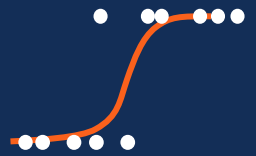
Classification – Fundamentals & Practical Applications

Course Learning Objectives

Understand what Classification is and its applicability to many real-world scenarios



Perform simple classification tasks using logistic regression in Excel

Understand the implicit assumptions behind Classification techniques and algorithms

Create classification models in Python using statsmodels and sklearn modules

Interpret and evaluate the performance of classification models, outputs and parameters

Explore more advanced evaluation techniques such as PDP plots and SHAP values to expand your horizons.

Classification Basics

Machine Learning Use Cases

Machine Learning can be used across a wide variety of tasks in Finance.

- The target categories should be **discrete variables**
- Predictions are made using one or more **input variables**

Recap: The Machine Learning Ecosystem

There are four main types of **Machine Learning** algorithms:

Supervised Machine Learning

...uses **labelled** datasets to train algorithms in classifying data

- Classification
- Regression

Unsupervised Machine Learning

...uses **unlabelled datasets** to train algorithms in classifying data

- Clustering
- Association

Reinforcement Learning

...uses reward maximization such that the algorithm determines the optimal behaviour in an environment

Deep Learning

...learns data patterns and structure from the data itself and is scalable to big data. It can be used for supervised and unsupervised learning

Types of Classification

Binary

- Classification tasks that have *two class labels*
- Outcomes must be **ONE** of the two classes
- Algorithm that only deals with binary classification include Logistic Regression and Support Vector Machines

Use Case	Input Variables	Output Classes
Tumour diagnosis	Variation, Texture, Contrast, Growth Rate etc	**Malignant** OR **Benign**
Customer Prediction	Purchase History, Click history, Customer Profile	**Will Buy** OR **Will Not Buy**
Email Spam Detection	Spelling Errors, Grammatical Errors, Email domain	**Spam** OR **Not Spam**

Types of Classification

Multi-Class

- Has *more than two class labels*
- Outcomes must be ONE of a range of classes
- Algorithm suited for multi-class problems include decision trees and random forests.

Use Case	Input Variables	Output Classes
Tumour diagnosis	Variation, Texture, Contrast, Growth Rate etc	**Malignant** OR **Benign** OR **Premalignant**
Customer Prediction	Purchase History, Click history, Customer Profile	**Will Buy** OR **Will Not Buy** OR **Insufficient Data**
Email Spam Detection	Spelling Errors, Grammatical Errors, Email domain	**Spam** OR **Not Spam** OR **Unsafe**

Types of Classification

Multi-label

- Has **two or more** class labels
- Outcome can be ***ONE or MORE*** of the class labels
- Difference between multi-label and multi-class is that each label in the former one represents a **different** but **related** classification problem

For example, a multi-label classifier may classify an email as both spam and unsafe, or classify the tumor as both benign and premalignant

Use Case	Input Variables	Output Classes (Labels)
Tumour diagnosis	Variation, Texture, Contrast, Growth Rate etc	**Malignant** OR /AND **Benign** OR / AND **Premalignant**
Customer Prediction	Purchase History, Click history, Customer Profile	**Will Buy** OR / AND **Will Not Buy** OR / AND **Insufficient Data**
Email Spam Detection	Spelling Errors, Grammatical Errors, Email domain	**Spam** OR / AND **Not Spam** OR / AND **Unsafe**

Common Classification Use Cases

Binary

- Machinery Outage Prediction - **Failure** OR **Not Failure**
- Anomaly Detection – **Fraud** OR **Not Fraud**
- Credit Card Default - **Customer Likely to Default** OR **Not Likely to Default**

Multi-Class

- Product Classification – **Red Wine** OR **White Wine** OR **Rose Wine**
- News Classification of Articles – **Sports** OR **Lifestyle** OR **Economy** OR **Current Affairs**
- Facial Image Recognition – **Happy** OR **Sad** OR **Angry**

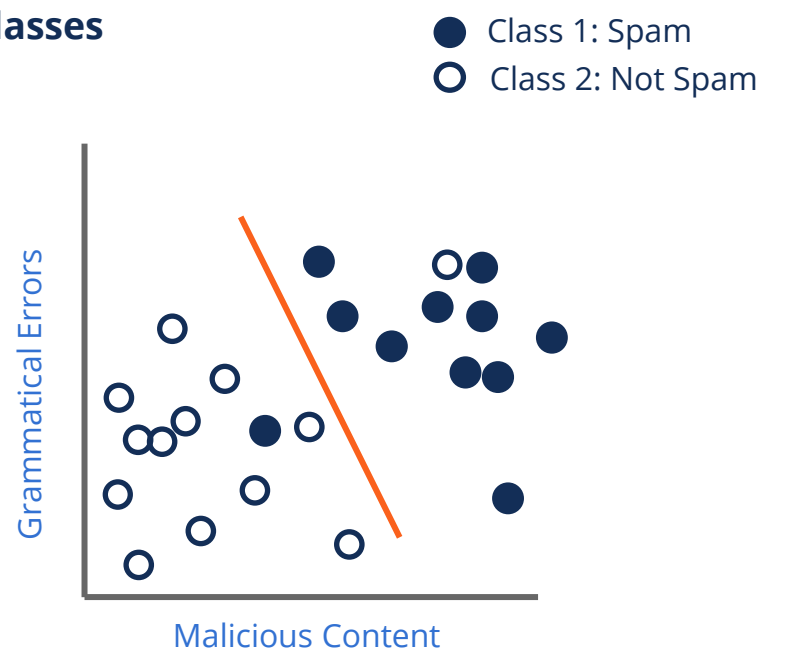
Multi-Label

- Social Tag Selection - **#TogetherAtHome** OR / AND **#COVID19** OR / AND #**WorkFromHome**

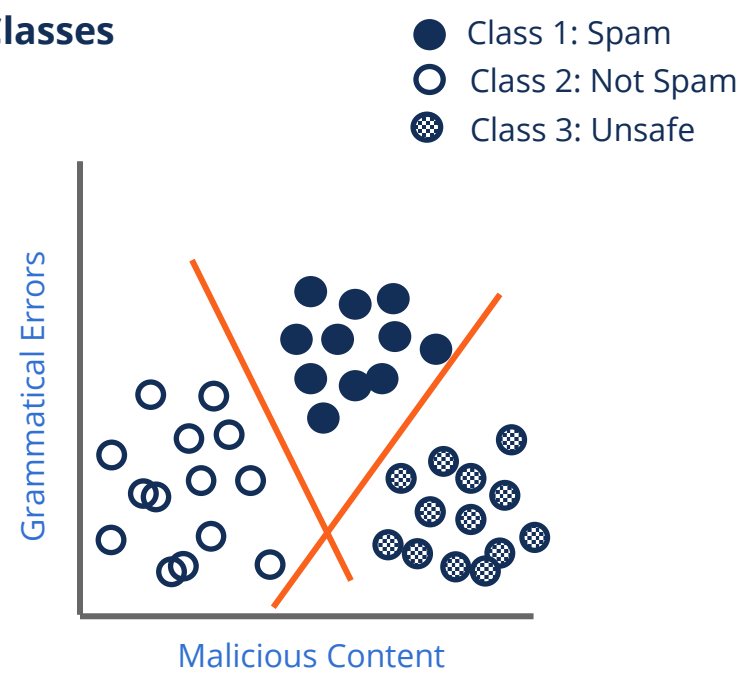
Visualizing Classification

For simple scenarios, it can help to visualize the **input variables** and **output classes** on a chart.

Two Classes



Three Classes

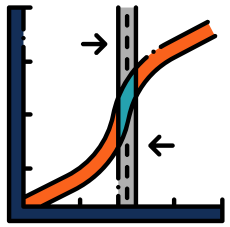


In each case, the goal is to use the input data to **separate the classes** as cleanly as possible.

Classification Algorithms

Throughout the course, we'll explore the most common classification algorithms.

Logistic Regression



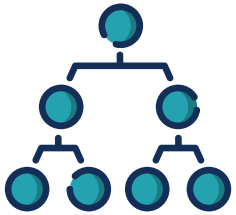
Uses regression principles to achieve separation between discrete classes

Naïve Bayes

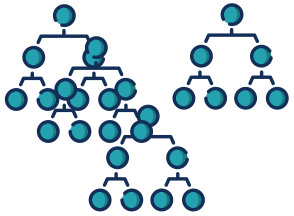
KNN

SVM

Decision Trees



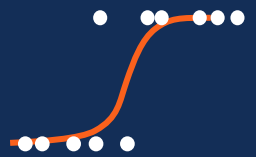
Random Forest



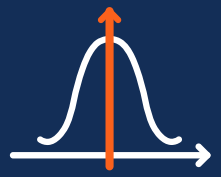
We will look at the benefits of each and how to interpret and evaluate the outputs.

Logistic Regression

Logistic Regression



Understand and learn the fundamental concepts of Logistic Regression



Understand and calculate the probabilities, log odds and how these impact model interpretation

Learn how to interpret log odds and the assumptions behind Logistic regression

Be comfortable with the mathematics behind Logistic Regression and how to manipulate it

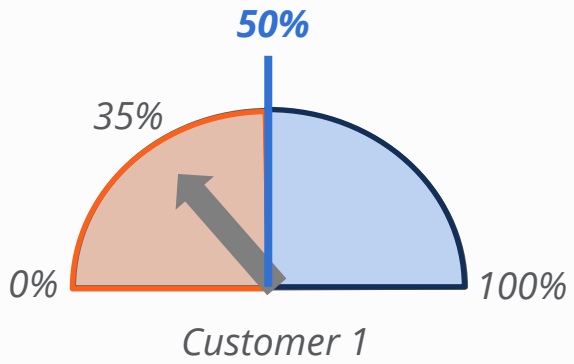
Be able to comprehend and interpret the outputs of logistic regression algorithm for business scenarios.

