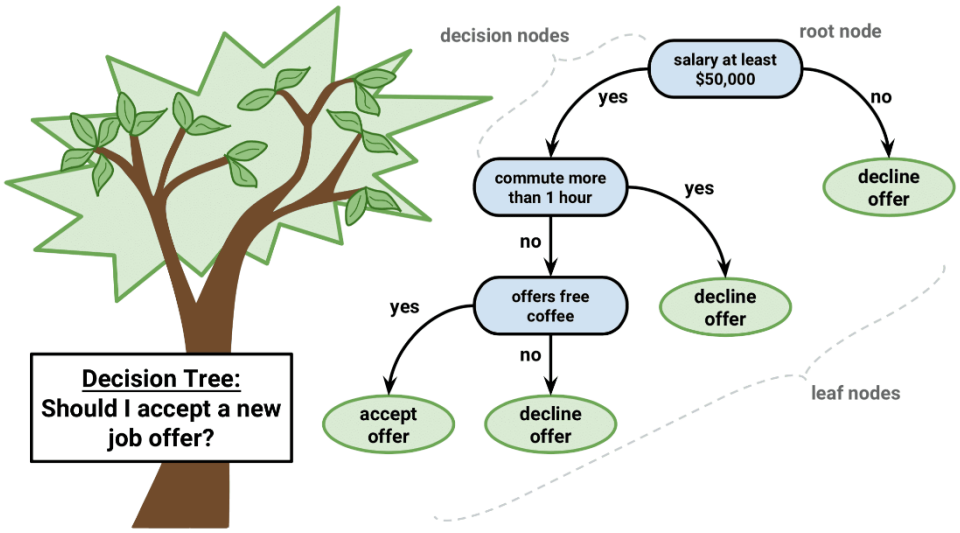
Jakub (Jacob) Strykowski

TABLE OF CONTENTS

TASK 1

DECISION TREE

Decision tree is an algorithm presentation tool, as can be on figure, decision tree presents data in way which is easy for human to understand the concept. Decision tree observations about an item are represented in the branches, item's target value is representing in the leaf. A leaf node represents a class label it follows that classification rules is the path from root to leaf. (nd, 2019) (nd, 2019) (Brid, 2018)



ID3 ALGORITHM

ID3 algorithm is an algorithm used to generate predictive modelling approaches used in statistics, data mining or machine learning or computer science and it was invented by Ross Quinlan. Id3 is a method to build the decision tree commonly used algorithm in machine learning which is covering classification and regression (Jazuli, 2018).

ADVANTAGES:

- The result of ID3 is interpretable by human a decisions tree. That help to deduce how it works or to find mistakes in an implementation.

- After build a tree (it takes a while) it very fastest classifier that is buildable. The cost for decision tree in Id3 is on the order of the depth of the tree, but that depth is going to be a lot smaller than the number of attributes.

- easily handles irrelevant attributes (Gain =0).

DISADVANTAGES:

- The algorithm at each level of the tree picking an attribute that is optimal for the current split. It maybe not the most optimal global solution for that reason ID3 is called greedy ones.
    - Exponentially many leaf nodes when depth of data is big, and it can be better decision tree than build by ID3 in some cases.

It is limited in the kind of splits it makes in the data, so it should be used to categorical data with discrete values of attributes. It follows that is not efficient in case of real value data (vertical or horizontal split). The Id3 algorithm is viable if we have categorical data like in our exercise: sunny, rainy, overcast. Other case it is used is if objective value has discrete output values (Jazuli, 2018).

DESCRIPTION OF THE ID3 DECISION TREE

General implementation of id3 based on iteration on this steps:

Step1. Calculate entropy and information gain of unused each attribute
Step2. Select one with the largest information gain
Step3. Set that attribute as noodle
Step4. Stop if childes nodes would be leaf node else go to step1

At the top of decision tree is root node, which would be chose, based value of information gain form all attributes in the set. Root node has the highest value gain. A necessary value to count the gain is Entropy. They are defined in that mathematical formula figure 1 (d, 2019):

Entropy [edit]

Entropy $\mathrm{H}(S)$ is a measure of the amount of uncertainty in the (data) set S (i.e. entropy characterizes the (data) set S).

$$\mathrm{H}(S) = \sum_{x \in X} -p(x) \log_2 p(x)$$

Where,

- S – The current dataset for which entropy is being calculated
  - This changes at each step of the ID3 algorithm, either to a subset of the previous set in the case of splitting on an attribute or to a "sibling" partition of the parent in case the recursion terminated previously.
- X – The set of classes in S
- $p(x)$ – The proportion of the number of elements in class x to the number of elements in set S

When $\mathrm{H}(S) = 0$, the set S is perfectly classified (i.e. all elements in S are of the same class).

Figure 1

Etropy shows up how deegree or disorder set is, what can be called set's dispersal. For exmple porces of making popcorn is increasing entropy in kitchen.

(d, 2019)

Information gain [edit]

Information gain $IG(A)$ is the measure of the difference in entropy from before to after the set S is split on an attribute A. In other words, how much uncertainty in S was reduced after splitting set S on attribute A.

$$IG(S, A) = \mathrm{H}(S) - \sum_{t \in T} p(t)\mathrm{H}(t) = \mathrm{H}(S) - \mathrm{H}(S|A).$$

Where,

- $\mathrm{H}(S)$ – Entropy of set S
- T – The subsets created from splitting set S by attribute A such that $S = \bigcup_{t \in T} t$
- $p(t)$ – The proportion of the number of elements in t to the number of elements in set S
- $\mathrm{H}(t)$ – Entropy of subset t

Figure 2

Based on definition information gain is difference in entropy of system before and after split an attribute. In ID3 process would be identificatory which instance gives largest about of information to training process.

