Machine Learning

Lesson 8

Multiclass Classification

Lecturer: Dr. Msagha J Mbogholi, PhD

Flashback from Lesson 7

- Reinforcement learning is a way by which a learner wishes to achieve a certain target (goal) and the steps that they take in order to achieve it.
- In reinforcement learning the agent must take a series of actions that will alter the state of the environment. In supervised learning the goal is known to the learner; RL is essentially a search to achieve the goal.
- When the discount rate is high the discount will be small. Conversely, when the discount rate is low the discount is big.
- RL tasks are classified as either being continual or episodic.
- Action selection methods include greedy, ε greedy and soft-max.
- RL learning methods are markov decision process (MDP), monte carlo and temporal difference (TD).
- Approaches to RL are value based, policy based and model based.

Content

- Introduction
- Binary Classification
- Multiclass Classification
- Multilabel Classification
- Imbalanced Classification

Part 1

Introduction

Introduction

- The different types of learning were introduced in lesson 1.
- Just as a quick refresher they were classified as: supervised, unsupervised, semi-supervised and reinforcement learning.
- Supervised learning involves the use of labels (desired outcomes) that are used to train the learner before introducing unseen data.
- Unsupervised learning involves the learner determining the outcomes by identifying patterns since there is no label provided; only inputs.
- Semi-supervised learning involves both supervised and unsupervised learning; that is to say, there is partially labeled data.
- Reinforcement learning is different in that it can be called a searching algorithm based on a reward system. This has been covered in detail in lesson 7.

Introduction (cont'd)

- Additionally models can be classified as being of two main types.
- In most domains the term descriptive is used to describe something that happened in the past so that it can be understood.
- For instance in database analytics, a descriptive model is used to explain or understand past behavior or trends.
- Thus a descriptive model will help to understand past behavior.
- A predictive model on the other hand deals with the future. The model is named after the verb 'predict' which essentially means to 'guess' an outcome.
- Machine learning is concerned with predictive models; in predictive modeling we aim to determine an outcome based on past and existing data (vide one of the learning methods above).

Introduction (cont'd)

- Predictive modeling in machine learning can be further grouped into three major groupings. It is worth noting that predictive modeling finds its strength in supervised learning.
- This is because in unsupervised learning there is no labeled data hence the learner must learn the different associations between the data and group them accordingly.
- Predictive models can further be classified as either classification models, regression models or artificial neural networks (ANN).
- The main difference between classification and regression is that whereas the former involves discrete data, the latter involves continuous data that results in a regression line.
- Artificial neural networks are not part of this course and will not be discussed.
- Classification requires a training set consisting of input data with the corresponding output labels. The learner will learn from the data and thus will be able to classify other data inputs according to classification.

Introduction (cont'd)

- A good example of classification is the classical spam filter. The learner is fed examples of spam and non spam (also called ham) and based on this it learns to classify other emails as one or the other based on the rules (the algorithm).
- So how is the performance of the model measured?
- One popular metric is the degree of accuracy in prediction.
- Further there are times when probability of class membership rather than class labels are used in determining the membership of data.
- Classification tasks can be further subdivided into four distinct groups: binary classification, multiclass classification, multilabel classification, and imbalanced classification.
- The relationships are captured in fig 1.
- The four subgroupings are the subject of this lesson.

Introduction (cont'd)