Practice a basic logistic regression example in Excel and Python.

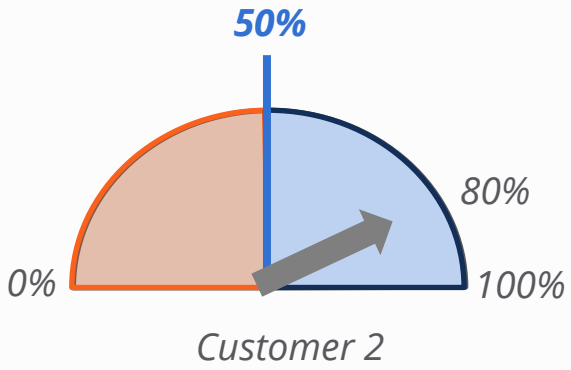
Logistic Regression

Logistic regression makes a classification based on the **probability of an event happening.**

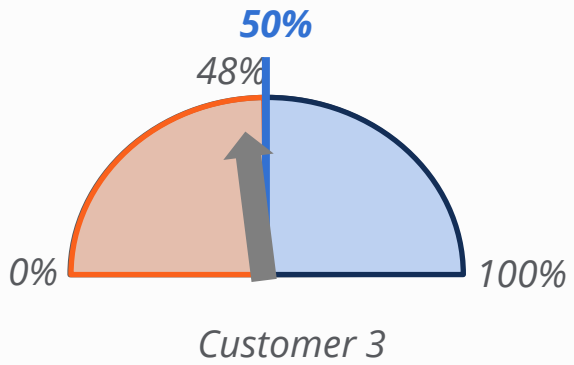
"What is the probability that this customer will purchase medical supplies?"



Prediction: Will Not Buy



Prediction: Will Buy



Prediction: Will Not Buy

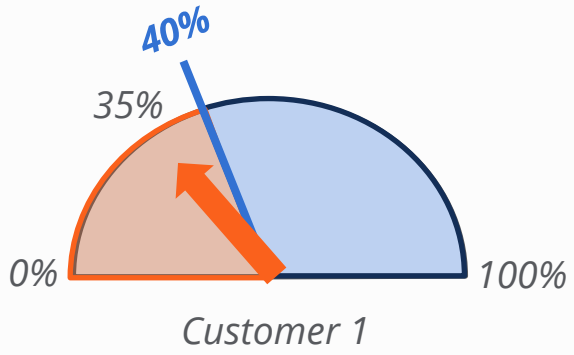
We determine a **threshold** (typically 0.5 or 50%) as the **cut off between prediction classes**.

In this case: customers **lower than the threshold** are predicted to **NOT buy.**

customers **higher than the threshold** are predicted to **WILL buy.**

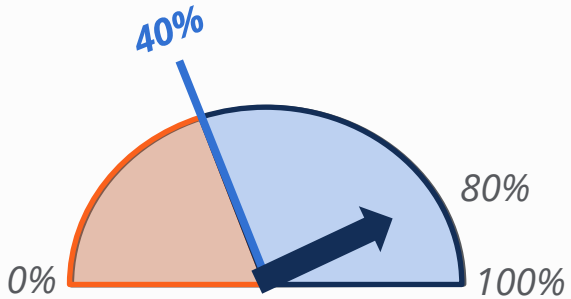
Logistic Regression

Changing the threshold will **change the prediction class** of some observations.



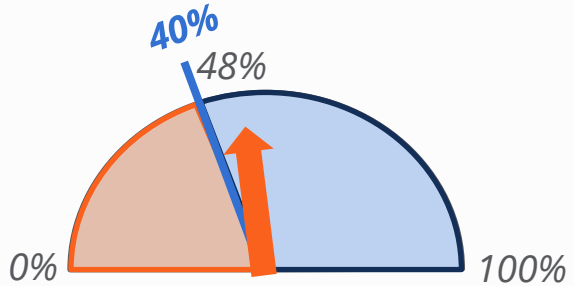
Customer 1

Prediction: Will Not Buy



Customer 2

Prediction: Will Buy



Customer 3

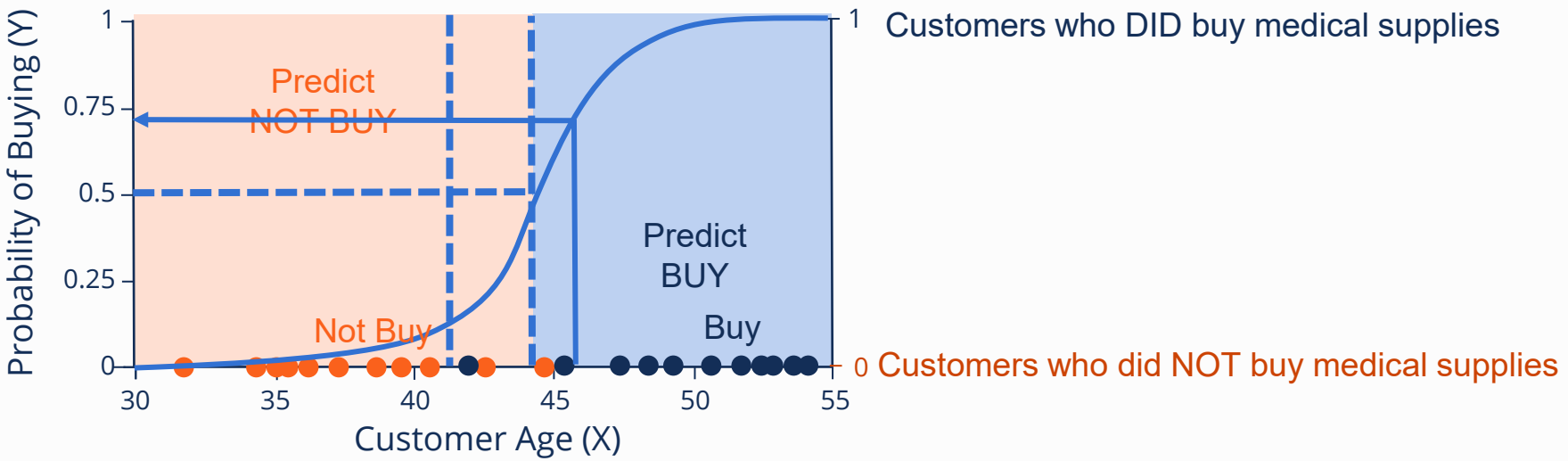
~~***Prediction: Will Not Buy***~~

Prediction: Will Buy

We can use **evaluation metrics** to help us decide on the **most appropriate threshold**.

Visualizing Logistic Regression

Logistic Regression probabilities are estimated using **one or more input variables**



Logistic Regression uses a curved line to summarize our observed data points

The logistic regression line generates probabilities **between 0 and 1**.

Defining the Logistic Regression Curve

- The logistic regression is based on the logistic function, also called the **sigmoid function**

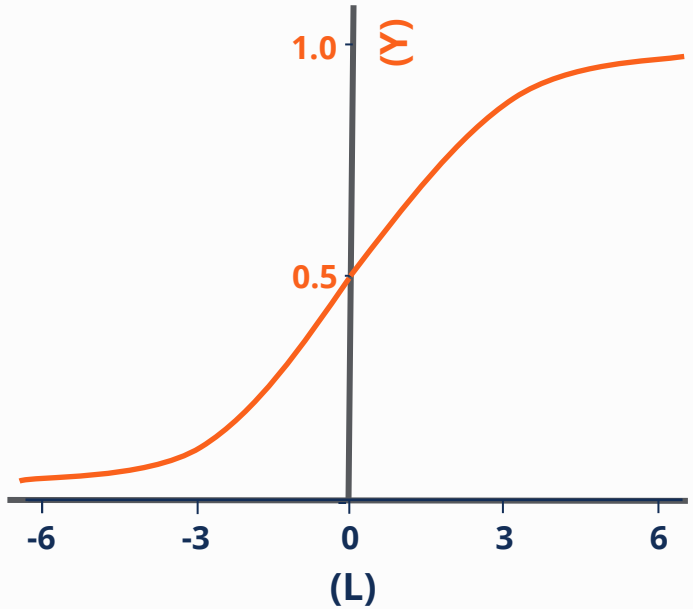
- Mathematically this is defined as

$$f(x) = \frac{1}{1 + e^{-L}}$$

(Where 'e' is the base of the natural logarithm)

- The **input value of L** determines the **output value of y**.

- The logistic function outputs numbers **between 0 and 1**.



At **input (L) 0**, **output (y) = 0.5**.

As **input (L) increases**, **output (y) increases**.

Defining the Logistic Regression Curve

To transform a linear regression into a logistic regression we take the linear regression equation...