This part:

$$\sum_{t \subset T} p(t)\mathrm{H}(t)$$

Can be show in different way and called Average Information of attribute:

$$I(Attribute) = \sum \frac{p_i + n_i}{p+n} Entropy(A)$$

The process of building decision tree starts with decide with attribute will be root node.

CALCULATIONS

STEP1.

Entropy(set) = 1 because it has 4 positive and 4 negatives values.

Next, I figure the Gain of all attributes. For example, Gain(outlook)

Outlook	**Play_tennis**
Sunny	No
Sunny	No
Sunny	No

Entropy($Outlook_{Sunny}$) = 0

It follows that when an instance has attribute Outlook = Sunny the result equates 'No' (play_tennis = No)

Outlook	**Play_tennis**
Overcast	Yes
Overcast	Yes

Entropy($Outlook_{Overcast}$) = 0

It follows that when an instance has attribute Outlook = Sunny the result equates 'Yes' (play_tennis = Yes)

Outlook	**Play_tennis**

Rainy	Yes
Rainy	Yes
Rainy	No

Positive = 2 Negative = 1

Entropy($Outlook_{Rainy}$) = -(2/ (2+1)) * log2(2/3) - (1/3) * log2(1/3) = 0.918

Average_Information(Outlook) = 3/8 * 0 + 2/8 * 0 + 3/8 * 0.918 = 0.344

Gain(Outlook) = 1 – 0.344 = 0.666

The same operation is used to number rest of gains. Values are presented in table below.

STEP2

Attribute	Average information	Gain
Outlook	**0.666**	**0.344**
Temperature	0.936	0.061
Humidity	0.950	0.050
Windy	0.950	0.050

The highest number is Outlook that why it is chosen for **root node.**

STEP3 AND CHECK (STEP4)

Outlook

Now, it is necessary to count Entropy(Rainy) a result will be the new E(S) (*Figure 1*). Next step is to figure Gain (Temperature, Humidity, Windy | Rainy)

STEP1

Temperature

Temperature	**Play_tennis**
Mild	yes

Entropy($Temperature_{Mild}$) = 0

Temperature	**Play_tennis**
Cool	Yes

Cool	No

Entropy($Outlook_{Overcast}$) = 1

Average_Information(Temperature) = 2/3 * 1 = 2/3

Gain(Temperature) = 0.918 - 2/3 = 0.251

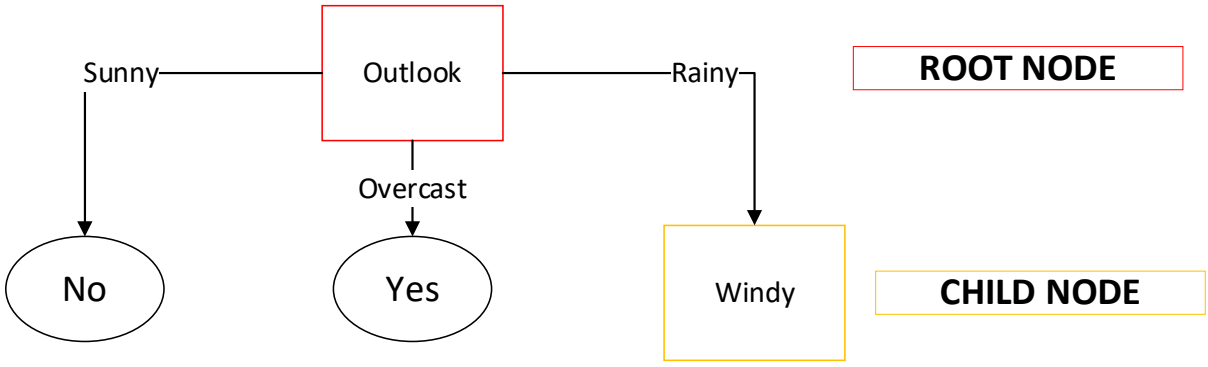
The same algorithm of counting it used to number Humidity and Windy.

STEP2

Attribute	Average information	Gain
Temperature	0.666	0.251
Humidity	0.666	0.251
Windy	**0**	**0.918**

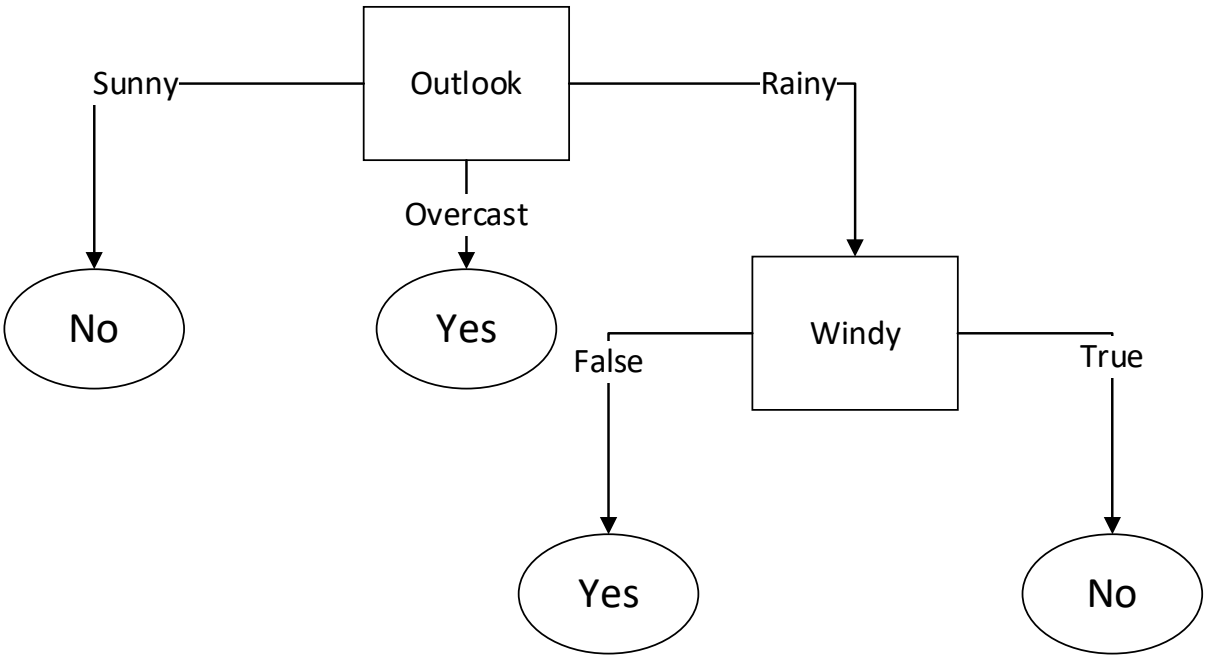
The attribute with the highest value of Gain is Windy, so it is child node.