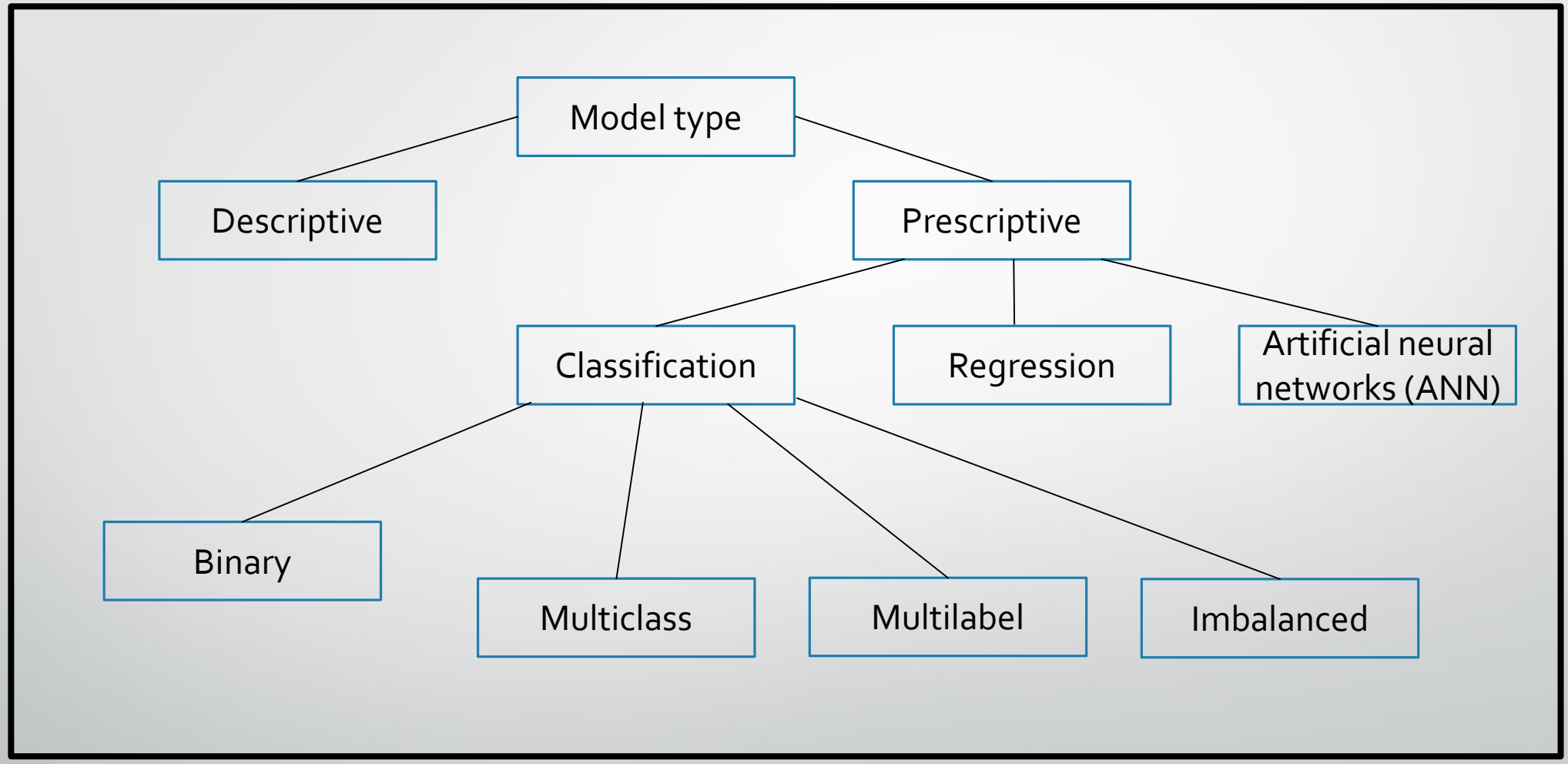


Fig 1. Model types taxonomy

Part 2

Binary Classification

2.1 Introduction

- Binary classification involves tasks that have two labels.
- You can look at binary classification in real life as the case of "either – or", meaning the data is either "this" or "that".
- Some examples of binary classification include:
- The classic email spam filter (spam or ham)
- Malaria diagnosis in health (has malaria or not)
- In loan applications (give loan or not)
- And so on.

2.2 How do they work?

- As can be seen one task will show the positive (normal) case, for example, the blood sample has no malaria; while the negative (abnormal) case will be that the blood sample has malaria.
- Each of the classes has to be assigned a label; the normal case is assigned a value 0, while the abnormal case is assigned the value 1.
- In our example the blood sample with no malaria is assigned the label value 0, while the sample which has malaria will be assigned the value 1.
- The algorithm in this case looks for the presence of the malaria parasite in the sample (depending on type of test).
- A probability distribution can also be used to determine the probability of data being in normal or abnormal state.
- Since there are only two possible outcomes the distribution used should support this; in this case a Bernoulli probability distribution is very suitable.
- A Bernoulli probability distribution is "a discrete probability distribution for a Bernoulli trial — a random experiment that has only two outcomes (usually called a "Success" or a "Failure")." (statitisticshowto.com)

2.2 How do they work? (cont'd)

- Applying the Bernoulli distribution to classification this means that the model will predict the probability that the data (example) belongs to the normal state (class 0), or otherwise (the abnormal state, class 1).
- Examples of algorithms covered so far using this classification method include:
- k – nearest neighbor
- Decision trees
- Logistic regression
- Support vector machine (SVM)
- As you may recall, logistic regression and SVM do not support more than two classes.
- Another popular algorithm that uses this classification method that will be covered in a later lesson is naïve bayes algorithm.