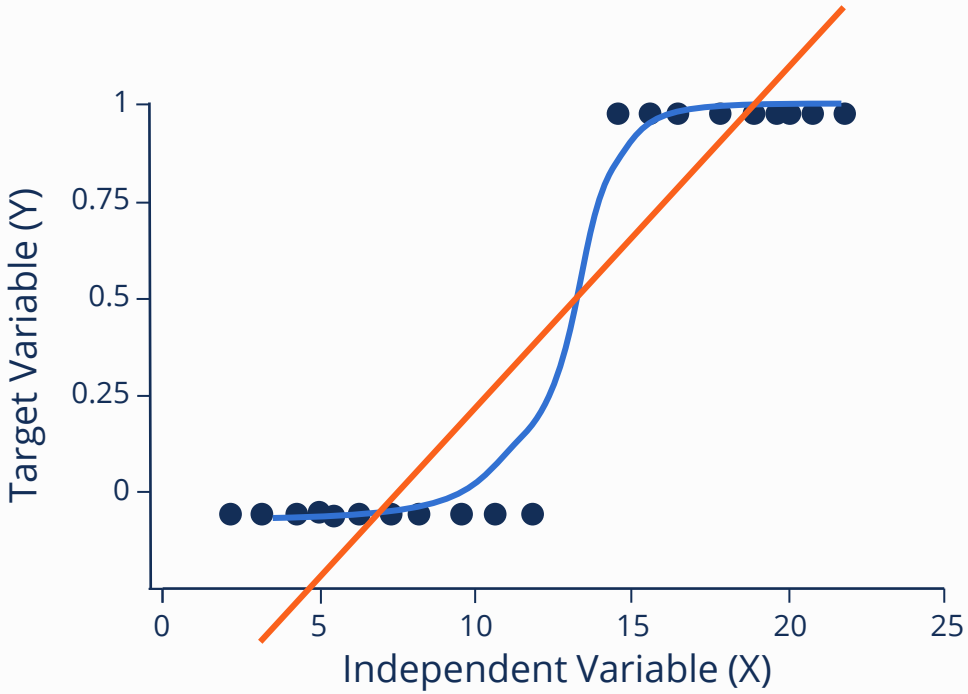
$$\beta_0 + \beta_1 x$$

and substitute it for L in our Logistic function...

$$\hat{y} = \frac{1}{1+e^{-L}}$$

$$P(Y = 1|X) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

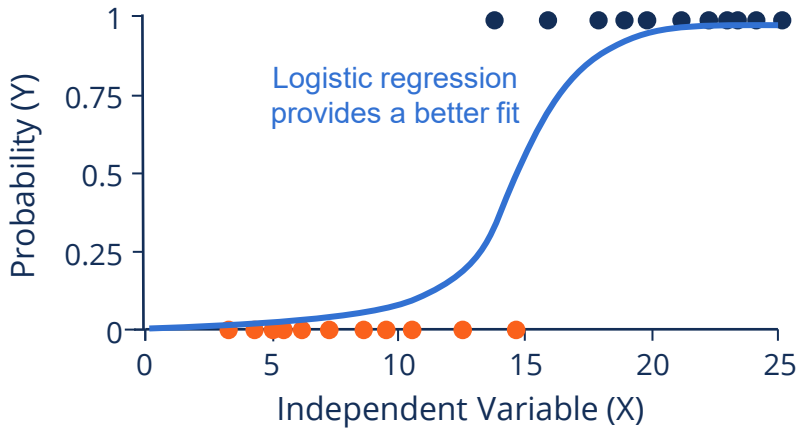
(where P(Y=1 |X) is the probability of Y=1 given X input features)



The logistic curve has been **transformed to fit** the input data.

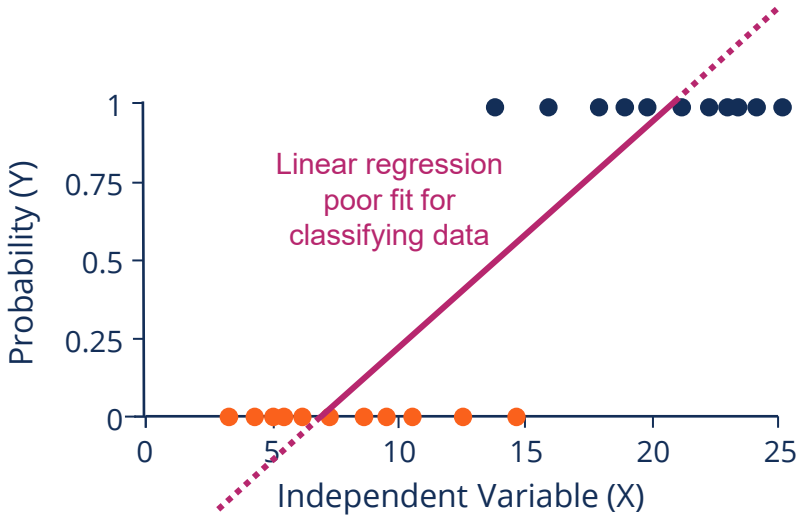
Summary of Logistic Vs Linear Regression

Logistic Regression



- Probabilities always sit **between 0 and 1**.
- More suitable for predicting a **binary outcome**.
- More suitable for **Classification**.

Linear Regression

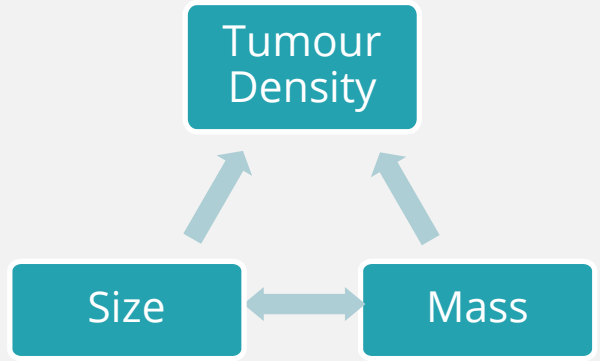


- Probabilities are **not limited to a sensible range**.
- Suited to predicting **number of something**.

Logistic regression assumptions

1. The dependent variable is binary i.e. fits into **one of two clear-cut categories**
2. There should be no, or very little, **multicollinearity** between the predictor variables— meaning the independent variables should be **independent of each other**
3. Logistic regression requires **large sample sizes**—the larger the sample size, the more reliable the results
4. The independent variables should be **linearly related to the log odds**

With **multicollinear** variables, the algorithm would be **unable to separate their effects** likely causing errors



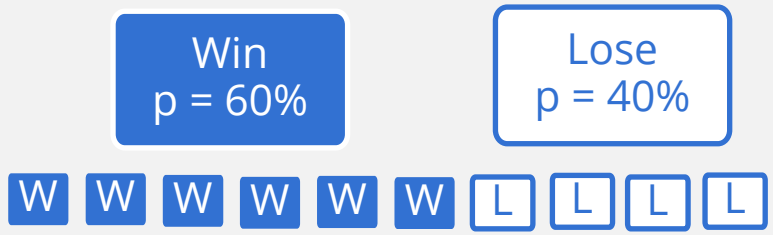
Multicollinearity would happen because **size and mass** hold **similar information**.

Probability, Odds and Log Odds

- **Probabilities** are the **chance** of something happening, **relative to all outcomes.**

- **Odds** are the chance of something happening, **relative to other outcomes**.

- **Log Odds** are simply the **log of the odds.**
- Log odds are **easier for statistical models** to work with.



$$\text{Odds Of Winning} = \frac{p(W)}{p(L)} = \frac{0.6}{0.4} = 1.5 = \frac{p(W)}{1 - p(W)}$$

$$\text{Log Odds Of Winning} = \ln\left(\frac{60}{40}\right) = \ln(1.5)$$

Logistic Probabilities, Odds and Log Odds

Interpreting the impact of a change in an input variable appears to be difficult at first.

$$P(Y=1|X) = \frac{1}{1+e^{-(\boldsymbol{\beta_0}+\boldsymbol{\beta_1 x})}}$$ ← Here x is our input variable.

Rearranging our logistic regression equation we can reach the following:

$$P(Y=1|X) = \frac{1}{1+e^{-(\boldsymbol{\beta_0}+\boldsymbol{\beta_1 x})}}$$

REARRANGE

$$\ln\left(\frac{\boldsymbol{P(Y=1)}}{\boldsymbol{1-P(Y=1)}}\right) = \ln\left(\frac{P(Y=1)}{P(Y=0)}\right) = \boldsymbol{\beta_0 + \beta_1 x}$$

$$\ln\left(\frac{\boldsymbol{P(Y=1)}}{\boldsymbol{1-P(Y=1)}}\right) \qquad = \boldsymbol{\beta_0 + \beta_1 x}$$

Log Odds of being 1 **Linear Inputs**

Interpreting Coefficients

We can interpret the coefficients in two ways:

$$\text{Log}\left(\begin{array}{c}\text{Odds of}\\ \text{being 1}\end{array}\right) = \beta_0 + \beta_1 x$$

For every **unit change in x**

the **log odds** will change by β_1.

$$\begin{array}{c}\text{Odds of}\\ \text{being 1}\end{array} = \exp\left(\beta_0 + \beta_1 x\right)$$

For every **unit change in x**

the **odds** will change by **exp(β_1)**

Odds are generally **easier to interpret** than log odds.

Interpretation Scenario

*You are tasked by your company with **predicting the likelihood of purchase** of medical supplies (Y).*

Input features:

- **X_1 = Customer tenure**
- **X_2 = Purchased in the last year**

Equation

Odds of Purchase = $\exp(\beta_0 + \beta_{Tenure}\, x_1 + \beta_{Purchased}\, x_2)$

Coefficients:

- Customer loyalty coefficient β_{Tenure} **= 0.6**
- Purchased last year coefficient $\beta_{Purchased}$ **= 0.2**

Odds Interpretations:

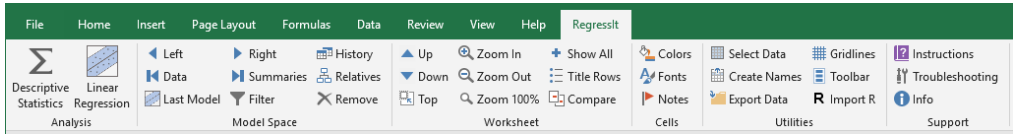
- $\exp(\beta_{Tenure})$ = **1.82**
  "Every extra year as a customer **increases the odds** of purchase by a factor of **1.82.**
- $\exp(\beta_{Purchased})$ = **1.22**
  "Customers that purchased in the last year have **22% higher odds** to buy again this year.