STEP3 AND CHECK (STEP4)



Windy is new set, but Entropy(False) = Entropy(True) = 0. The complete decision tree can be finished.

STEP4 STOP WHEN CHILD NODES IS LEAF

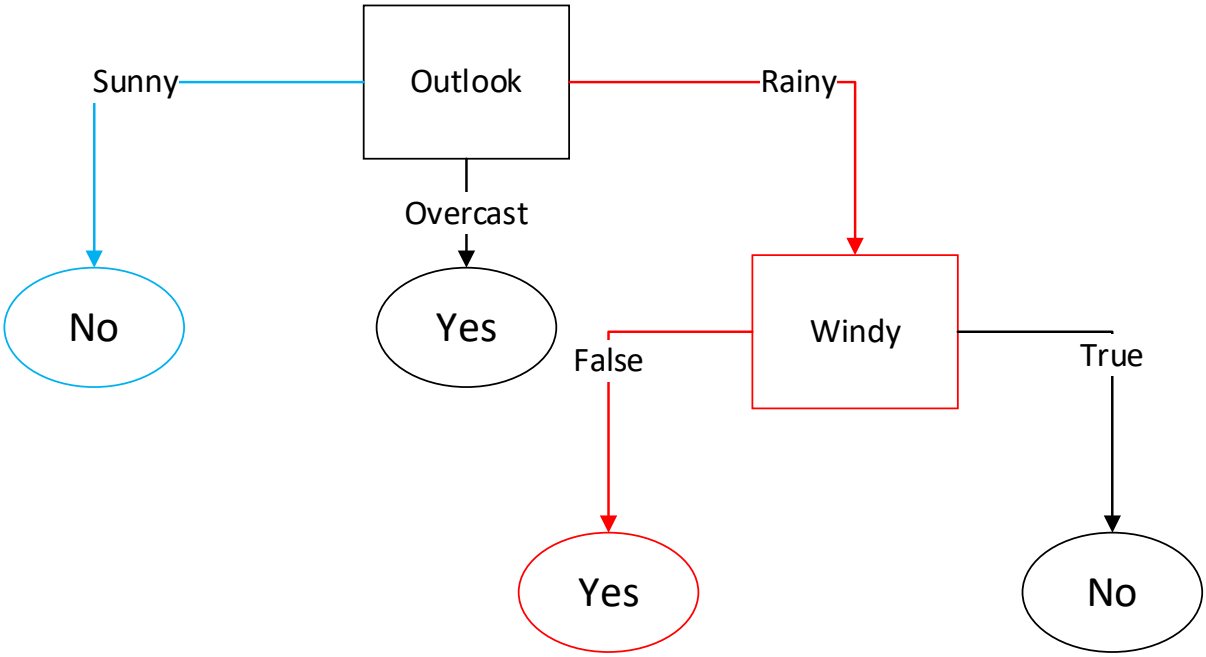


DESCRIPTION OF THE PREDICTION OF CLASSIFICATION LABEL OF THE TEST INSTANCES

	Outlook	Temperature	Humidity	Windy	Play_Tennis
Instance 9	Sunny	Cool	Normal	False	No
Instance 10	Rainy	Mild	Normal	False	Yes
Instance 11	Sunny	Mild	Normal	True	No

For instances 9 and 11 is simple decision path (blue path on decision tree). An answer is 'No'.

in case of instance 10 first step is match value of Outlook to Rainy and move to child node Windy. Next operation is to assign Windy to False and read form decision tree a result, 'Yes'. (Red path on decision tree)



REFERENCES

Anon., 2019. *F1 score.* [Online] Available at: https://en.wikipedia.org/wiki/F1_score

Anon., 2019. *Precision and recall.* [Online] Available at: https://en.wikipedia.org/wiki/Precision_and_recall

Brid, R. S., 2018. *Introduction to Decision Trees.* [Online] Available at: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb [Accessed October 2019].

d, n., 2019. *Decision tree learning.* [Online] Available at: https://en.wikipedia.org/wiki/Decision_tree_learning

Jazuli, H., 2018. *An Introduction to Decision Tree Learning: ID3 Algorithm.* [Online].

nd, 2019. *Decision tree learning.* [Online] Available at: https://en.wikipedia.org/wiki/Decision_tree_learning [Accessed 10 October 2019].

nd, 2019. *ID3 algorithm.* [Online] Available at: https://en.wikipedia.org/wiki/ID3_algorithm [Accessed 10 October 2019].