2.3 Example

- To demonstrate the working of a binary classifier let us consider the classical email spam filter. I call it classical since it is used as an example in many books and websites….probably due to the fact it is an easy enough example that many students can associate with? Perhaps.
- With many of these filters the rules are that the algorithm (learner) is given some examples of instances which are classified as spam; this will usually involve the presence of a word or phrase that can classify an email as spam.
- The absence of the word or phrase indicates that it is ham.
- Consider fig 2. this filter is trained to find spam using the word 'Viagra' and 'lottery'.

Example (cont'd)

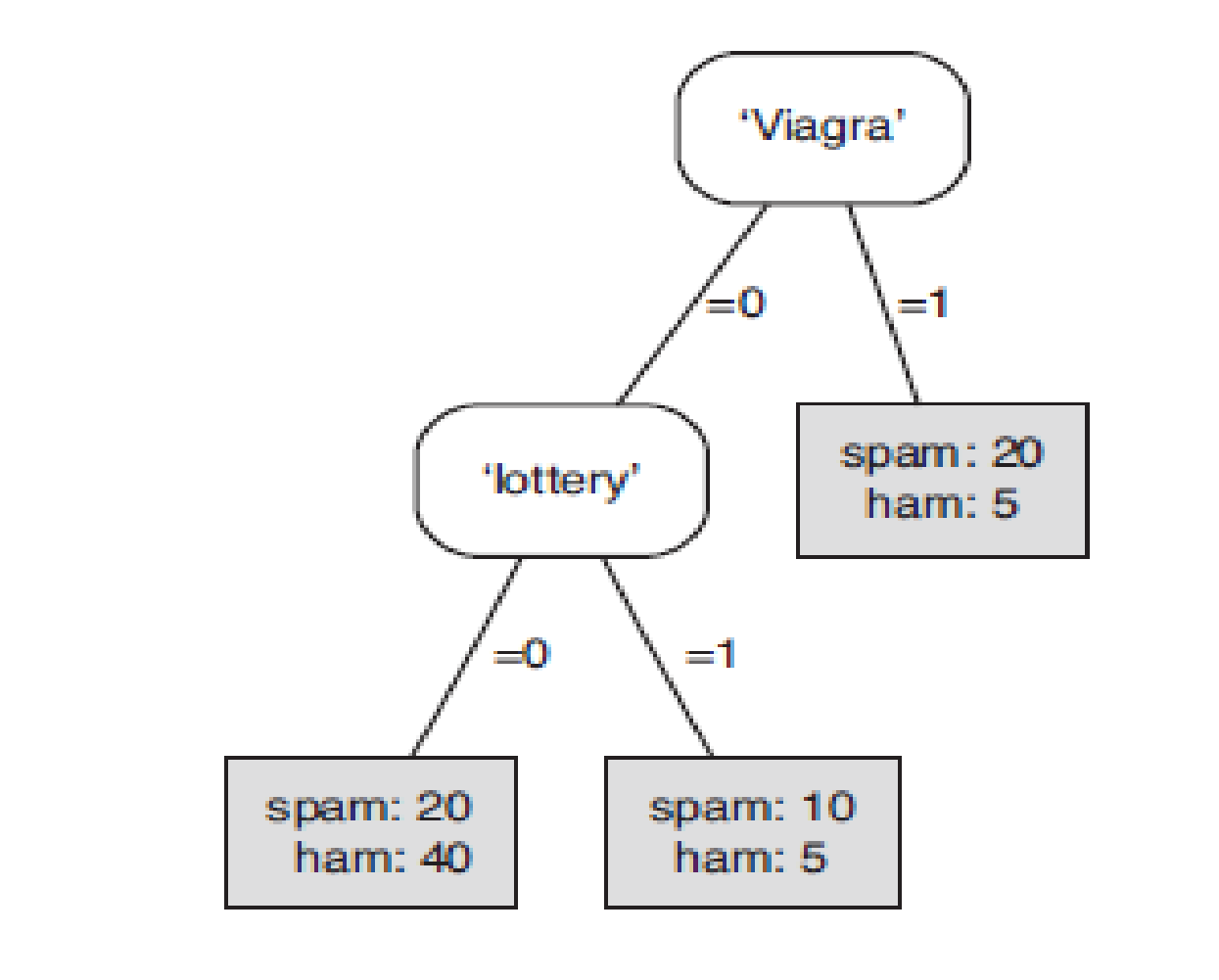


Fig 2. Feature tree showing class distribution (Flach, 2017)

Example (cont'd)

- The classifier in fig 2 works as follows:
- It looks for the presence of the word 'Viagra' in the email; if it finds it the email is immediately classified as spam.
- Otherwise it looks for the word 'lottery'; if it finds it then the email is classified as spam; otherwise it is ham.
- The 0 in the feature tree indicates a true (normal) state, while the 1 indicates a false (abnormal) state. This means the branches with 0 indicate ham, while those with 1 indicate spam.
- Let us follow the sequence from the top:
- The classifier looks for the word 'Viagra' when it finds it (the '1' branch) the emails are classified as spam. The remaining emails are passed to the alternate branch (0).
- The next thing it does is to look for the word 'lottery'; it if finds it the emails are passed to the branch labeled 1, these are spam.
- The remaining emails pass to the alternate branch labeled 0; these are ham.