Logistic Regression in Practice

Excel

- ***RegressIt*** is a powerful Excel add-in tool that performs multivariate descriptive data analysis and linear as well as logistic regression

RegressIt™ Free Excel add-in for linear regression and multivariate data analysis



Python packages

Scikit-learn *provides a range of supervised and unsupervised learning algorithms*

statsmodels *provides classes and functions for many different statistical models and tests*

Classification Algorithms

Classification Algorithms

Learn about the basic Classification algorithms and when and how to use them

Understand the underlying considerations and assumptions of these algorithms

Learn the inner workings of the algorithms and understand how they categorize observations

Understand how to interpret the parameters, outputs and evaluation metrics for the algorithms

Be able to apply these to real-world scenarios and learn about their applications

Apply each classification algorithm in Python identify the quality of results from each.

Algorithms Overview

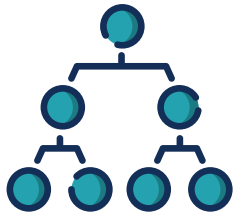
We will review five common algorithms used to build classification models:

Naïve Bayes

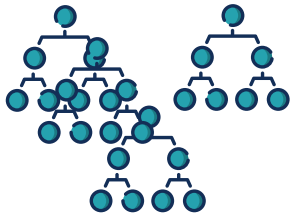
KNN

SVM

Decision Trees



Random Forest



Naïve Bayes

- Naïve Bayes is a probabilistic model based on **Bayes theorem**; it gives us classifications based on probabilities
- **Bayes theorem** generates the probability of one event, given the probabilities of other events

Strengths

- Easy to use and good for large datasets
- Can be used to solve multi-class prediction problems

Weaknesses

- Assumes independence in features, which makes it less applicable on most real-world datasets

- Bayes Theorem states the **probability (P)** of an event A happening given that an event B occurred can be given by:

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

Where A is the ***hypothesis*** or ***outcome variable,*** and B is the ***evidence*** *or* ***features***

Naïve Bayes – Example

We want to predict the likelihood of an email being spam, given it contains a grammatical error.

Observation	Spam Email?	Grammatical Errors?
1	Yes	Yes
2	No	No
3	No	Yes
4	No	No
5	No	No
6	Yes	No
7	No	No
8	No	No
9	No	Yes
10	No	No
…	…	…

20 Emails	Spam	Not Spam	
Grammatical Errors	3	4	7
No Grammatical Errors	1	12	13
	4	16	20

Naïve Bayes – Example

We want to predict the likelihood of an email being spam, given it contains a grammatical error.

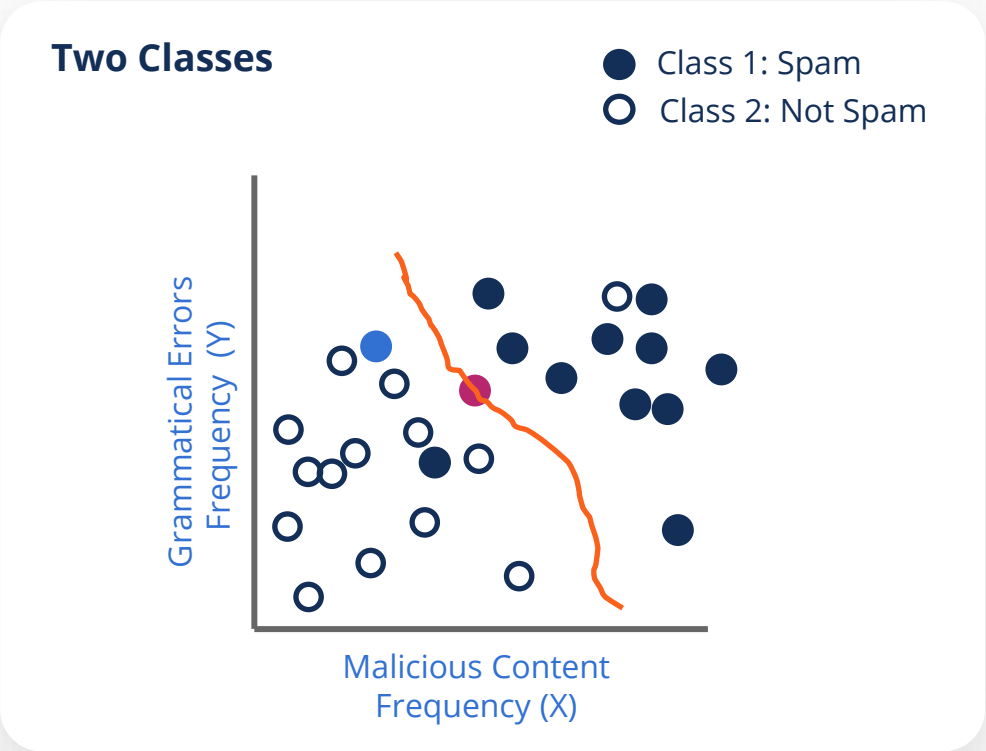
20 Emails	Spam	Not Spam	
Grammatical Errors	3	4	7
No Grammatical Errors	1	12	13
	4	16	20

- $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$
- P(Spam | Error) =
  - $P(Error|Spam) = 3/4$ or **0.75**
  - $P(Spam) = 4/20$ or **0.20**
  - $P(Error) = 7/20$ or **0.35**
- P(Spam | Error) = $\frac{0.75 * 0.20}{0.35}$ = **0.43** or **43% likely**

K-Nearest Neighbours (KNN)

KNN assigns output classes based on the most similar observations in our sample space.

By similar, we mean those who have the closest input values.



- **New observation** - Which class would you likely assign it to? **Not Spam!**

  The new observation is closer to the Not Spam observations, meaning its characteristics are more similar.

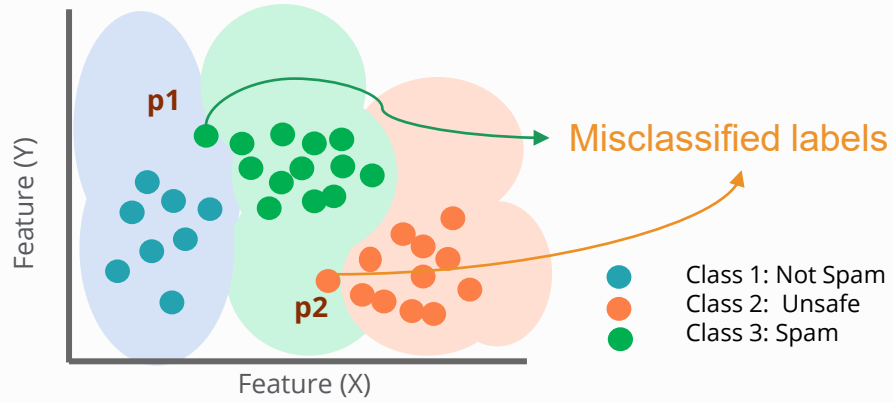
- **New observation** - The nearest observation is Spam.

  However, the next 3 closest are Not Spam.

  How many of the nearest observations should we choose?

- Once we have optimized this number of nearest neighbours, we can visualize our decision boundary, which represents the boundary between Spam and Not Spam.

K-Nearest Neighbours - Example



As the misclassified point **p1** (Not Spam) has a green point (Spam) closest, a value of K=1 misclassifies the Not Spam email as Spam.

The exact thing happens in the case of **p2**

Choosing the right value for K is critical

K=1 will simply classify the data point on the basis of *one closest neighbour*

K=1000 will not identify categories in the data at all

Increasing K improves prediction due to averaging of the distance, the algorithm selects the most suitable point

An optimal value of K must be selected depending on the size of the dataset

Strengths

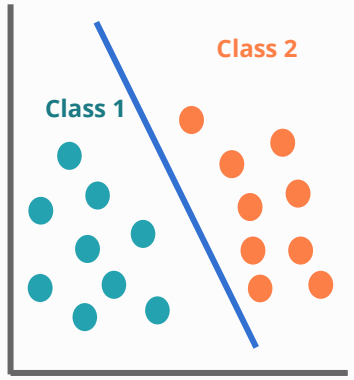
- Simple and easy implementation (no assumptions)
- Can be used for classification and regression

Weaknesses

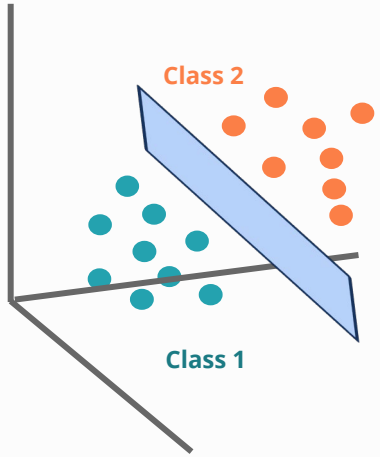
- Performance decreases as the number of examples and/or independent variables increases

Support Vector Machines (SVM)

In 2D sample space, separation is created using a line



In 3D sample space, separation is created using a plane



SVMs separate data points with a line or plane through the sample space

SVM aims to find a plane that maximises the separation distance between data points of both classes

The algorithm defines the criterion for a boundary that is maximally far away from any data point

This distance to the closest data point from the decision surface determines the margin of the classifier

Strengths

- Uses a subset of training points in the decision function so it is memory efficient

Weaknesses

- The algorithm does not directly provide probability estimates, these are calculated using expensive techniques