Nicholson, C., 2019. *Evaluation Metrics for Machine Learning - Accuracy, Precision, Recall, and F1 Defined.* [Online].

Shung, K. P., 2018. *Accuracy, Precision, Recall or F1?.* [Online] Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

TASK2

MY RECOMMENDATION FOR CEO

The question which algorithm is more accurate is difficult to answer. The process of preparation a result, for this question, requires looking for many variables. Both scenarios where algorithm would make mistakes in prediction are unsafe for patients. When value of 'precision' is high, but value of 'recall' is low the algorithm will have proper accuracy in detection cancer in group of ill people, otherwise, some person with cancer might not to be properly diagnosed (Algorithm Trees). In contrary scenario, value of 'precision' is low, and value of 'recall' is high, an algorithm will diagnose more ill person than first one, otherwise, it might classify healthy people as ill ones (Algorithm KNN).

$$Accuracy = \frac{True\ Positive + True\ Negatives}{Totalexamples}$$

HOW ACCURATE IS THAT MODEL?

The accuracy is parameter showing how a model works in general (It is too trivial to use it in this case). It is a simple model, otherwise, it does not contain information about severe class imbalance. (Shung, 2018) In many cases it is required to balance between 'false alarm' (false positive) and missing in detection (false negative). One of the possible answers for this case is F1 score, this metric is a result of equation of two other parameters 'precision' and 'recall'. That parameters are answers for those questions 'How useful the search are?' and 'How complete the result are?' (Nicholson, 2019).

My recommendation is to choose second algorithm 'Decision Trees. After my calculation that algorithm has bigger value of F1 score than 'Algorithm k-NN'. What does it mean?

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

THE PRECISION IS 'HOW USEFUL THE SEARCH RESULT ARE'

The necessity for using 'precision' is to avoid 'false alarms'. In some cases, it needs more accuracy to get proper results. For example: healthy patients have result that they are ill, and they are starting a dangerous and expensive medical treatment.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

RECALL IS "HOW COMPLETE THE RESULTS ARE".

Recall parameter can how the accuracy when the information about true danger is crucial. For example, the detection of nuclear missiles (Nicholson, 2019). If it was not detected, many people would die. Also, in the medical conditions it is important, because ill patient would be left without necessary treatment and they may even die.

$$F1score = \frac{2 * precision * recall}{precision + recall}$$

WHY F1 SCORE IS USEFUL?

F1 score is needed in cases require balance between precision and recall. Where an F1 score has a best value in 1 that means that this algorithm does not have false alarms and classify all threats (Anon., 2019). However, in real world there are not perfect algorithms. That is why it is essential to consider what is our priority. The golden middle is F1 score which contain both of important valuables. However, when requirements would be different, in some cases it should be focus on different parameters. The question is what is more important: to avoid 'false alarms' or classification more cases. The answer is depending from specific conditions of each task.

CALCULATION AND ANALYSE THE DATA

	(actual) health status of patient	K-NN: (predicted) health status of patient	Decision Trees: (predicted) health status of patient
Patient 1	healthy	cancer	healthy
Patient 2	healthy	cancer	healthy
Patient 3	cancer	cancer	cancer
Patient 4	cancer	cancer	cancer
Patient 5	cancer	cancer	cancer
Patient 6	cancer	cancer	healthy
Patient 7	healthy	cancer	healthy
Patient 8	healthy	healthy	healthy
Patient 9	healthy	cancer	healthy
Patient 10	healthy	cancer	healthy

True Positive	actual status = cancer	result = cancer

False positive	actual status = no cancer	result = cancer
False negative	actual status = cancer	result = no cancer
True negative	actual status = no cancer	result = no cancer

Actual status	Algorithm KNN	Predicted Events	
		+	-
	+	TP = 4	FP = 5
	-	FN = 0	TN = 1

Actual status	Algorithm Decision Trees	Predicted Events	
		+ cancer	-
	+	TP = 3	FP = 0
	-	FN = 1	TN = 6

Actual status	Algorithm KNN	Predicted Events	
		+	-
	+	TP = 4	FP = 5
	-	FN = 0	TN = 1

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{4}{4+5} = \frac{4}{9}$$

This score means that four patients with cancer have positive result and five heathy patients have result that they are ill. There is a risk of treatment the healthy people, what can be paradoxical dangerous for their health.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{4}{4} = 1$$

The recall score of 1 is highest possible. That means that all ill patients would have true result about their illness.

$$F1score = \frac{2 * precision * recall}{precision + recall} = \frac{2 * \frac{4}{9} * 1}{\frac{4}{9} + 1} = \frac{\frac{8}{9}}{\frac{13}{9}} = \frac{8}{13} = 61\%$$

Actual status	Algorithm Decision Trees	Predicted Events	
		+ cancer	-
	+	TP = 3	FP = 0
	-	FN = 1	TN = 6

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{3}{3} = 1$$

The precision score of 1 is highest possible. That means that all healthy patients would have true result about their illness.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{3}{3+1} = \frac{3}{4}$$

The recall score means that one patient with cancer was not diagnosed properly. There is the risk of this not detected cancer, the patients probably will die.

$$F1score = \frac{2*precision*recall}{precision+recall} = \frac{2*1*\frac{3}{4}}{1+\frac{3}{4}} = \frac{\frac{6}{4}}{\frac{7}{4}} = \frac{6}{7} = 86\%$$

