We acknowledge the Australian Aboriginal and Torres Strait Islander peoples as the traditional owners of the lands and waters where we live and work.

Week 3: Introduction to Data Mining

Dr Max Cao
Performance and Insights Manager at Brisbane City Council & Lecturer at James Cook University

Discussion Topics

- Logistics Regression
- Confusion Matrix
- ROC & AUC
- KNN

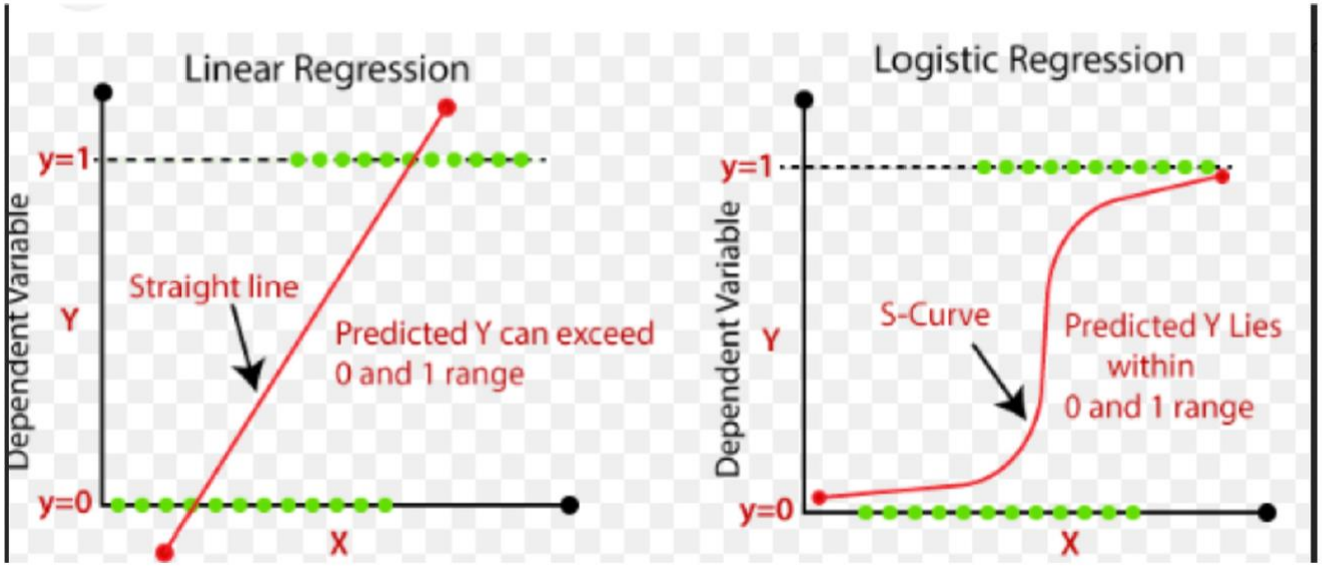
Why not linear regression for classification?

Consider that we got to predict the medical condition of a patient in the emergency room on the basis of her/his symptoms. In this simplified example, there are three possible diagnoses: stroke, drug overdose, and epileptic seizure. We could consider encoding these values as a quantitative response variable, Y , as follows:

Y =1 if stroke;2 if drug overdose;3 if epileptic seizure. Using this coding, least squares could be used to fit a linear regression model to predict Y on the basis of a set of predictors X1,...,Xp. Unfortunately, this coding implies an ordering on the outcomes, putting drug overdose inbetween stroke and epileptic seizure, and insisting that the difference between stroke and drug overdose is the same as the difference between drug overdose and epileptic seizure.

In practice there is no reason that this needs to be the case. For instance, one could choose an equally reasonable coding, Y =1 if epileptic seizure;2 if stroke; 3 if drug overdose, which would imply a totally different relationship among the three conditions. Each of these codings would produce fundamentally different linear models that would ultimately lead to different sets of predictions on test observations. For a binary (two level) qualitative response, the situation is better, however, it has it owns cons and those will be discussed in the next slides. **Refer textbook 4.2 for more details.**

Linear Vs Logistic Regression



Logistic Regression

Logistic Regression Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary outcome variable(Y(0/1)) based on linear regression formula.

Logistic regression (or logit regression),the logit function is the natural log of the odds that Y equals one of the categories. For mathematical simplicity, we're going to assume Y has only two categories and code them as 0 and 1(binary). It can be used for multiple outcome variables.

Assumptions:

1.The logistic regression assumes that there is minimal or no multicollinearity among the explanatory variables.

2.The Logistic regression assumes that the explanatory variables are linearly related to the log of odds.

3.The Logistic regression assumes the observations to be independent of each other.