Decision Trees

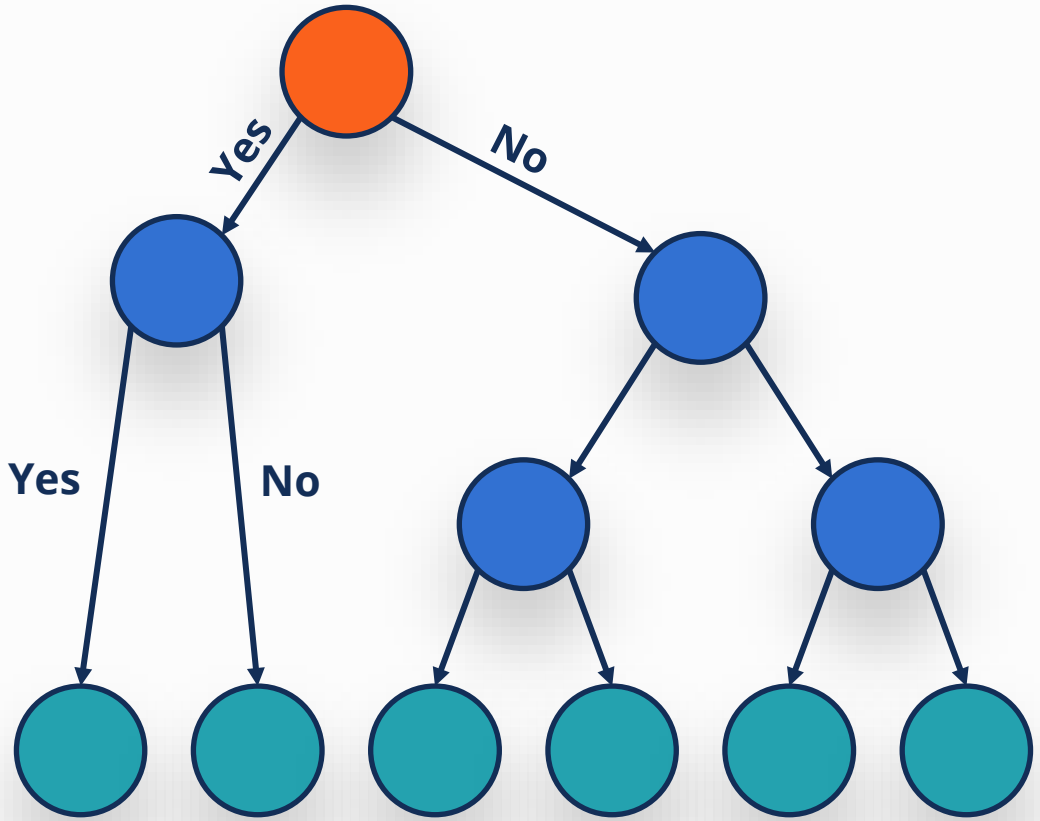
A decision tree is a cascading set of questions, used to incrementally separate classes and improve predictive power.

Strengths

- Simple to understand and interpret (visualise)
- Used for classification or regression
- Requires little data preparation as it can handle both numerical and categorical

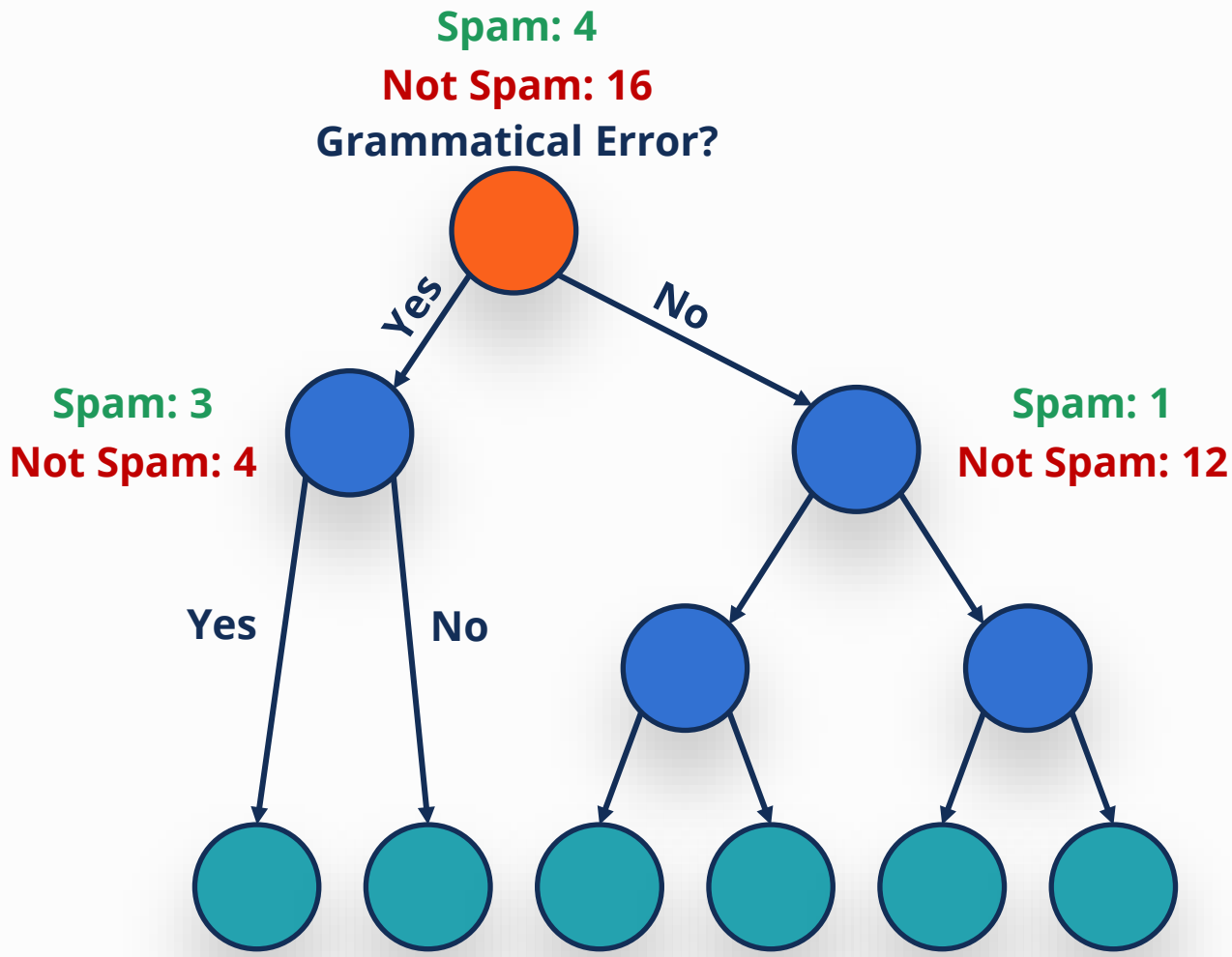
Weaknesses

- Tends to overfit (the trained tree doesn't generalise well to unseen data)
- Small changes in data tends to cause big difference in tree (instability)
- Can become expensive to compute with high dimensionality



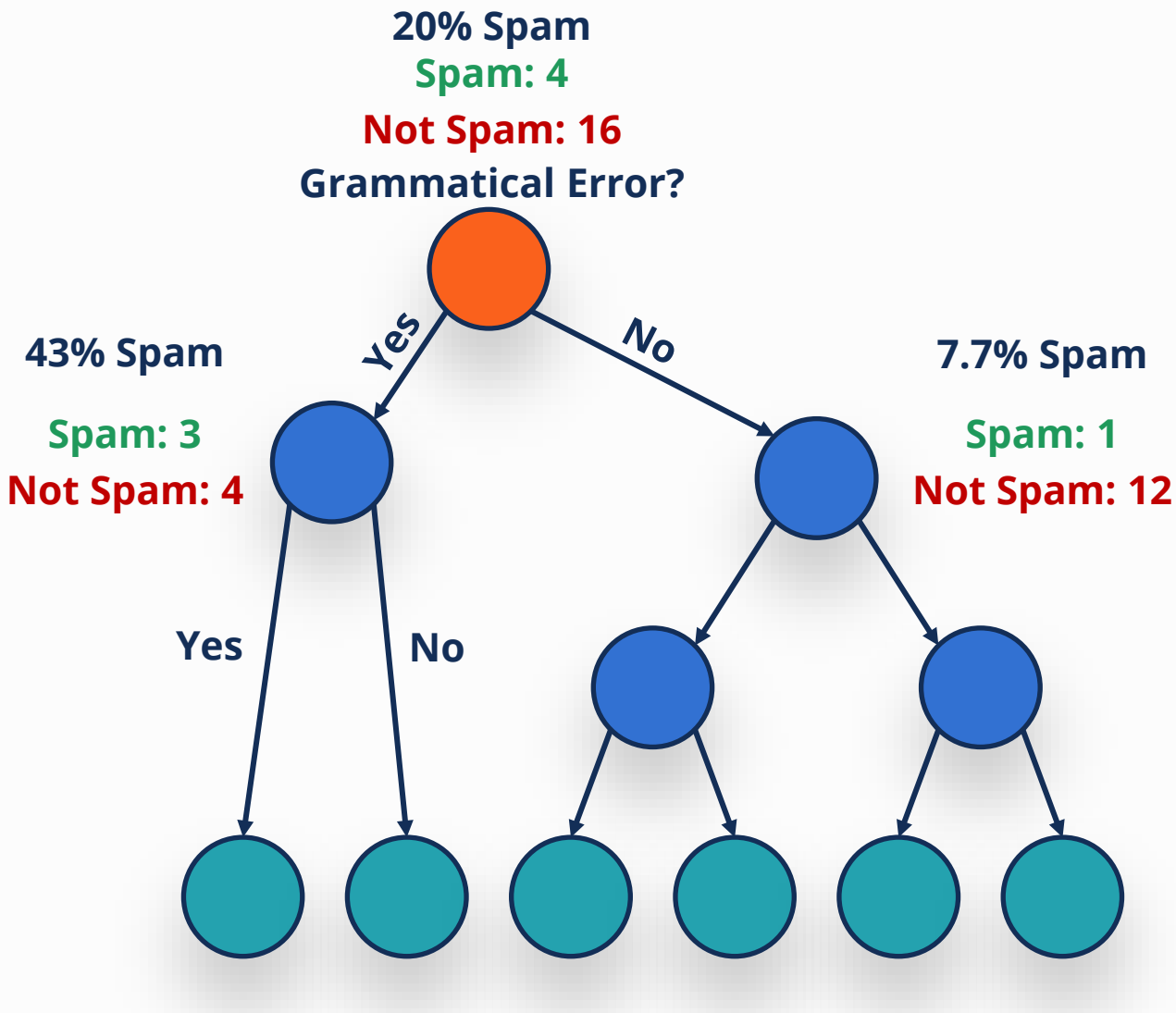
Decision Trees - Example

- Each node represents a feature/attribute of the data set and **branches** represent a **decision**
- The decision tree starts at the **root node**
- We go down the tree asking true/false questions at **decision nodes**...
- ...until the **leaf node** (or **outcome**) is reached
- Once the tree is completed, it can be used to evaluate each email by answering the questions until a leaf node is reached and a prediction is made.