Upper calculation prove theory that better algorithm in this condition is decision trees, which presented higher value of F1 score which as was written in introduction can be determinant for machine learning algorithms. The result where decision tree is better than KNN to test help of patient are interesting, because KNN can be seen like better way assessment humans' health, because humans are similar between each other and on KNN graphs would be present like groups of points, this characteristic of data makes knn better in predictions. However, the KNN produces bad assumption when dataset contain noisy instance for example one healthy patient smoking cigarettes in training set where many people died for caner. It can be assumed that in larger dataset and for higher k value knn would produce better predictions. On the other hand, as can be seen in this task decision tree presented better results than knn, this algorithm do not have problem with noisy data, because in process of building decision tree are chosen labels with higher information gain. The disadvantage of decision tree is making bad assumption based on lacky training label, in other words when training label do not have not enough different training instances the decision tree would not cover all cases and will make not good assumptions. However, when training dataset covered all possibly results, the decision tree would have f1 score equals 1. To sum up both decision trees and Knn have advantages and disadvantages, in that reason choosing an algorithm to prediction should base on calculation.

TASK 2.2 (ADVANCED – ONLY FOR 70+ MARKS)

ANSWER

The doctor suggestion means that more important is recall than precision. It should not be used F1 score, because it is desired to show balance between recall and precision. In this case require comparation of recall.

Algorithm	Recall
KNN	1
Algorithm Trees	$\frac{3}{4}$

The highest possible value of recall has KNN, there is the best algorithm to meet doctor's requirements. The recall score of 1 means that all ill patients would have true result about their illness. (Anon., 2019). Recall factor equals one means that all ill patients would be treated, in case of cancer treatment more important is to treat even healthy person in case of mistake, because KNN has lower precision. Often machine learning algorithm with increasing recall would lose precision and analogical in other way. The doctor decide that cancer illness will potentially kill fast person without medical treatment and in other case healthy person would lose health in case of mistake. The choosing between recall and precision is more than machine learning problem, because in both cases of mistakes produces ethical problem, in that reason algorithms should be consult with specialist in industry like in this task doctor.

REFERENCES

Anon., 2019. *F1 score.* [Online] Available at: https://en.wikipedia.org/wiki/F1_score

Anon., 2019. *Precision and recall.* [Online] Available at: https://en.wikipedia.org/wiki/Precision_and_recall

Brid, R. S., 2018. *Introduction to Decision Trees.* [Online] Available at: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb [Accessed October 2019].

d, n., 2019. *Decision tree learning.* [Online] Available at: https://en.wikipedia.org/wiki/Decision_tree_learning

Jazuli, H., 2018. *An Introduction to Decision Tree Learning: ID3 Algorithm.* [Online].

nd, 2019. *Decision tree learning.* [Online] Available at: https://en.wikipedia.org/wiki/Decision_tree_learning [Accessed 10 October 2019].

nd, 2019. *ID3 algorithm.* [Online] Available at: https://en.wikipedia.org/wiki/ID3_algorithm [Accessed 10 October 2019].

Nicholson, C., 2019. *Evaluation Metrics for Machine Learning - Accuracy, Precision, Recall, and F1 Defined.* [Online].

Shung, K. P., 2018. *Accuracy, Precision, Recall or F1?.* [Online] Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

TASK 4

Background

In this task, you will develop a supervised classification algorithm that predicts whether a passenger of the RMS Titanic would survive or not according to the following 3 attributes:

1. ticket_class={first, second, third}
2. age={young, old}
3. cabin={yes, no}

You are required to train a Naïve Bayes classifier on the provided training dataset (i.e., Table 1) and then apply your trained algorithm to the test dataset (i.e., Table 2) in order to automatically predict the label of the 1 test instance.

	ticket_class	age	cabin	survived
Instance 1	third	young	no	0
Instance 2	first	young	yes	1
Instance 3	first	old	yes	0
Instance 4	second	young	yes	0

Table 1.Training dataset

	ticket_class	age	cabin	survived
Instance 5	third	old	no	?

Table 2.Test dataset

NATIVE BAYES ALGORITHM

INTRODUCTION

Naïve Bayes is one of commonly used machine learning algorithm. A classifier in a model used to separate different instance based on certain instances**Invalid source specified.**. The algorithm applying Bayes's Theorem**Invalid source specified.**.

$$P(A|B) = \frac{\left(P(B|A) * P(A)\right)}{P(B)}$$

The world 'naïve' in the name of it follow from assumption of independence between pair of features in one training process. The classification process can be represented by Bayesian network.

Invalid source specified.