Example (cont'd)

- The results can be interpreted as follows starting from the leftmost leaf to the rightmost leaf (they are only 3 leaves).
- Left leaf: 40 emails correctly labeled ham (True positive TP); 20 emails mislabeled (they should have been spam)(false positive FP)
- Middle leaf: 10 emails correctly labeled spam (True negative TN); 5 emails mislabeled (they were ham)(false negative FN)
- Right leaf: 20 emails correctly labeled spam (True negative TN); 5 emails mislabeled (they were ham) (FN false negative)
- Result:
- Spam: 30 (TN) out of 50 correctly classified (red font emails);
- Ham: 40 (TP) out of 50 correctly classified (black font emails).

Example (cont'd)

	Predicted ⊕	*Predicted* ⊖	
Actual ⊕	30	20	50
Actual ⊖	10	40	50
	40	60	100

Table 1. Contingency table/ confusion matrix (Flach, 2017)

Example (cont'd)

- Table 1 presents a contingency table/confusion matrix that is used in such scenarios to assess the performance of the classifier.
- The table shows the results in tabular form; the figures marked in blue circles (diagonal) represent the correctly classified emails.
- The figures in red (also diagonal) represent the wrongly classified emails.
- Based on the data we can find the accuracy of our classifier as follows:
- Accuracy = (total number of correctly predicted emails)/ (total number of emails)
- Accuracy = 70/100 = 70%; this also implies an error rate of 30%. (total being 100%)

Example (cont'd)

- Other parameters of interest that can also be determined from the table are:
- Precision = correct outputs (in our case all the correct spams)/ (total correct predictions (all correctly predicted spams and hams).
- => precision = 30/(30 + 40) = 30/70 = 43% or 0.43 (kind or low, right?)
- Recall = out of all positive cases (those correctly identified spam), how well did the model predict correctly?
- Recall = (total spams correctly predicted)/ (totals spams correctly predicted + total spams wrongly predicted)
- => recall = 30/(30 + 20) = 30/50 = 60% or 0.60
- ROC curve : also used to assess the performance of the classifier. The learner is encouraged to do some out of class reading on this important curve.

Part 3

Multiclass Classification

3.1 Introduction

- In binary classification the learner was faced with the task of distinguishing between two labels (classes). However, it is never that simple.
- Consider a scenario where the learner is faced with the task of classifying more than two labels.
- An example of such a scenario is the face recognition system on your phone; have you ever considered how many different labels are used to identify you? How does the learner sometimes even say that you are not you!
- Other examples include handwriting recognition systems, and even virus species classification systems.
- These systems which involve classification of more than two labels are referred to as multiclass systems.
- The beauty of machine learning at times lies in the systems you use on a daily basis that you find so helpful and useful, without giving much thought to the science behind their development.

Introduction (cont'd)

- It follows that in terms of probability a multiclass system does not use a Bernoulli probability distribution since the outcomes are more than two.
- In this case a multinoulli or categorical, distribution is used. This distribution is like an extended Bernoulli distribution to accommodate more than two outcomes (0 or 1).
- In a multinoulli distribution if there are k possible outcomes and x_i is a random variable that takes value 1 with the i-th outcome and 0 otherwise, then vector X defined as: $X = [x_1, x_2, \ldots, x_k]$ is a multinoulli vector.
- In machine learning this means that the model predicts the probability of an example belonging to each class label $[x_1, x_2, \ldots, x_k]$

Introduction (cont'd)

- Many of the algorithms used for binary classification can be used for multiclass classification. Examples include:
  - Gradient boosting
  - k – nearest neighbor
  - Random forest
  - Decision trees
  - Naïve bayes
- Further some multiclass algorithms can also be reduced to binary; these will be discussed later in this lesson.

3.2 Example

- In multiclass classification we have more than 2 outputs.
- Let us consider the situation whereby a patient is diagnosed with rheumatism.
- Naturally the doctor will want to perform further tests to confirm this diagnosis and also to know which type of rheumatism it is.
- This is a classic example of multiclass classification.
- As discussed earlier this is a multinoulli distribution whereby if there are k classes then a k by k contingency table will be required.
- Table 2 depicts such a scenario; 3 tests have been performed and we are interested in determining parameters such as accuracy for our learner.