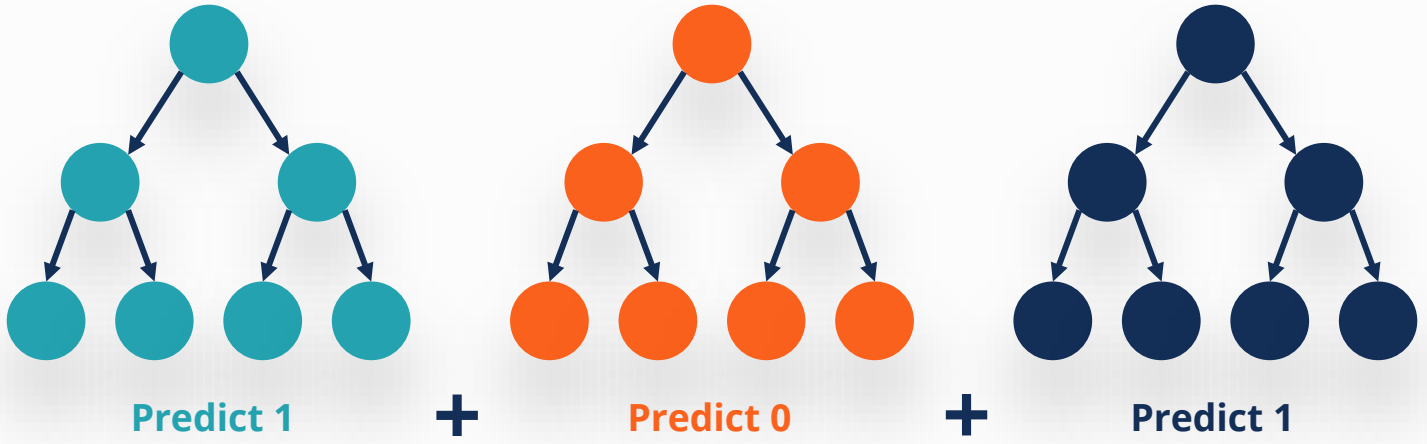
Decision Trees - Example

- **Splitting** is the process of dividing a node into child nodes
- The goal is to make each of the child nodes more “pure” or homogenous – containing more similar classes of observations
- Separating emails according to grammatical errors adds predictive value to the model



Random Forest

A random forest is known as an ensemble model since it combines the results from multiple other models; in this case decision trees.



Strengths

- Improved accuracy (reduces overfitting) and more powerful than decision trees
- Used in regression and classification

Weaknesses

- The complexity of the algorithm can cause it to become slow and inefficient
- Real-time predictions can be slow due to large inputs

Evaluation & Interpretability

Evaluation & Interpretability

Understand the basic outcomes of classification and how they are represented visually in a confusion matrix.

Learn how to evaluate and compare models with evaluation metrics such as accuracy, precision and recall.

Understand when certain metrics may be more appropriate than others.

Learn more advanced evaluation metrics that build on basis Precision and Recall, such as F-scores.

Understand how to interpret AUC-ROC curves and use them to compare model results.

Implement the above model evaluation techniques in Python using SkLearn.

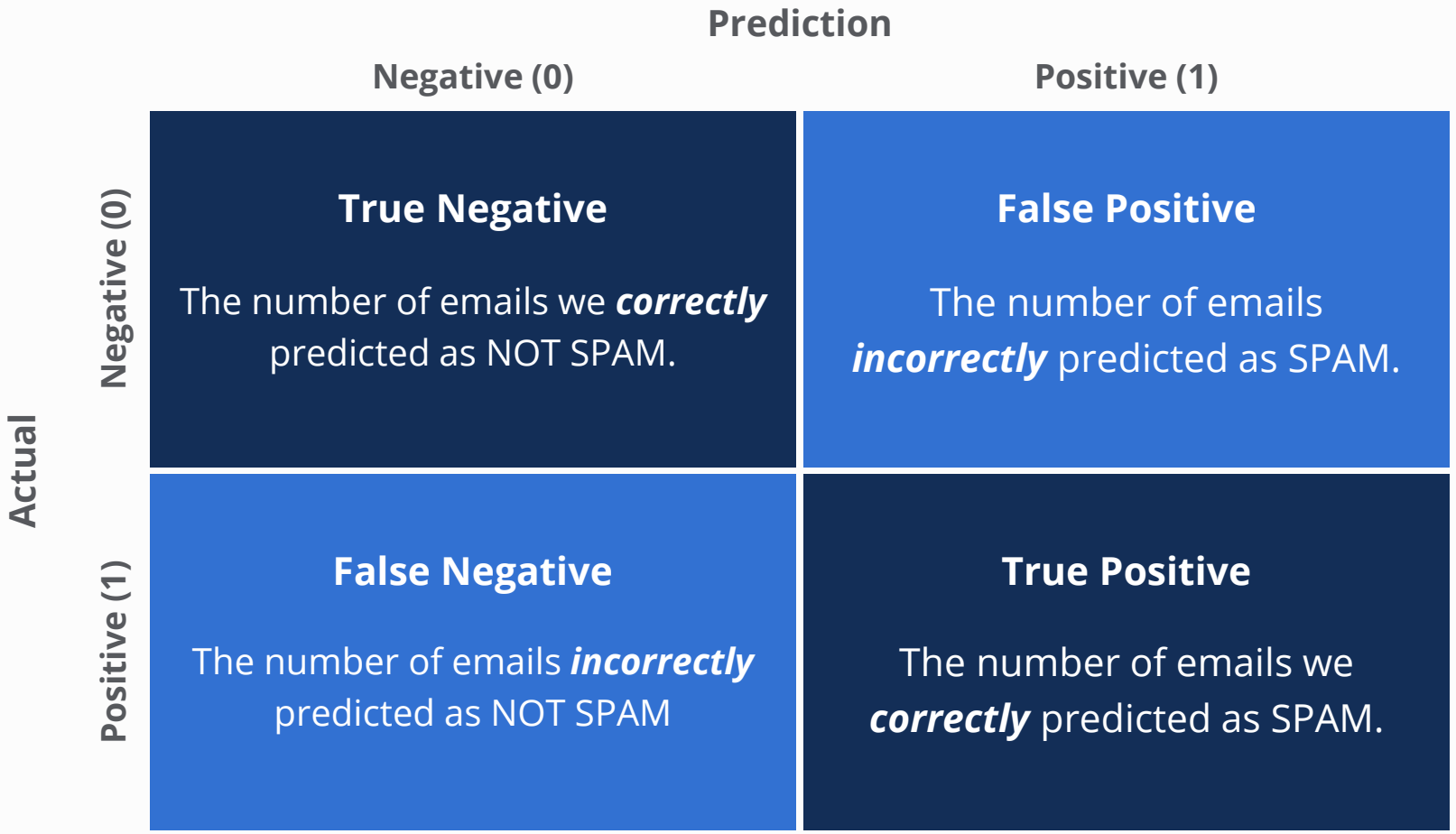
Model Evaluation Basics

- Evaluation metrics are important because they ensure that the model is performing correctly
- To set up our model evaluation, the dataset is often split into **training** and **testing** data

- The model learns from the **training data**
- The **testing data** is used to test how well the model performs on **new *unseen*** data
- Better evaluation results ensure that the model can be used in real-world on **new data** reliably

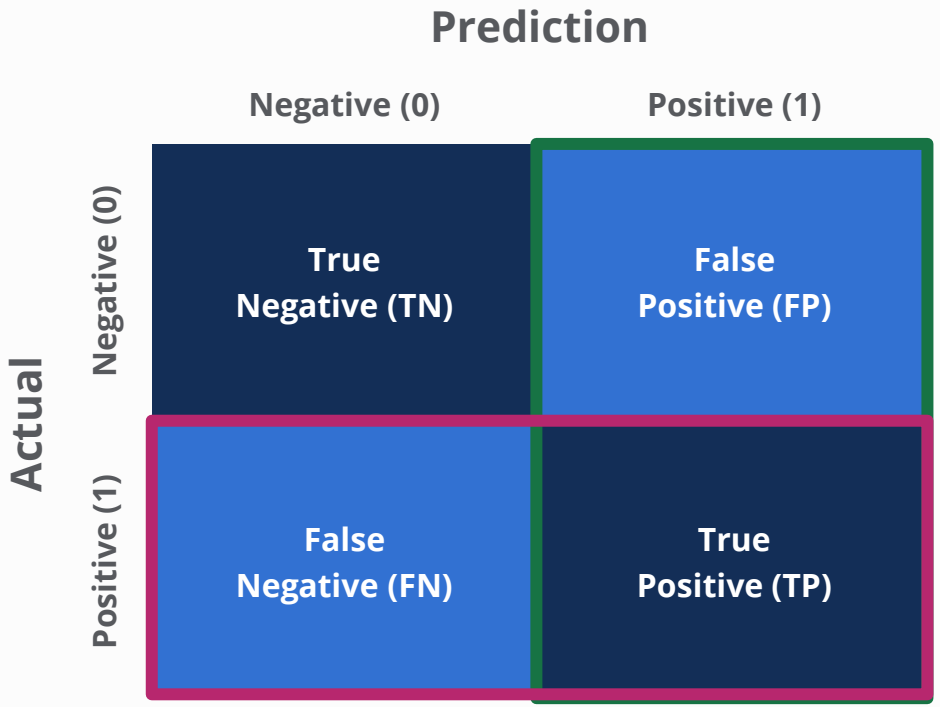
Confusion Matrix

The confusion matrix helps us understand the quality of our predictions.