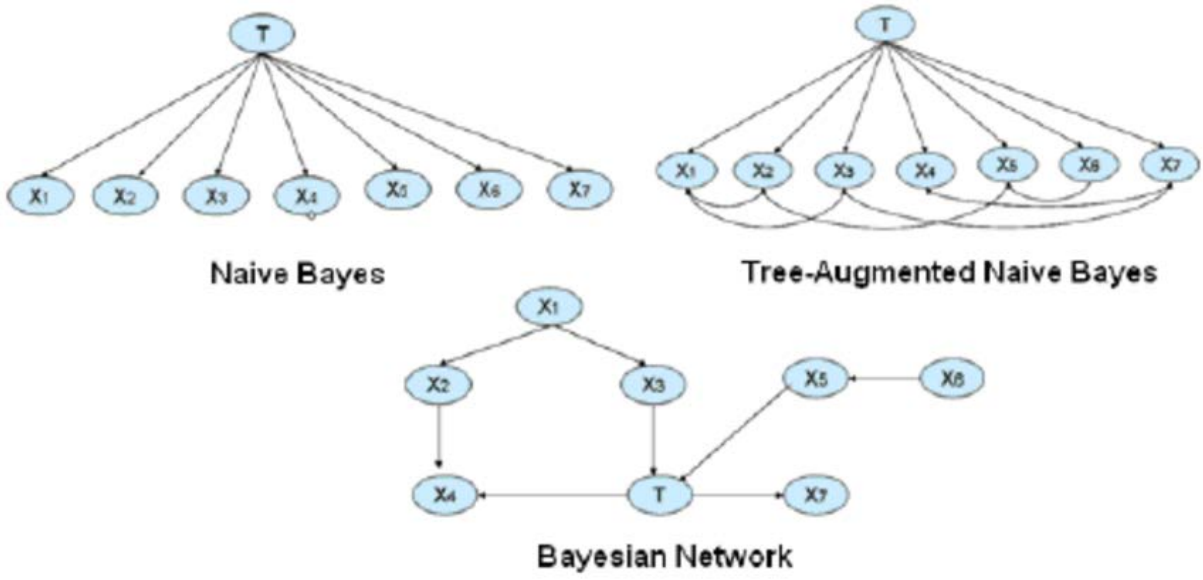


Figure 3

First step in analysing the Bayesian Network is reading it from top to the bottom. On the diagram *(figure1)* can be seen different between different assumptions. In naïve the assumption is that all features are independent that's why on diagram is flat and all feature are the same weight and not affect each other**Invalid source specified.**. This assumption makes this algorithm not always the best solution for real world problems where one event often follows next one, otherwise Naïve Bayes is used in sentiment analysis, spam filtering, recommendation systems etc**Invalid source specified.**. The naïve Bayes is commonly-used because it is cost-effective and easy to code.

ADVANTAGES

- The class of data set can be assumed in simple and fast way.
- The Naïve Bayes also have very good results in predicting in compare to different algorithms.
- It has good performance in situation of categorical input variables compared to numerical variables. The result of assuming for numerical variable is the normal distribution**Invalid source specified.**.

LIMITATION

- In case an instance has a category, which was not learn from training data set the model will assign 0 probability. It makes the algorithms useless in that concrete situation.
- Naïve Bayes is weak estimator**Invalid source specified.**.

GENERAL IMPLEMENTATION NAÏVE BAYES

General implementation contain tree for loops, there implementation will depend on size of datasheet, training algorithm can be shown in, where tabulation like in python programming means duration of for loop:

Step 1. For each label in dataset compute probability of that label

Step 2. For each feature of this label

Step 3. For each value of that feature compute probability of value given label

STEP-BY-STEP DESCRIPTION OF TRAINING PROCESS OF THE NAÏVE BAYES

The dataset provides contain irrumations about passengers of RMS Titanic. The feature this task would predict is survived.

STEP 1 for each label Survived and NoSurvived count overall possibility of survived and death.

P(Survived) = 25%

P(No Survived) = 75%

STEP2 To compute probability of labels for each feature ticket_class, age, cabin and building the frequency tables.

STEP3 for feature: ticket class

Feature: Ticket_class	*Yes*	*No*
First	½	½
Second	0	1
third	0	1

Table 1 represents the data feature 'ticket_class'. It shows that no one with third class ticket survived.

STEP3 for feature: age

Feature: age	*Yes*	*No*
young	1/3	2/3

Old	0	1

Table 2 represents the data feature 'age'. It shows that no one with old person survived.

STEP3 for feature: cabin

Feature: Cabin	*Yes*	*No*
Yes	1/3	2/3
No	0	1

Table 3 represents the data feature 'cabin'. It shows that no one without cabin survived.

STEP-BY-STEP DESCRIPTION OF TEST (PREDICTION) PROCESS OF THE NAÏVE BAYES

third	old	no

A predicting a result for test process of the Naïve Bayes requires a counting probability of given model. This model is a passenger with features: third ticket class, old and without cabin. Overall probability that passengers would like to survive is 25% (according to step 1 in training process). To solve an equation are needed probabilities of each one feature.

$$\boldsymbol{P(survieved|third \cap old \cap no)} = 25\% * 0 * 0 * 0 = 0$$

Next step is to count how probably is that the passenger did not survived.

$$\boldsymbol{P(no\_survieved|third \cap old \cap no)} = 75\% * 1 * 1 * 1 = 75\%$$

After calculations the result is 75%. The probability of 75% is higher than 0%, int that reason the second option is more probable. The highest probability has the case that passenger would have not survived with these features.

The results may change when the dataset would be larger. Then it will give more instances and probability can be higher than 0. Based on historic accident, overall probability of survival the RMS Titanic's sinking is about 30%**Invalid source specified.**.

REFERENCES

Anon., 2019. *F1 score.* [Online] Available at: https://en.wikipedia.org/wiki/F1_score

Anon., 2019. *Precision and recall.* [Online] Available at: https://en.wikipedia.org/wiki/Precision_and_recall

Brid, R. S., 2018. *Introduction to Decision Trees.* [Online] Available at: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb [Accessed October 2019].

d, n., 2019. *Decision tree learning.* [Online] Available at: https://en.wikipedia.org/wiki/Decision_tree_learning

Jazuli, H., 2018. *An Introduction to Decision Tree Learning: ID3 Algorithm.* [Online].

nd, 2019. *Decision tree learning.* [Online] Available at: https://en.wikipedia.org/wiki/Decision_tree_learning [Accessed 10 October 2019].

nd, 2019. *ID3 algorithm.* [Online] Available at: https://en.wikipedia.org/wiki/ID3_algorithm [Accessed 10 October 2019].

Nicholson, C., 2019. *Evaluation Metrics for Machine Learning - Accuracy, Precision, Recall, and F1 Defined.* [Online].