Logistic Regression

Log Odds and the Logit Function

Odds are likelihood ratios, and tell us how likely it is that something particular will happen. The odds ratio is the probability of success/probability of failure. As an equation, that's P(A)/P(-A), where P(A) is the probability of A, and P(-A) the probability of 'not A' (i.e. the complement of A).

Taking the logarithm of the odds ratio gives us the log odds of A, which can be written as $\log(A) = \log(P(A)/P(-A))$,

Since the probability of an event happening, P(-A) is equal to the probability of an event not happening, 1 – P(A), we can write the log odds as

$\log [p/(1-p)]$

Where:

p = the probability of an event happening

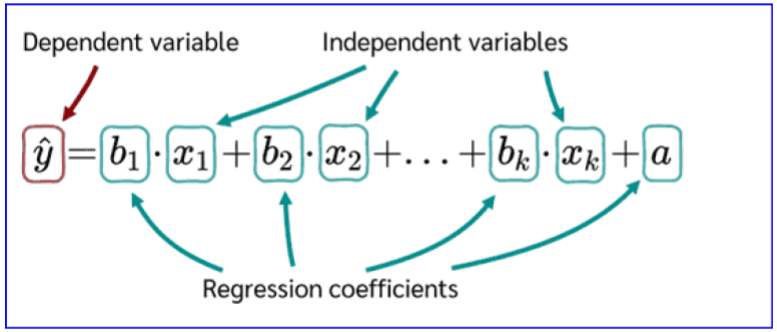
1 – p = the probability of an event not happening

When a function's variable represents a probability, p (as in the function above), it's called the logit function .

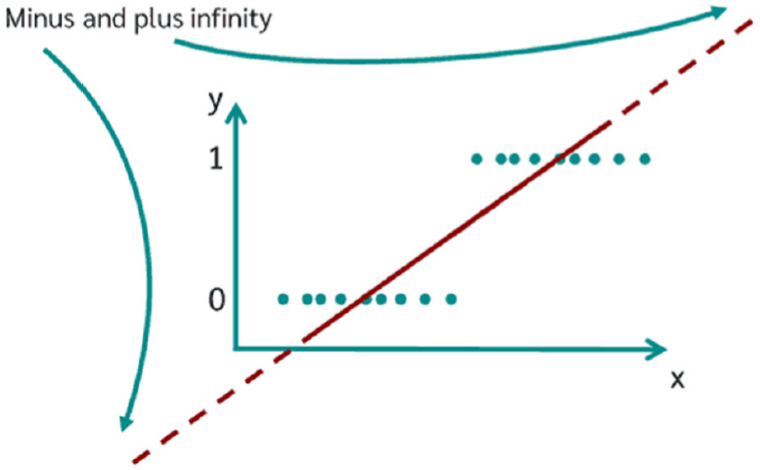
Logistic Regression

Logistic Regression – Statistics behind the model

To build a logistic regression model, the linear regression equation is used as the starting point.



However, if a linear regression were simply calculated for solving a logistic regression, the following result would appear graphically.



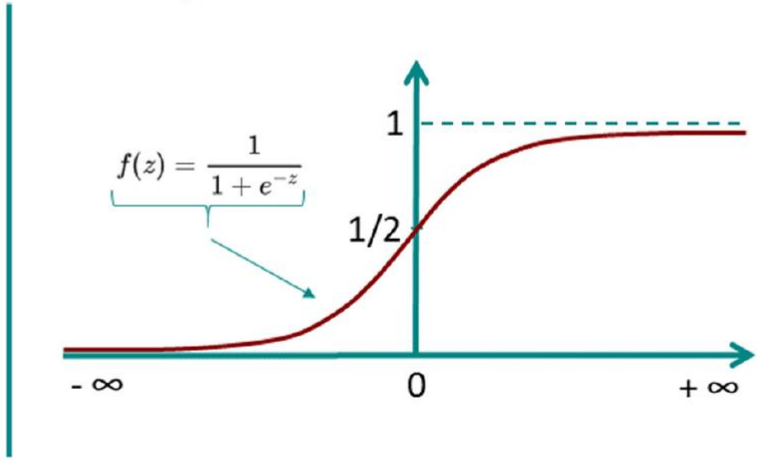
Logistic Regression

As can be seen in the graph in the previous slide, **values between plus and minus infinity** can now occur.

The goal of logistic regression, however, is to estimate the probability of occurrence and not the value of the variable itself. Therefore, this equation must still be transformed. To do this, it is necessary to restrict the value range for the prediction to the range between 0 and 1. To ensure that only values between 0 and 1 are possible, the **logistic function *f*** is used.

Logistic function

The logistic model is based on the logical function. The special thing about the logistic function is that for values between minus and plus infinity, it always assumes only values between 0 and 1(S-shaped curve).