3.2 Example (cont'd)

	Predicted			
	Class 1	Class 2	Class 3	
	15	2	3	20
Actual	7	15	8	30
	2	3	45	50
	24	20	56	100

Table 2. Contingency table where k = 3 (adapted from Flach, 2017)

3.2 Example (cont'd)

- The formulas used are the same as for the binary confusion matrix:
- Accuracy = (total number of correctly predicted outcomes)/ (total number of instances)
- => Accuracy = (15 + 15 + 45) / 100 = 75/ 100 = 75% or 0.75
- Precision for each class is also of interest:
- Precision = (correct outputs)/ (total outputs)
- For class 1: precision = 15/24 = 0.625 (or 62.5%)
- For class 2: precision = 15/20 = 0.75 (or 75%)
- For class 3: precision = 45/56 = 0.80 (or 80%)

3.2 Example (cont'd)

- We are also interested in recall for each class:
- Class 1: recall = 15/20 = 0.75
- Class 2: recall = 15/30 = 0.5
- Class 3: recall = 45/50 = 0.9
- To get the precision and recall for the whole classifier we only need to average the numbers:
- Precision = (0.625 + 0.75 + 0.8)/3 = 0.725 (or 72.5%)
- Recall = (0.75 + 0.5 + 0.9)/3 = 0.72

3.2 Example (cont'd)

- Thus we can generalize that for k classifiers,
- Precision = $\sum_{k=0}^{n} C_k/k$ where C_k is the precision for classifier k, and k = number of classifiers
- Recall = $\sum_{k=0}^{n} R_k/k$ where R_k is the recall for classifier k, and k = number of classifiers
- Another method that may be used is to take the weighted average for each classifier; this will take into account the weight of each classifier:
- Precision = (20/100) * 0.625 + (30/100) * 0.75 + (50/100) * 0.8 = 0.75
- The same weights can be used to calculate the recall.

3.3 Multiclass to Binary

- As mentioned in the introduction part of this lesson many of the binary algorithms are the same ones used to solve multiclass problems.
- There are different approaches to do this, but in all cases it is desirable to reduce the multiclass problem into several binary problems using some approach, in order for the problem to be solved using a binary classification algorithm. In essence this involves splitting the multiclass dataset into several binary datasets.
- There are two ways in which this is done in machine learning:
  - One – vs- rest
  - One – vs – one

3.3.1 One – vs – rest (OvR)

- This strategy is also known as one – vs- all (OvA).
- The multiclass problem is split into several binary datasets.
- A binary learner is then trained on each binary classification problem; the model that is most confident is then used for predictions.
- Let us pick an example to demonstrate this; in our earlier example we described a situation where a doctor would like to conduct tests to determine the type of rheumatism a patient has.
- Supposing that he would like tests to be done to determine among, rheumatoid arthritis, lupus and gout.

3.3.1 OvA (cont'd)

- In this setup 3 different experiments will have to be set up as follows:
- Problem 1: rheumatoid vs [lupus, gout]
- Problem 2: lupus vs [rheumatoid, gout]
- Problem 3: gout vs [rheumatoid, lupus]
- As can be seen we have 3 classes namely, rheumatoid arthritis, lupus and gout.
- This implies that for k classes we will have to develop k models; this is quite some work as the number of classes increase.
- With this method it is a requirement that each model should predict a class membership probability score.
- This will point to specific algorithms such as Logistic regression.

3.3.2 One – vs – one (OvO)

- This method also splits the multiclass dataset into several binary datasets.
- The difference comes in how the method splits them.
- In OvR we had one binary dataset for each class; in OvO we have one dataset for each class vs every other class.
- Let us demonstrate this by extending the example used in OvR; in this case we add an additional class ankylosing spondylitis (AS).
- Thus we now have 4 classes: rheumatoid arthritis (RA), lupus (L), gout (G) and AS.
- Consequently there will be 6 binary classification datasets as follows:

3.3.3 OvO (cont'd)

- We have: RA, L, G, AS
- Model 1: RA vs L
- Model 2: RA vs G
- Model 3: RA vs AS
- Model 4: L vs G
- Model 5: L vs AS
- Model 6: G vs AS
- This is a significantly higher number of models and by extension, datasets.

3.3.3 OvO (cont'd)

- The formula for calculating the number of models is given by $n(n-1)/2$, where n is the number of classes involved in the problem.
- In our example $n = 4$, since there are 4 classes.
- Applying the formula gives $4\ (3)/2 = 6$ models and datasets.
- Each model will then predict one label; the model with most predictions or votes is then predicted.
- This method is mostly used with support vector machine (SVM).