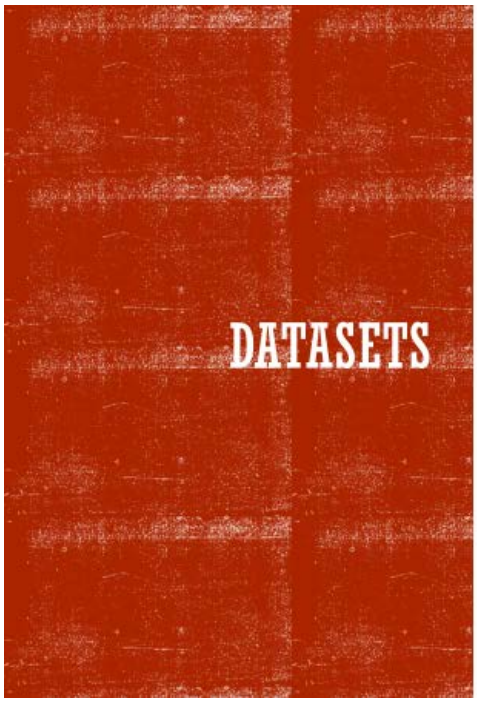
Shung, K. P., 2018. *Accuracy, Precision, Recall or F1?.* [Online] Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

TASK 3

CW 1.3 PRESENTATION

Nathan Dewhurst

Jakub Strykowski



This Challenge consisted of 4 different datasets:

- Tic Tac Toe
- Connect 4
- Chess
- Customer Churn

Each dataset had different numbers of training and test instances, and various attributes.

TIC-TAC-TOE DATASET

This dataset determines whether player 'X' can win in a certain configuration of Tic Tac Toe.

The dataset contains:

- 574 Training Instances
- 384 Testing Instances
- 9 Attributes

Each training instance contains:

- 9 attributes to represent the board configuration
- Each attribute is assigned a number - '2' represents an X, '1' represents an O, and '0' represents a blank space.
- A label that is assigned a number to determine if the current board configuration is a win or lose situation -
  - 1 – A win for player X
  - 0 – A draw or loss for player X

top-left-s	top-midd	top-right-	middle-le	middle-m	middle-ri	bottom-le	bottom-n	bottom-ri	win_for_x
1	1	2	1	2	1	1	2	1	0

F1 SCORE = 1

```
with open(filepath) as fp:
        line = fp.readline()
        cnt = 1
        while line:
            print("Line {}: {}".format(cnt, line.strip()))
            print(check(line))
            if check(line) == 1 and line[9] == '1':
                tp+= 1
            elif check(line) == 0 and line[9] == '0':
                tn+= 1
            elif check(line) == 1 and line[9] == '0':
                fp1+= 1
            elif check(line) == 0 and line[9] == '1':
                fn+= 1
            line = fp.readline()
            cnt += 1
precision = tp/(tp+fp1)
recall = tp/(tp+fn)
fscore = (2 * precision * recall)/(precision + recall)

def check(temp):
    if line[0] == '2' and line[1] == '2' and line[2]== '2':
        return 1
    elif line[3] == '2' and line[4] == '2' and line[5]== '2':
        return 1
    elif line[6] == '2' and line[7] == '2' and line[8]== '2':
        return 1
    elif line[0] == '2' and line[3] == '2' and line[6]== '2':
        return 1
    elif line[1] == '2' and line[4] == '2' and line[7]== '2':
        return 1
    elif line[2] == '2' and line[5] == '2' and line[8]== '2':
        return 1
    elif line[0] == '2' and line[4] == '2' and line[8]== '2':
        return 1
    elif line[2] == '2' and line[4] == '2' and line[6]== '2':
        return 1
    else:
        return 0
```

CONNECT 4 DATASET

This dataset determines whether player 'Red' can win in a certain configuration of Connect 4.

The dataset contains:

- 30,400 Training Instances
- 4,000 Testing Instances
- 42 Attributes

Each training instance contains:

- 42 attributes to represent the board configuration
- Each attribute is assigned a number - '2' represents an X, '1' represents an O, and '0' represents a blank space.
- A label that is assigned a number to determine if the current board configuration is a win or lose situation -
  - 1 – A win for player Red
  - 0 – A draw or loss for player Black

a1	a2	a3	...	g5	g6	win_for_x
1	1	0	...	0	0	1
1	2	2	...	0	0	1
1	2	1	...	0	0	1

CHESS DATASET

This dataset determines the number of moves needed for the player 'white' to win the game

The dataset contains:

- 14,028 Training Instances
- 14,028 Testing Instances
- 6 Attributes

Each training instance contains:

- Attributes 1 and 2 that hold the Column and Row numbers of the White King
- Attributes 3 and 4 that hold the Column and Row numbers of the White Rook
- Attributes 5 and 6 that hold the Column and Row numbers of the Black King
- A label that states the number of moves needed for white to win (0 – 16) or if it is a draw(17)

White_King_column	White_King_row	White_Rook_column	White_Rook_row	Black_King_column	Black_King_row	depth_win_white
3	2	7	8	5	7	14
3	2	3	6	2	5	17
4	2	6	2	6	4	14

CUSTOMER CHURN DATASET

This dataset determines if a customer is likely to leave a company based on their information.