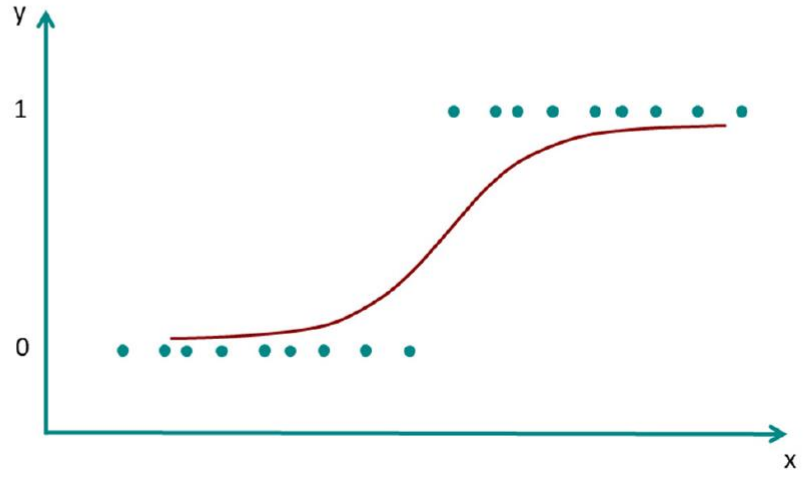
Logistic Regression

So, the logistic function is perfect to describe the **probability *P(y=1)***. If the logistic function is now applied to the upper regression equation the result is

$$\hat{y} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

$$f(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}$$

This now ensures that no matter in which range the x values are located, only numbers between 0 and 1 will come out. The new graph now looks like this:



Logistic Regression

Explanatory variables Outcome variable

The probability that for given values of the explanatory variables, the dichotomous outcome variable y is 0 or 1 is given by

$$P(y=1|x_1,\ldots,x_n)=\frac{1}{1+e^{-(b_1\cdot x_1+\ldots+b_k\cdot x_k+a)}}$$

$$P(y=0|x_1,\ldots,x_n)=1-\frac{1}{1+e^{-(b_1\cdot x_1+\ldots+b_k\cdot x_k+a)}}$$

To calculate the probability of a person being sick or not using the logistic regression for the example above, the model parameters b_1, b_2, b_3 and a must first be determined. Once these have been determined, the equation for the example above is

$$P(\text{Diseased})=\frac{1}{1+e^{-(b_1\text{Age}+b_2\text{Gender}+b_3\text{Smoking status}+a)}}$$

Logistic Regression

Interpretation of the results

The relationship between outcome and explanatory variables in logistic regression is not linear. Therefore, the regression coefficients cannot be interpreted in the same way as in linear regression.

For this reason, odds are interpreted in logistic regression. The odds are calculated by relating the two probabilities that y is "1" and that y is "not 1“**Where:**

•p = the probability of an event happening

•1 – p = the probability of an event not happening

$$odds = \frac{p}{1-p}$$

This quotient can take any positive value. If this value is now logarithmized, values between minus and plus are infinitely possible

$$z = Logit = ln(\frac{p}{1-p})$$

These logarithmic odds are usually referred to as "logits".

Chi-square test tells, whether the model is significant overall or not. The null hypothesis is that both models are the same. If the p-value is less than 0.05, this null hypothesis is rejected, and Ha will be true.

Logistic Regression

Maximum Likelihood Method

To determine the model parameters for the **logistic regression equation**, the **Maximum Likelihood Method** is applied. The maximum likelihood method is one of several methods used in statistics to estimate the parameters of a mathematical model. Another well-known estimator is the least squares method, which is used in linear regression. Read 4 -4.5 in the textbook for better understanding.

The Likelihood Function

To understand the **maximum likelihood method**, we introduce the **likelihood function** *L*. *L* is a function of the unknown parameters in the model, in case of logistic regression these are $b_1,\ldots b_n$, *a which are our coefficient parameters for input x, and* an additional coefficient that provides the intercept or bias. Therefore, we can can also write $L(b_1,\ldots b_n, a)$ or $L(\theta)$ if the parameters are summarized in θ.