Part 4

Multilabel Classification

4.1 Introduction

- As the name implies multilabel classification refers to those problems involving two more labels for each example provided. Confusing, perhaps?
- Recall that in multiclass and binary class problems we had each example being classified with one label? That is to say an example can be classified as RA and not AS or L or G. (I hope this makes it simpler to understand).
- Have you ever used google photos? In a single photo several labels can be identified by the app….you, members of your family (if you had told the app who they are), bicycles, cars, etc. This is an application of multilabeling.
- The picture is the example, and in it, there are several labels to be identified.
- The model to be used in this type of scenarios is one that can predict multiple outputs; each output is predicted as a Bernoulli probability distribution.
- This means that with each example there are multiple binary classification predictions.

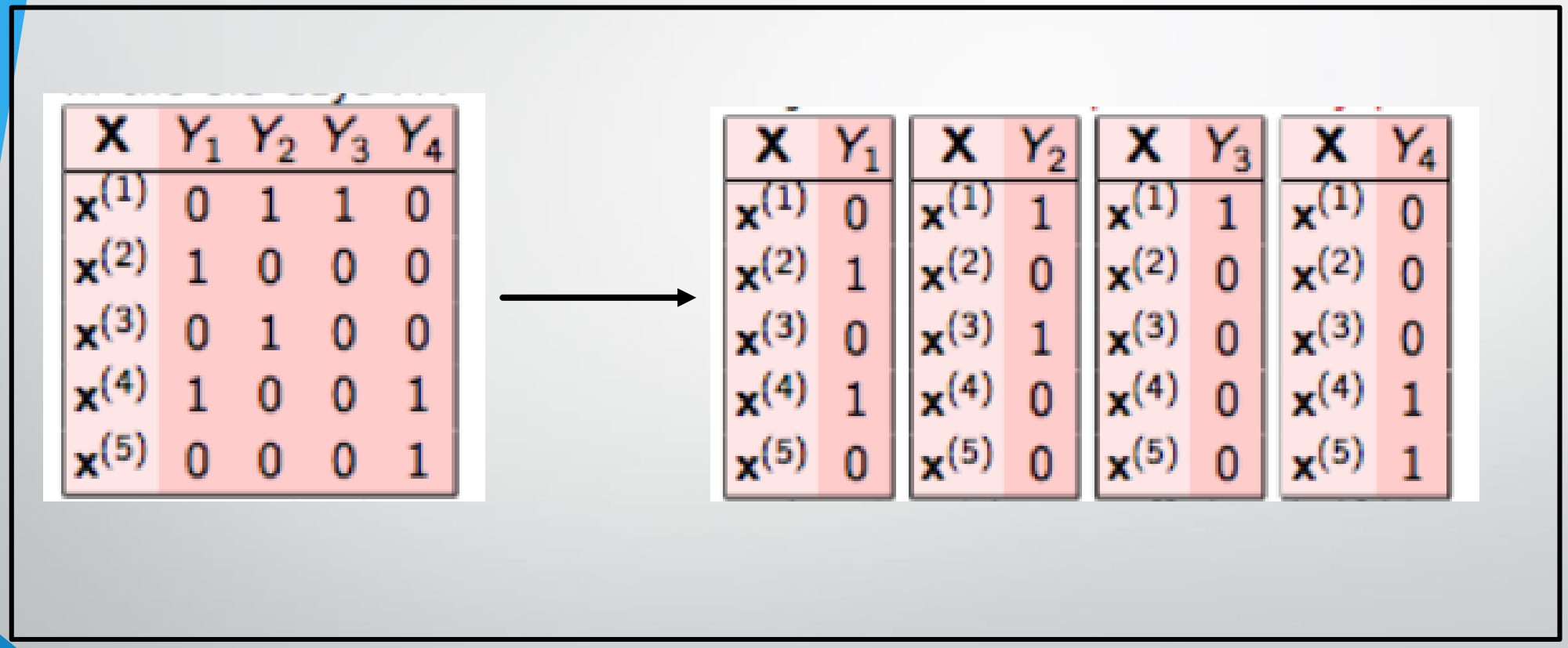
4.2 Techniques

- There are 3 techniques that are used to solve multilabel classification problems. These are :
  - Problem transformation
  - Adapted algorithm
  - Ensemble approaches
- Let us briefly discuss how these techniques work.