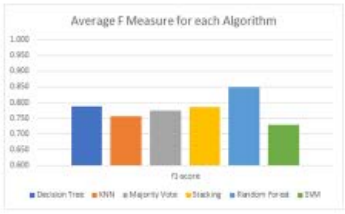
The dataset contains:

- 3,516 Training Instances
- 3,516 Testing Instances
- 19 Attributes

Each training instance contains:

- Various attributes that contains information on areas such as gender, tenure, and payment method
- A label that states if the customer will leave (1), or if the customer will not leave (0).

gender	SeniorCit	Partner	Depende	tenure	PhoneSe	MultipleL	InternetS	OnlineSe	OnlineBa	DevicePr	TechSup	Streamin	Streamin	Contract	Paperles	Payment	MonthlyC	TotalCha	Churn
0	0	1	0	1	0	1	0	0	2	0	0	0	0	0	1	2	29.85	29.85	0
1	0	0	0	34	1	0	0	2	0	2	0	0	0	1	0	3	56.95	1889.5	0



Results	precision	recall	f1-score
Decision Tree	0.795	0.789	0.789
KNN	0.759	0.762	0.756
Majority Vote	0.781	0.783	0.774
Stacking	0.787	0.785	0.785
Random Forest	0.849	0.852	0.848
SVM	0.760	0.755	0.729

BEST PERFORMING ALGORITHM

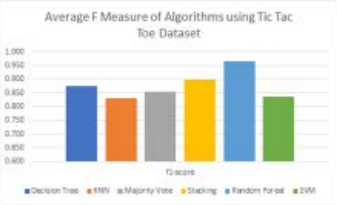
Algorithms Tested:

- Decision Tree
- K-NN
- Majority Vote
- Stacking
- Random Forest
- SVM

Best Performing Algorithm – **Random Forest (0.848)**

Number of Trees in Forest – **150**

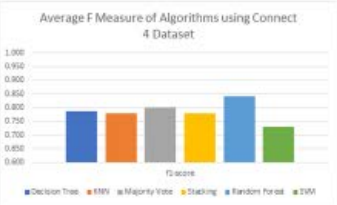
Features for each tree to sample - **7**



Tic Tac Toe	precision	recall	f1-score
Decision Tree	0.877	0.878	0.876
KNN	0.847	0.841	0.832
Majority Vote	0.874	0.862	0.854
Stacking	0.900	0.898	0.899
Random Forest	0.968	0.966	0.966
SVM	0.874	0.849	0.835

EXPERIMENTS: TIC TAC TOE DATASET

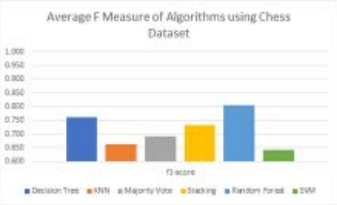
- Random Forest is the best performing algorithm when using the Tic Tac Toe dataset, resulting in an F-score of **0.966**.
- The second-best performing algorithm was the stacking algorithm, with a score of **0.899**.
- The stacking algorithm works by runninng various other algorithms, such as K-NN or decision, and then creating its own feature set to catagorise data with.
- The K values used for the two K-NN algorithms were '5' and '200'. This produced the highest score for the stacking algorithm.



Connect 4	precision	recall	f1-score
Decision Tree	0.787	0.787	0.787
KNN	0.780	0.785	0.780
Majority Vote	0.800	0.803	0.800
Stacking	0.780	0.779	0.779
Random Forest	0.844	0.846	0.842
SVM	0.764	0.757	0.731

EXPERIMENT: CONNECT 4 DATASET

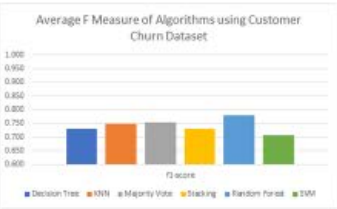
- Random Forest is the best performing algorithm when using the Connect 4 dataset, resulting in an F-score of **0.842**.
- The majority vote algorithm performed much better when using this dataset when compared to other algorithms, with a resulting score of **0.8**, and is greater than its average score of **0.774**.
- The majority vote consisted of two K-NN algorithms with K set to '6' and '12', as well as a decision tree.
- This combination of of K values yielded the greater F-score for the majority vote algorithm.



Chess	precision	recall	f1-score
Decision Tree	0.787	0.761	0.761
KNN	0.667	0.663	0.663
Majority Vote	0.694	0.690	0.690
Stacking	0.736	0.733	0.733
Random Forest	0.807	0.806	0.806
SVM	0.643	0.649	0.642

EXPERIMENT: CHESS DATASET

- Random Forest is the best performing algorithm when using the Chess dataset, resulting in an F-score of **0.806**.
- Unlike with other datasets, the Decision Tree algorithm performed best after the random forest, producing a score of **0.761**.
- This was likely due to the lower number of attributes used within this dataset. It allowed the decision tree algorithm to out-perform most other algorithms.



Customer Churn	precision	recall	f1-score
Decision Tree	0.730	0.730	0.730
KNN	0.743	0.758	0.748
Majority Vote	0.757	0.775	0.752
Stacking	0.730	0.729	0.729
Random Forest	0.776	0.788	0.779
SVM	0.759	0.763	0.707

EXPERIMENT: CUSTOMER CHURN DATASET

- Random Forest is the best performing algorithm when using the Customer Churn dataset, resulting in an F-score of **0.779**.
- The K-NN and Majority Vote algorithms were the best performing algorithms after Random Forest.
- The K-NN algorithm performed better than most algorithms using this data set because of the small number of data used in the dataset.
- The K-NN algorithm set K equal to 5. This meant that the algorithm was checking the 5 closest instances of data when testing.

The results were very clear that the best performing algorithm was Random Forest. This algorithm had the highest score when using any of the datasets.

However, the results also show different algorthms performing better when using some datasets rather than others, for example the K-NN algorithm. This shows that particukar algorithms perform best under particular circumstances.

An interesting fact is that the ml algorithm do not require knowledge of rules of games. They can made good prediction based only on training data. This feature makes them were cost-effective.