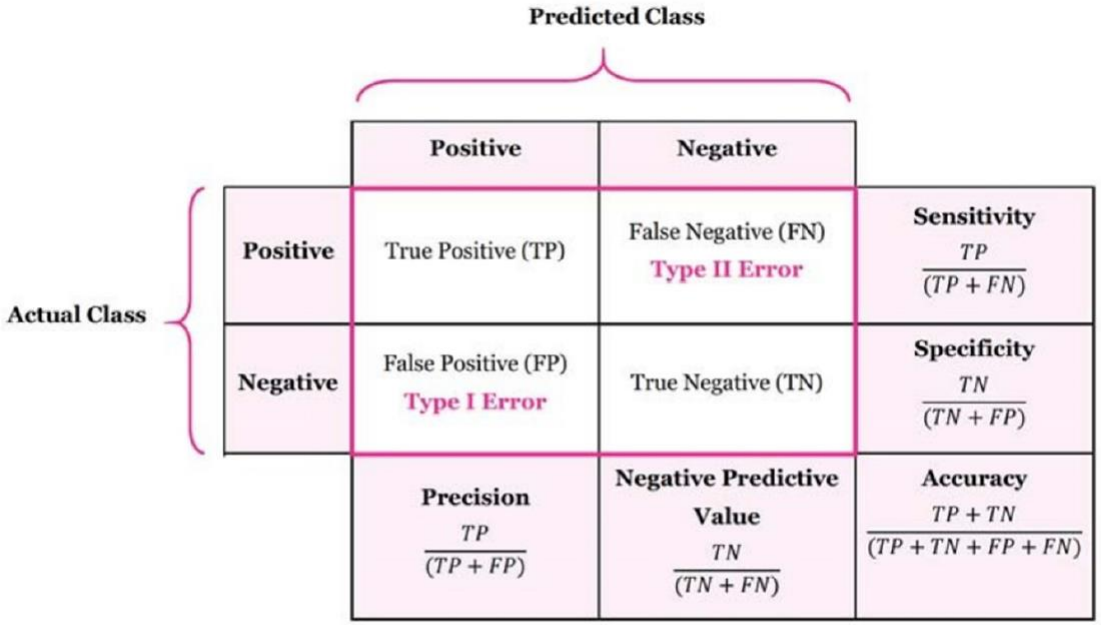
$L(\theta)$ now indicates how probable it is that the observed data occur. With the change of θ, the probability that the data will occur as observed changes.

The Maximum Likelihood Estimator can be applied to the estimation of complex nonlinear as well as linear models. In case of logistic regression, the goal is to estimate the parameters $b_1,\ldots b_n$, a, which maximize the so-called log likelihood function $LL(\theta)$.

Multinomial logistic regression

If the outcome variable has two characteristics (e.g. male, female), i.e. is dichotomous, **binary logistic regression** is used. However, outcome variable has more than two instances, e.g. which mobility concept describes a person's journey to work (car, public transport, bicycle), **multinomial logistic regression** must be used.

Confusion Matrix- Model Evaluation



Key Factors to lookout:
Sensitivity, Specificity

Model Evaluation-Confusion Matrix

Performance metrics of an algorithm are accuracy, precision, recall, and F1 score, which are calculated on the basis of TP, TN, FP, and FN.

Accuracy of an algorithm is represented as the ratio of correctly classified patients (TP+TN) to the total number of patients (TP+TN+FP+FN).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Precision of an algorithm is represented as the ratio of correctly classified patients with the disease (*TP*) to the total patients predicted to have the disease (*TP*+*FP*).

$$Precision = \frac{TP}{TP + FP}$$

Recall metric is defined as the ratio of correctly classified diseased patients (*TP*) divided by total number of patients who have actually the disease.

$$\text{Recall} = \frac{TP}{TP + FN}$$

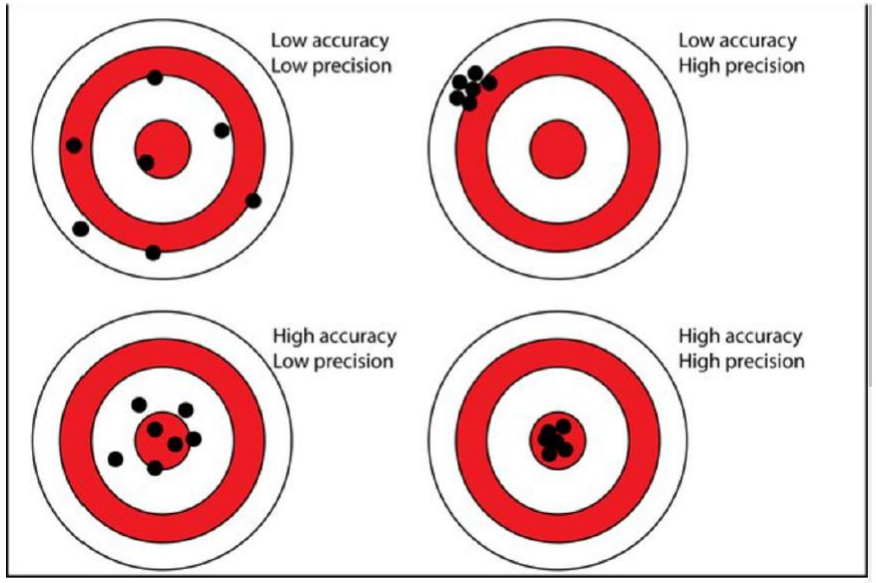
The perception behind recall is how many patients have been classified as having the disease. Recall is also called as sensitivity.**F1 score** is also known as the F Measure. The F1 score states the equilibrium between the precision and the recall.