4.2.1 Problem transformation

- With this technique the problem is 'transformed' into several single label problems. This is achieved using 3 different approaches:
- Binary relevance – each label is treated as a single classification problem; thus with n target variables there will be n single classification problems. In fig 3 X is the independent feature and Y the target variable (n); thus in transformation we have 4 single classification problems. It can be seen that every target variable is treated independently; the overall accuracy can then be predicted using appropriate Python tools.
- Classifier chains – the formula used here is the same as for binary relevance but the implementation is different. The first learner is first trained on the input, next learner on the input and first label, then next learner on input, first and second label, and so on. Thus we do see that we wind up with the same notation; for n labels we have (n -1) classifiers. This is why the chain notation is used. In fig 4 we have the input data (X) and the chain progresses for each label till all labels are exhausted.

a. Binary relevance



X	Y_1	Y_2	Y_3	Y_4
$x^{(1)}$	0	1	1	0
$x^{(2)}$	1	0	0	0
$x^{(3)}$	0	1	0	0
$x^{(4)}$	1	0	0	1
$x^{(5)}$	0	0	0	1

X	Y_1
$x^{(1)}$	0
$x^{(2)}$	1
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	0

X	Y_2
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	1
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_3
$x^{(1)}$	1
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	0
$x^{(5)}$	0

X	Y_4
$x^{(1)}$	0
$x^{(2)}$	0
$x^{(3)}$	0
$x^{(4)}$	1
$x^{(5)}$	1

Fig 3. Binary transformation (adapted from analyticsvidhya.com)

b. Classifier chains

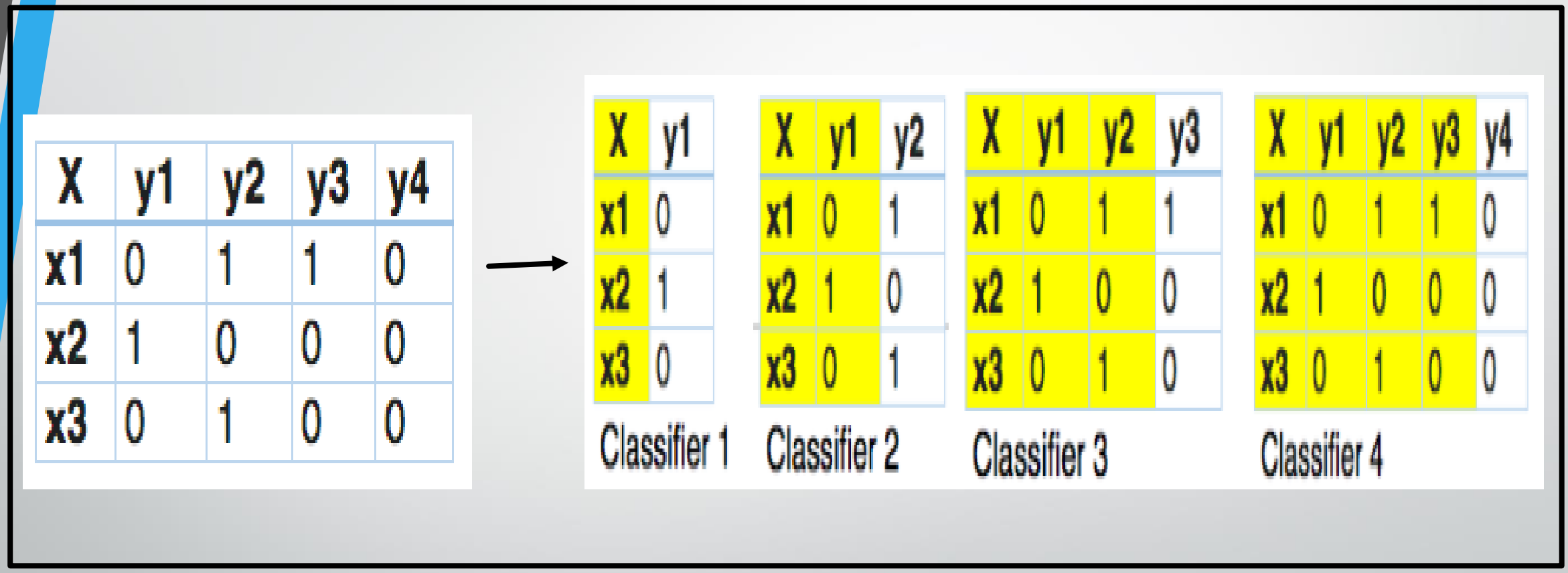


Fig 4. Classifier chains (adapted from analyticsvidhya.com)

4.2.1 Problem transformation (cont'd)

- Label power set – ever heard of the term power combination in the corporate world or in politics? It refers to the combining of like minded politicians or corporate players who are perceived as a winning combination. The same applies here, literally. Like terms are combined; similar label sets are combined to form a single multiclass problem.
- Fig 5 demonstrates this aptly. X1/x4, and x3/x6 share the same labels. They are thus combined/ transformed to form a single multiclass problem (right).