Evaluation Metrics

There are **four key metrics** that can help summarize the observations in the confusion matrix:



1. **Accuracy = (TN + TP) / Total Predictions**

   Describes what proportion of predictions were correct (may not always be the best indicator of performance).

2. **Precision = TP / (TP + FP)**

   How good are the positive predictions? Out of those predicted positive, how many were actually positive?

3. **Recall = TP / (TP + FN)**

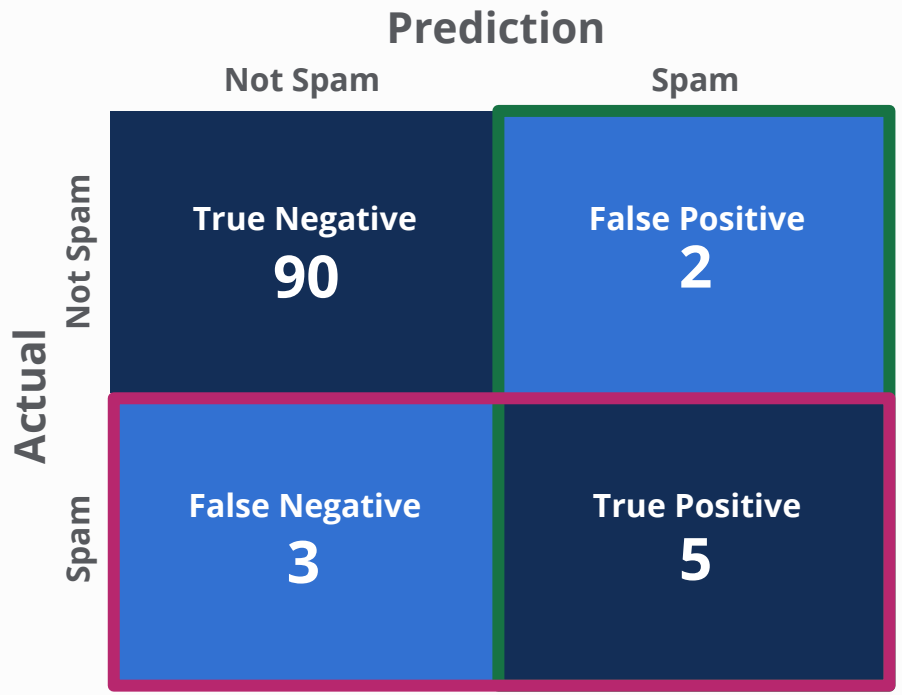
   Describes what proportion of the actual positive cases were correctly identified.

4. **F1 Score = 2 * [(Precision*Recall) / (Precision+Recall)]**

   Provides a balance between precision and recall.

Evaluation Metrics Example

Consider an example where we have 100 emails and we use a model to predict whether an email is spam or not. Here **SPAM** emails is our positive class and **NOT SPAM** is our negative class.



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{95}{90 + 3 + 2 + 5} = 0.95$$

$$Precision = \frac{TP}{TP + FP} = \frac{5}{5 + 2} = 0.71$$

$$Recall = \frac{TP}{TP + FN} = \frac{5}{5 + 3} = 0.62$$

In this situation, **Precision** and **Recall** are equally important:

- we don't want to pass any SPAM emails as NOT SPAM (as they could be dangerous)

and at the same time:

- we would not want any NOT SPAM emails going into our SPAM box (as that email could be important to us).

Precision Vs Recall

- Accuracy works well for **balanced** classes (having roughly equal number of samples of every class)
- **Precision** is a good choice of metric when we have imbalanced classes and we want to minimized false positives.
- **Recall** is a good choice of metric when we have imbalanced classes and we want to minimize false negatives.

  - In tumour risk detection, we would want to reduce the number of **False Negatives.**
  - We cannot afford to miss any malignant samples in the data.
  - **Recall** would be the preferred metric here because it measures the proportion of actual malignant tumours that we detected.
  - Misclassifying a low risk tumour as risky is obviously not ideal, but is a secondary priority.

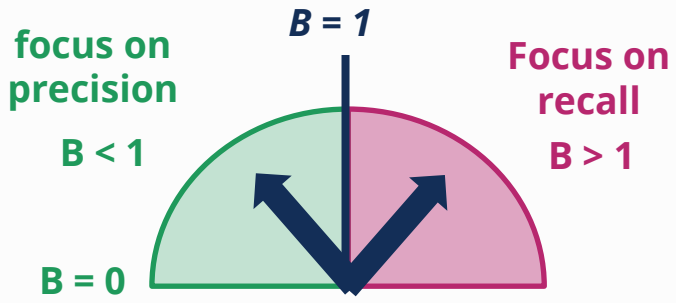
- Ideally, a model with both a high recall and precision score would be preferred.

F_β –Score

F_β is a combined evaluation metric that balances **precision** and **recall.**

By choosing any value of β, we can modify our equation to control the weight of Precision and Recall in our calculations.

$$F_\beta = \frac{1 + \beta^2}{\frac{\beta^2}{Recall} + \frac{1}{Precision}}$$



F_1 and F_2 are the most common iterations of F_β score

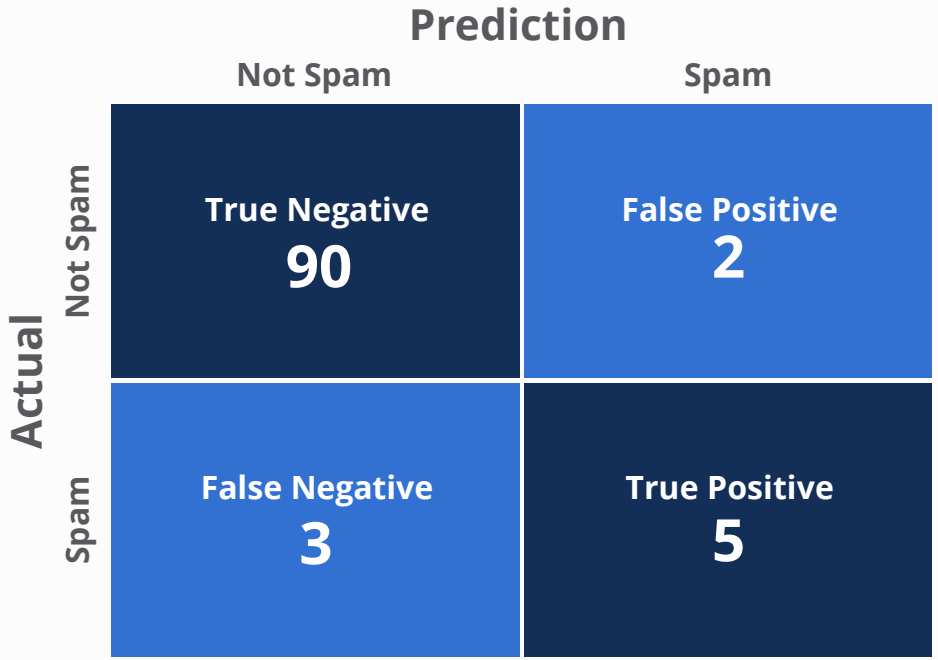
Practical Example F_β

- We can calculate F_1 score as:

$$F_1 = \frac{1 + 1^2}{\frac{1^2}{0.62} + \frac{1}{0.71}} = 0.66$$

- And we can calculate F_2 as:

$$F_2 = \frac{1 + 2^2}{\frac{2^2}{0.62} + \frac{1}{0.71}} = 0.63$$



- F_1 gives equal importance to both **Precision** and **Recall**
- In this example, F_1 would be a relatively better choice as we prefer both Precision and Recall somewhat equally here.

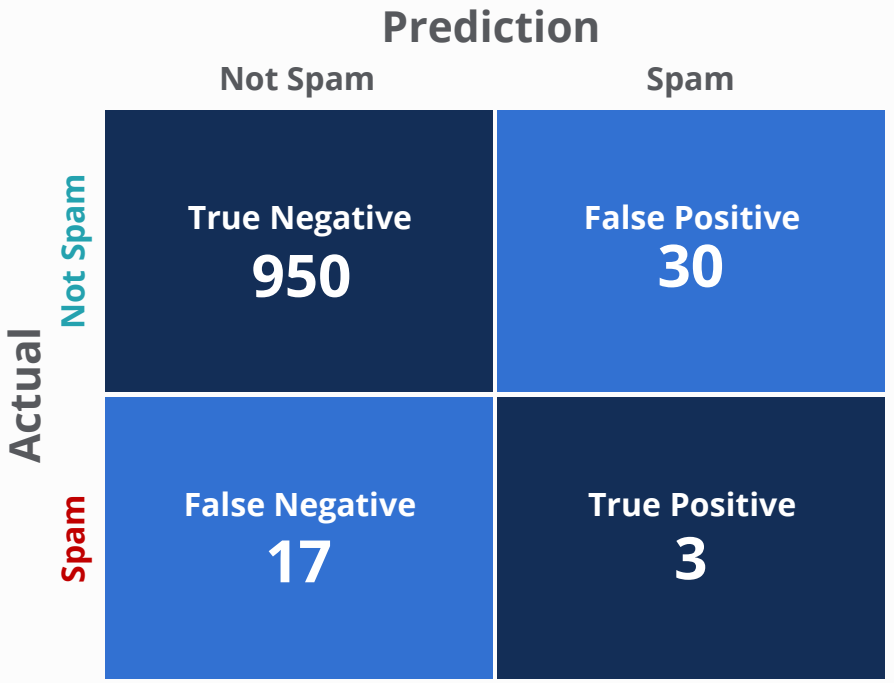
Is Accuracy the Best Choice?