$$F1Score = \frac{2 * precision * recall}{precision + recall}$$

Model Evaluation-Confusion Matrix

Accuracy(TP+TN rate) and Precision(TP+ FP rate)

Comparison	Accuracy	Precision
Definition	The extent to which readings are close to the real value	The extent to which the measured readings are close
Measure of	Statistical bias	Statistical variability
Degree of	Conformity	Reproducibility
Dependency	Dependent on precision	Independent of accuracy
Origin	Systematic errors	Random errors



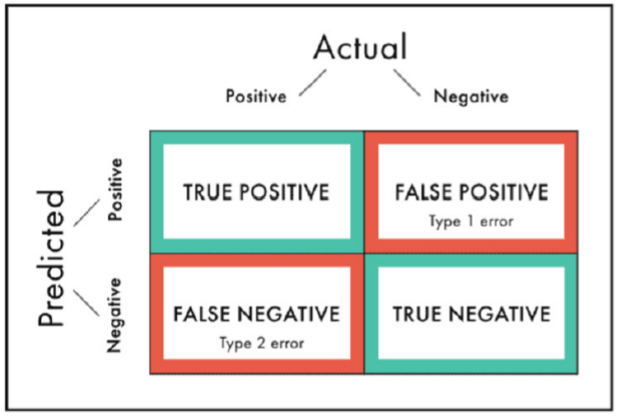
Model Evaluation-Confusion Matrix

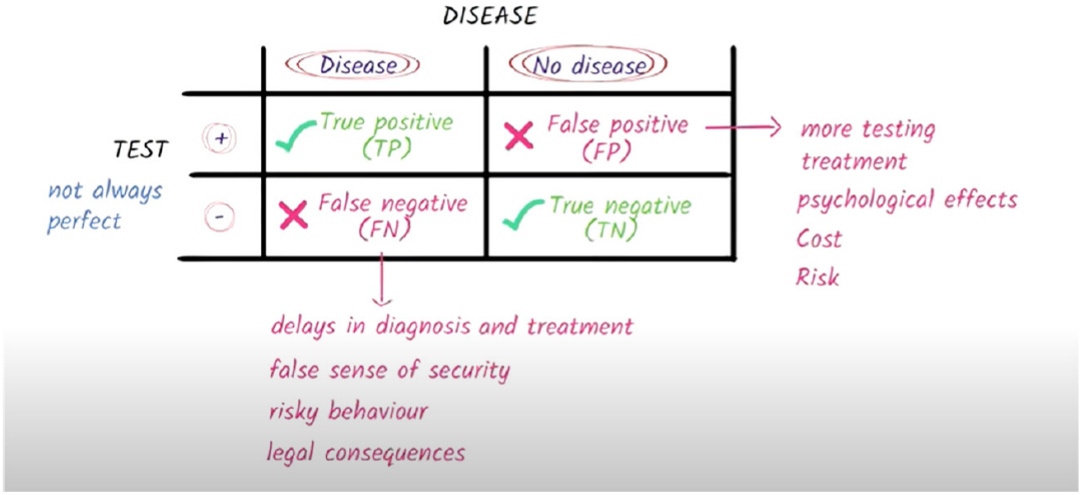
Why we need Confusion Matrix, though the model has good accuracy rates? Any link to the context of the problem/hypothesis?

For example – Lets consider these scenarios,

1) Incase of finding a person guilty or not guilty – TN,FP rates are important as the legal team's focus would on not punishing an innocent.
2) In case of identifying a chronic disease like lung cancer/the recent covid pandemic- the goal would be on the TP and FN rates, as the patients with actual disease needs to treated quickly and FN rates pose a greater risk of delay in treatments of the actual patients, worsening of the physical condition and even legal consequences. FP rates also poses a great deal of risk with the normal people being treated for the disease, unnecessary psychological trauma and again may lead to legal consequences.
3) For banking , the usually the focus is on defaulters than non-defaulters.TP/TN?