c. Label power set

X	y1	y2	y3	y4
x1	0	1	1	0
x2	1	0	0	0
x3	0	1	0	0
x4	0	1	1	0
x5	1	1	1	1
x6	0	1	0	0

X	y1
x1	1
x2	2
x3	3
x4	1
x5	4
x6	3

Fig 5. Label power set (adapted from analyticsvidhya.com)

4.2.2 Adapted algorithm

- As the name implies this technique involves adapting an algorithm to perform multilabel classification.
- Multilabel versions usually have ML signifying it's a multilabel adaptation.
- For example multilabel kNN will be denoted as MLkNN.
- Other implementations include: ML decision trees, ML random forests, ML gradient boosting, and so on.
- The implementations are normally done with Python.

4.2.3 Ensemble techniques

- This involves using ensemble classification functions.
- These are easily found in scikit – multilearn library in Python.

4.3 Application areas

- Multilabel classification is applied in so many areas in our day to day life that we normally see it as just "cool"; the process behind as you can see, is not really that simple:
- When you visit a music site you find a song can be categorized using multilabels, for example, "slow, relaxing", "pop, dance" and so on. In some car stereos the app will do this for you.
- In photos, the scenery can receive different labels identifying various aspects of the photo.
- In bioinformatics it is used to identify different genes in a data set.
- How about google news? Different articles are categorized under different headings; you can check it out for yourself. An article may fall in more than one category; there is a lot of research work going on in text categorization.
- Now that you know, keep your eyes open for other areas you can identify multilabel classification.

Part 5

Imbalanced Classification

5.1 What are they?

- This type of classification involves an instance space where the examples are not equally distributed.
- Typically the tasks here involve a scenario where the normal case is in the majority while the abnormal case is in the minority.
- These types of problems are modeled as binary classification tasks and require adaptations of known algorithms.
- Moreover, special methods have to be used to bring some balance in the training set between the examples in the majority (the normal case) and those in the minority (the abnormal case).
- The idea is to undersample the majority and oversample the minority. Methods that do this include random undersampling and smote oversampling.

5.2 Algorithms

- Examples of algorithms that have been adapted to be used in these scenarios include:
  - Cost – sensitive logistic regression
  - Cost – sensitive decision trees
  - Cost – sensitive support vector machines.

5.3 Application areas

- This class of problems will be found in the following areas:
- Fraud detection – there is currently an uproar over the vulnerability of the mobile app in one of our country's biggest banks. Imbalanced classification can be used in this area to detect fraud.
- Medical diagnostic tests – it is not the norm that a patient will test positive for a condition, especially those rare diseases; imbalanced classification is applicable in this scenario.

Summary

- Classification tasks can be further subdivided into four distinct groups: binary classification, multiclass classification, multilabel classification, and imbalanced classification.
- In binary classification each of the classes has to be assigned a label; the normal case is assigned a value 0, while the abnormal case is assigned the value 1.
- Systems which involve classification of more than two labels are referred to as multiclass systems.
- Reducing a multiclass problem to a binary problem is done using one vs all (OvA a.k.a. OvR) or one versus one method.
- Multilabel classification refers to those problems involving two more labels for each example provided; they are solved using 3 techniques: Problem transformation, adapted algorithm and Ensemble approaches.
- Imbalanced classification involves an instance space where the examples are not equally distributed.

References

- Brownlee, J. (2020, August 19). *4 types of classification tasks in machine learning*. Machine Learning Mastery. Retrieved June 2, 2022, from https://machinelearningmastery.com/types-of-classification-in-machine-learning/
- Brownlee, J. (2021, April 26). *One-vs-rest and one-vs-one for multi-class classification*. Machine Learning Mastery. Retrieved June 2, 2022, from https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/
- Flach, P. A. (2017). *Machine learning: The art and science of algorithms that make sense of data*. Cambridge University Press.
- Glen, S. (2022, January 12). *Bernoulli distribution: Definition and examples*. Statistics How To. Retrieved June 2, 2022, from https://www.statisticshowto.com/bernoulli-distribution/
- Jain, S. (2020, December 23). *Multi label classification: Solving Multi Label Classification problems*. Analytics Vidhya. Retrieved June 2, 2022, from https://www.analyticsvidhya.com/blog/2017/08/introduction-to-multi-label-classification/
- JavaTpoint. (n.d.). *Confusion matrix in machine learning - javatpoint*. www.javatpoint.com. Retrieved June 2, 2022, from https://www.javatpoint.com/confusion-matrix-in-machine-learning