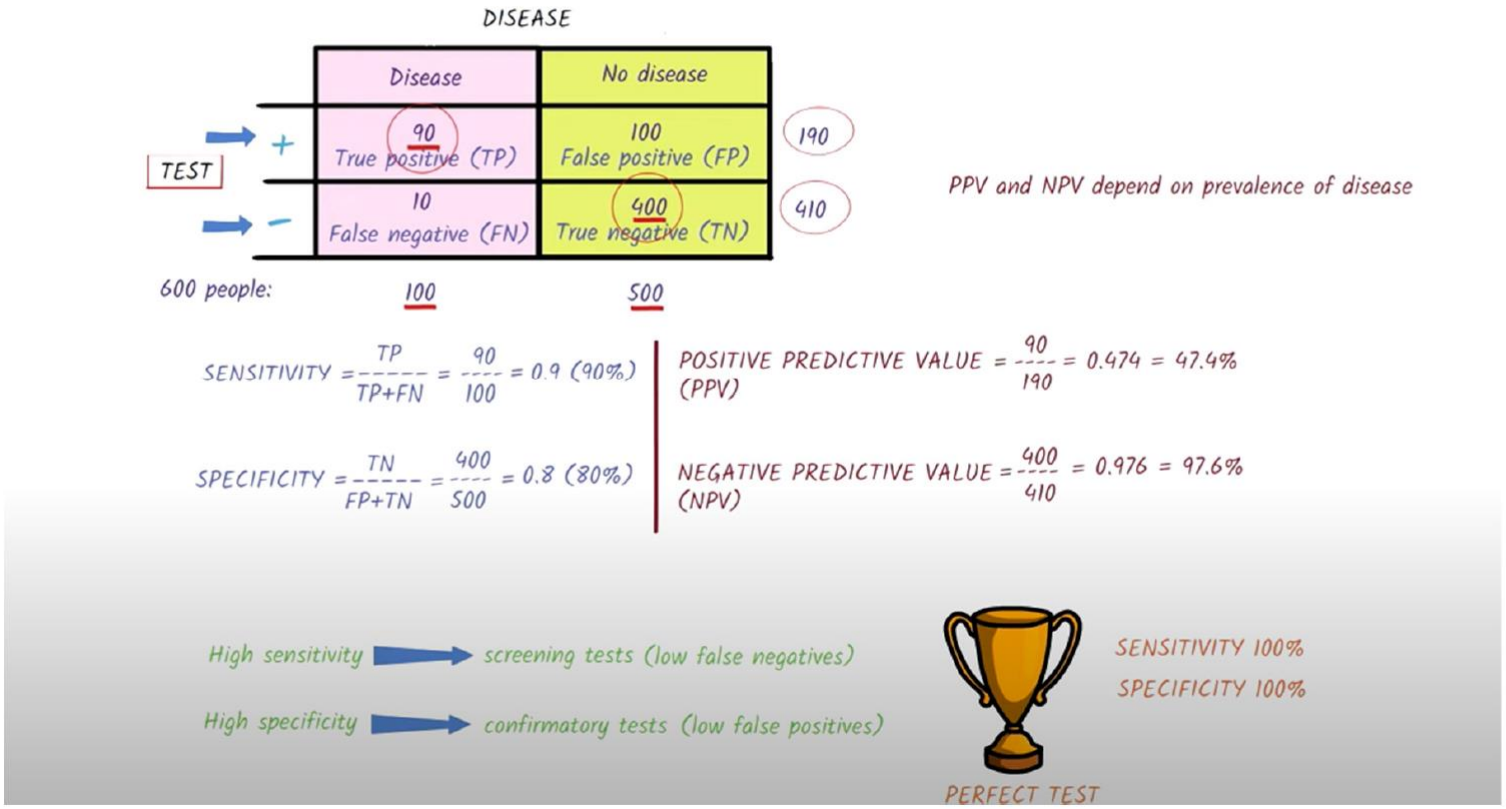
Model Evaluation-Confusion Matrix





- The rows in the confusion matrix corresponds to the predicted values by the machine learning algorithm.
- The columns corresponds to the known truth or the actual values.

Model Evaluation-Confusion Matrix



Model Evaluation-Confusion Matrix

Confusion Matrix example for multi-classification problem- Between 3 movies "Troll2, Gore Police and Cool as Ice" . Green ones are correctly predicted values and the ones in red are the incorrect predictions by the machine learning algorithm.

		Actual		
		Troll 2	Gore Police	Cool as Ice
Predicted	Troll 2	12	102	93
	Gore Police	112	23	77
	Cool as Ice	83	92	17

ROC-Receiver operating curve

ROC or Receiver Operating Characteristic curve represents a probability graph to show the performance of a classification model at different threshold levels. The curve is plotted between two parameters, which are:

- **True Positive Rate or TPR**
- **False Positive Rate or FPR**

In the curve, TPR is plotted on Y-axis,

whereas FPR is on the X-axis.

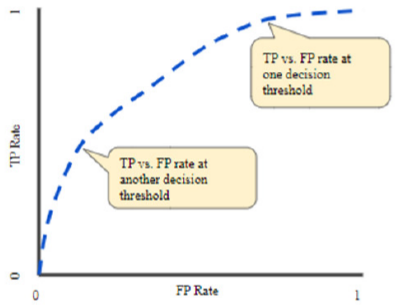
True Positive Rate (**TPR**) is a synonym for recall and is therefore defined as follows:

$$TPR = \frac{TP}{TP + FN}$$

False Positive Rate (**FPR**) is defined as follows:

$$FPR = \frac{FP}{FP + TN}$$

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.



To compute the points in an ROC curve, we could evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Sorting algorithm that can provide this information for us, called AUC is considered efficient.

ROC-Receiver operating curve

AUC: Area Under the ROC Curve

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).

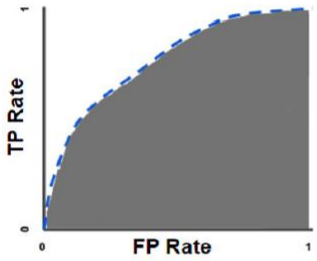
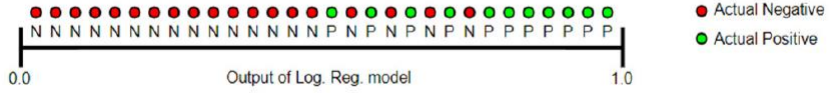


Figure 5. AUC (Area under the ROC Curve).

AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. For example, given the following examples, which are arranged from left to right in ascending order of logistic regression predictions:



Predictions ranked in ascending order of logistic regression score.

AUC represents the probability that a random positive (green) example is positioned to the right of a random negative (red) example.

AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

AUC is desirable for the following two reasons:

- AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
- AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

However, both these reasons come with caveats, which may limit the usefulness of AUC in certain use cases:

- **Scale invariance is not always desirable.** For example, sometimes we really do need well calibrated probability outputs, and AUC won't tell us about that.
- **Classification-threshold invariance is not always desirable.** In cases where there are wide disparities in the cost of false negatives vs. false positives, it may be critical to minimize one type of classification error. For example, when doing email spam detection, you likely want to prioritize minimizing false positives (even if that results in a significant increase of false negatives). AUC isn't a useful metric for this type of optimization.

ROC-Receiver operating curve

When to Use AUC-ROC

AUC is preferred due to the following cases:

•AUC is used to measure how well the predictions are ranked instead of giving their absolute values. Hence, we can say AUC is **Scale-Invariant.**

•It measures the quality of predictions of the model without considering the selected classification threshold. It means AUC is **classification-threshold-invariant.**

When not to use AUC-ROC

•AUC is not preferable when we need to calibrate probability output.

•Further, AUC is not a useful metric when there are wide disparities in the cost of false negatives vs false positives, and it is difficult to minimize one type of classification error.

How AUC-ROC curve can be used for the Multi-class Model?

Although the AUC-ROC curve is only used for binary classification problems, we can also use it for multiclass classification problems. For multi-class classification problems, we can plot N number of AUC curves for N number of classes with the One vs ALL method.

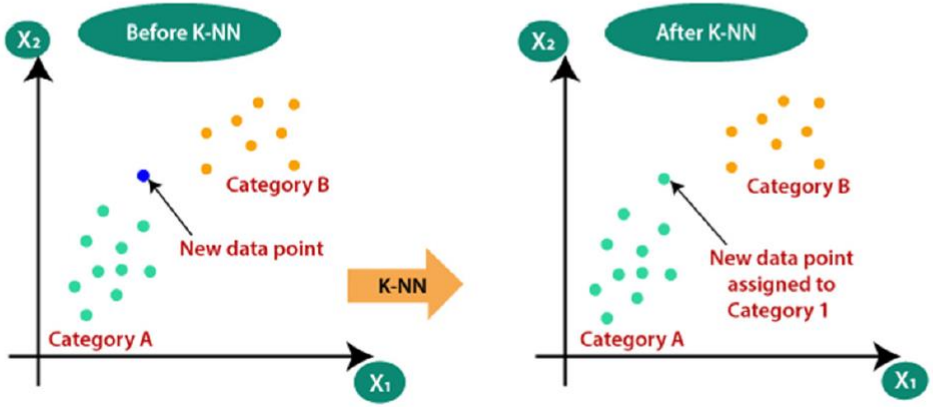
For example, if we have three different classes, X, Y, and Z, then we can plot a curve for X against Y & Z, a second plot for Y against X & Z, and the third plot for Z against Y and X.

K-Nearest Neighbor(KNN)

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories based on the **distance metrics.**
- K-NN algorithm stores all the available data and classifies a new data point based on the **similarity.** This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.K-NN is a **non-parametric algorithm**, which means it does not make any significant assumption on underlying data and is capable of estimating the events of any form.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

K-Nearest Neighbor(KNN)

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



K-Nearest Neighbor(KNN)

The K-NN working can be explained on the basis of the below algorithm:

•**Step-1:** Select the number K of the neighbors

•**Step-2:** Calculate the Euclidean distance(in this example-K is integer) of **K number of neighbors.** It is the square root of the sum of squares of differences between corresponding elements.

•**Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance. **Step-4:** Among these k neighbors, count the number of the data points in each category.

•**Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.

Explore and understand the different distance metrics and choose the distance metric relevant to the dataset.

Choosing distance metric:
The distance metric is the function that measures how similar or dissimilar two instances are. It affects how the algorithm defines the neighborhood of a new instance. There are many distance metrics to choose from, such as Euclidean, Manhattan, Minkowski, Hamming, Cosine, etcetera.

The choice of distance metric depends on the type and scale of the features, as well as the domain knowledge and assumptions of the problem. For example, **"Euclidean"** distance is a good choice for continuous features**(numeric)** with similar scales, while **"Hamming"** distance is suitable for categorical features(**non-numeric).**

You can also use cross-validation to evaluate the performance of different distance metrics on your data and select the one that works best for your model.

KNN – Distance Functions

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Hamming Distance

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

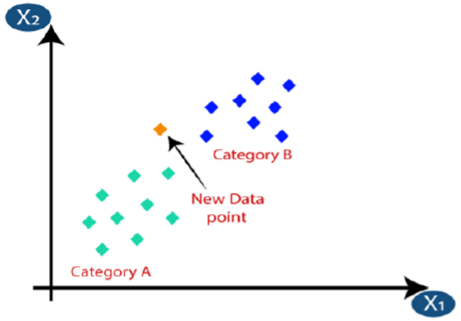
$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

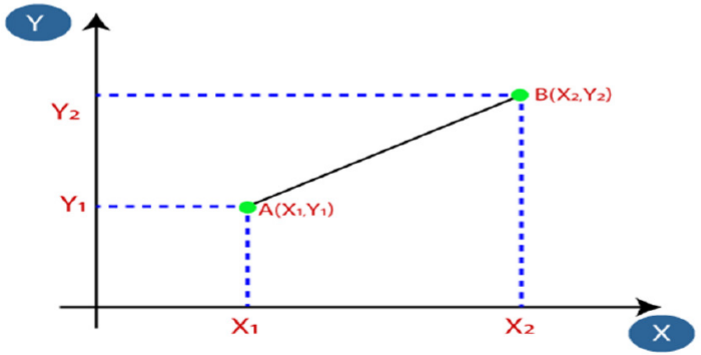
X	Y	Distance
Male	Male	0
Male	Female	1

K-Nearest Neighbor(KNN)

Example-we have a new data point, and we need to put it in the required category.



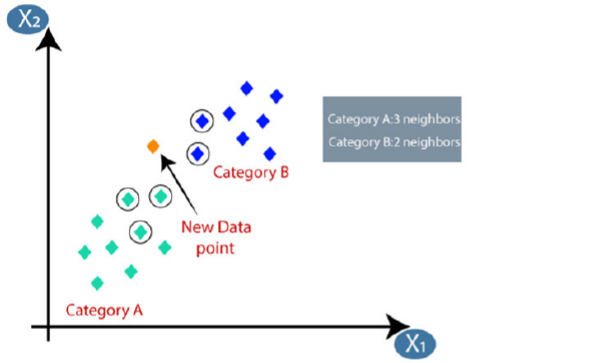
- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

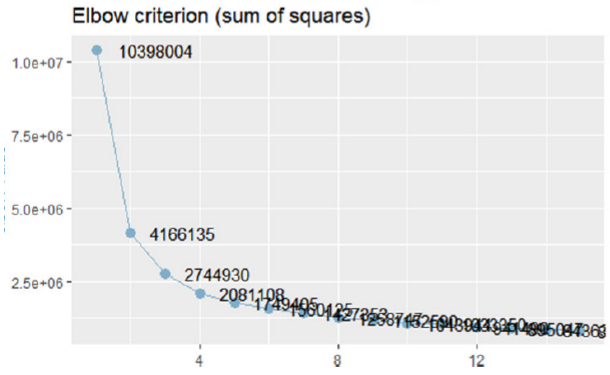


Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

K-Nearest Neighbor(KNN)

By calculating the Euclidean distance, we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image, as we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.





The k value in the k-NN algorithm defines how many neighbors will be checked to determine the classification of a specific query point. For example, if k=1, the instance will be assigned to the same class as its single nearest neighbor.

How to select the value of K in the K-NN Algorithm?

- We can use sum of squares method to find the optimal K ("Elbow Method). The bent is at 2 , so the optimal K value is 2 as an example –Refer figure Elbow Criterion.
- Generally used value for K is 5.
- A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
- Large values for K are good, but it may find some difficulties. Try multiple iterations

References

Textbook- An introduction to Statistical Learning with Application in R- Authors Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani .

Textbook online reference -https://www.stat.berkeley.edu/users/rabbee/s154/ISLR_First_Printing.pdf

https://datatab.net

https://javapoint.com

https://statsquest.com

https://towardsdatascience.com/