Now consider we have 1000 sample observations:

- **Not Spam** (-ve) 980 samples
- **Spam** (+ve) 20 samples

This is an **imbalanced class** problem.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{953}{10+950+30+17} = 0.95 = 95\%$$



However, we are failing to correctly predict the class we care about - the **SPAM** emails, and we can see it in the poor performance of the other metrics:

- Precision: 3/33 = 10%
- Recall: 3/20 = 15%

So **Accuracy may not be the best choice** for all problems

The ROC Curve

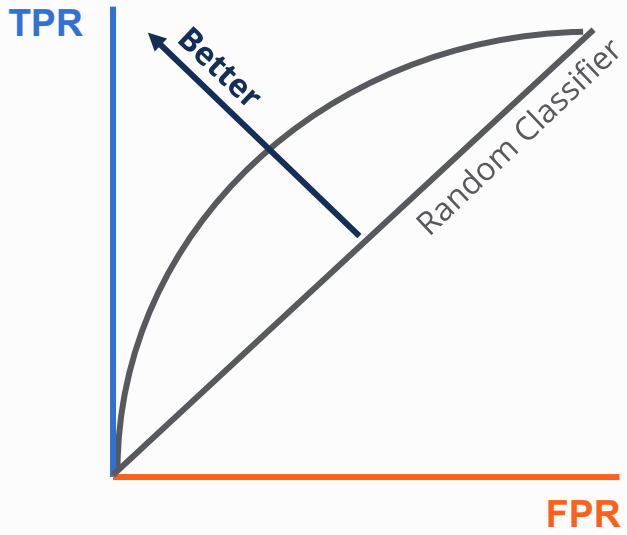
The Receiver Operating Characteristic (ROC) curve visualizes the model performance and is a useful way to evaluate the results of a binary classification model.

True Positive Rate

Describes the proportion of **actual positives** that we correctly identified.

Same as recall.

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



False Positive Rate

Describes the proportion of **actual negatives** that we flagged as positive.

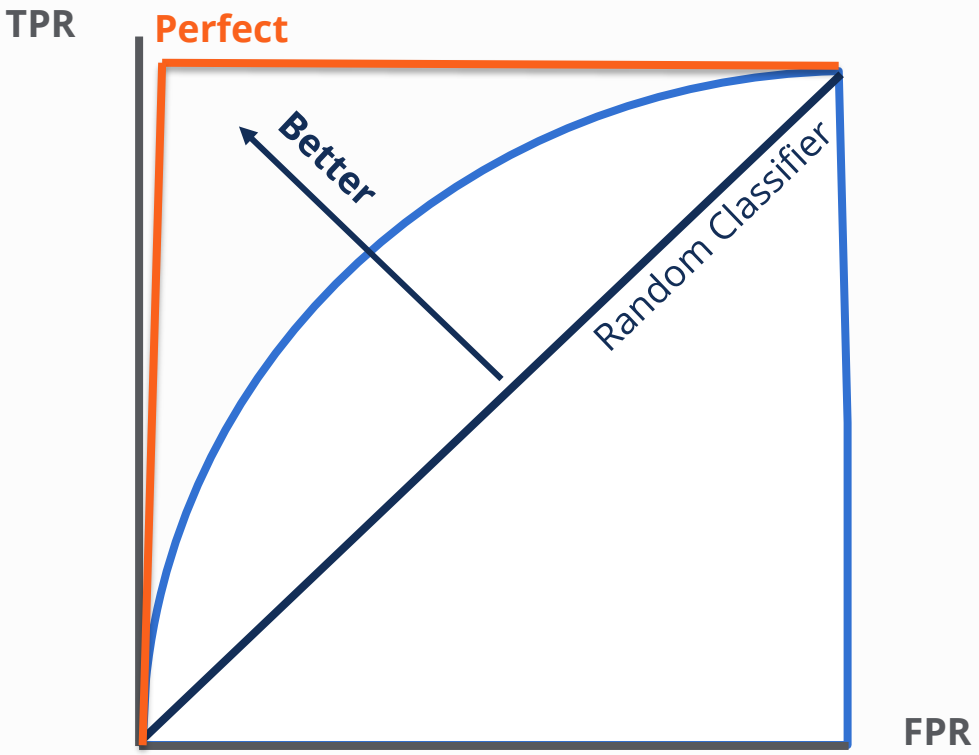
$$FPR = \frac{False\ Positive}{True\ Negative + \ False\ Positive}$$

Each classification model we create can be **plotted as a curve** on this chart.

The further the ROC curve from the **random classifier** (towards TPR), **the better the model is at predicting overall results.**

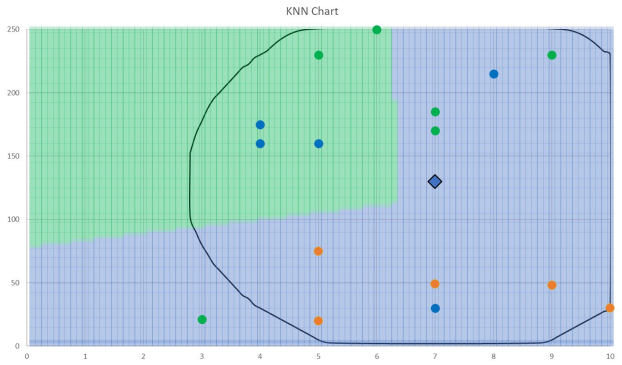
The ROC Curve - AUC

- AUC stands for **Area Under the Curve**. It is calculated as the surface or area that sits underneath the curve.
- Helps summarise the ROC curve into a single number between 0 and 1, to help compare **different algorithms**
- **Higher AUC** (close to 1) means the model is better at separating classes.
- A **perfect model would display a square** in order to maximize the area.



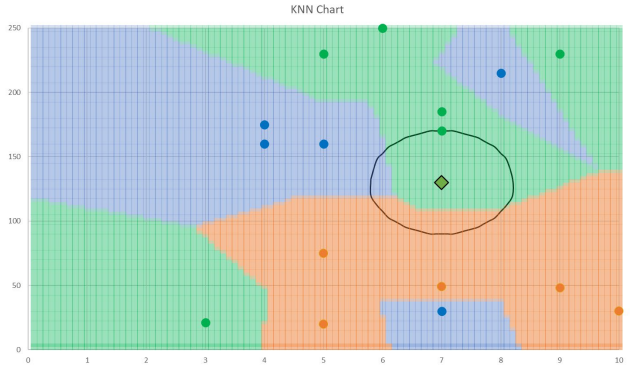
Overfitting Vs Underfitting

Underfitting and Overfitting are terms that helps us summarize the performance of a model.



Underfitting

- Is **too simple** for the scenario
- Does **not perform well on the training data**
- Oversimplifies patterns in the data
- Will **not perform well on testing data**



Overfitting

- **Learns the training data too well** (effectively learns the answers)
- Will **perform very well on training data**.
- But is unable to look at the bigger picture and generalize trends
- Will **likely perform badly on testing (new) data**

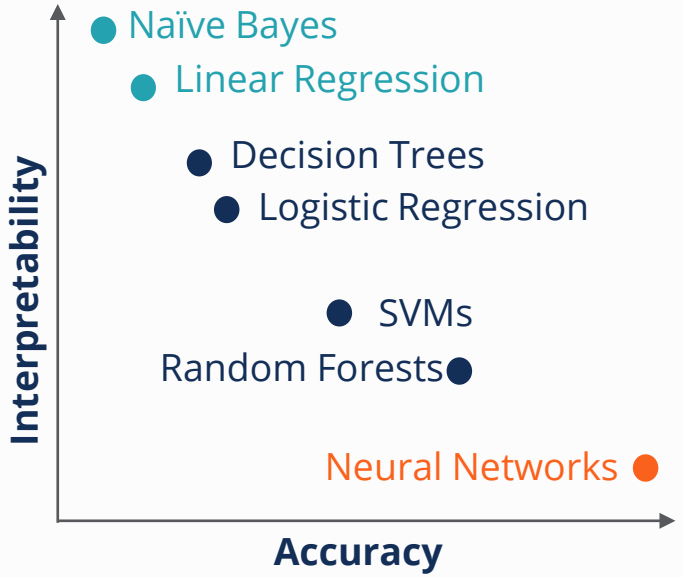
We've built a good working model, now what?

Often a model is the spark for follow up questions, such as:

- To what extent can we **explain the reasons why** our model makes the predictions that it does?
- What are the **main drivers** for making this **class prediction**?
- What are the main drivers for predicting this **customer will purchase my product**?
- What **influences positively/negatively** on that purchase prediction?
- How large is the **influence of each input variable**?
- How can we **explain the difference** between one prediction and the next?

Interpretability of Machine Learning Models

- A model is **interpretable** if it can be understood by anyone without additional explanation.
  - Interpretability ensures that the model is **reliable, fair, robust and reasonable**
  - Basic models, like **Linear models** or **Naïve Bayes models**, are highly interpretable and user-friendly
- A model is **accurate** is it can make higher quality predictions on average.
  - For better **accuracy** in models that use real-world data we tend to require more complex models (often less interpretable) like **Neural Networks.**



We must consider the extent to which we are required to dissect and provide interpretable outputs.

Interpretability and Explainability

Interpretability A model is interpretable if you can work out the simple **cause and effect relationship between inputs and outputs with relatively simple logic**.

Explainability A model is **explainable** if we can fully dissect each part of the model and explain it's role in the decision making process.

White Box Models are **easy to interpret and explain.**

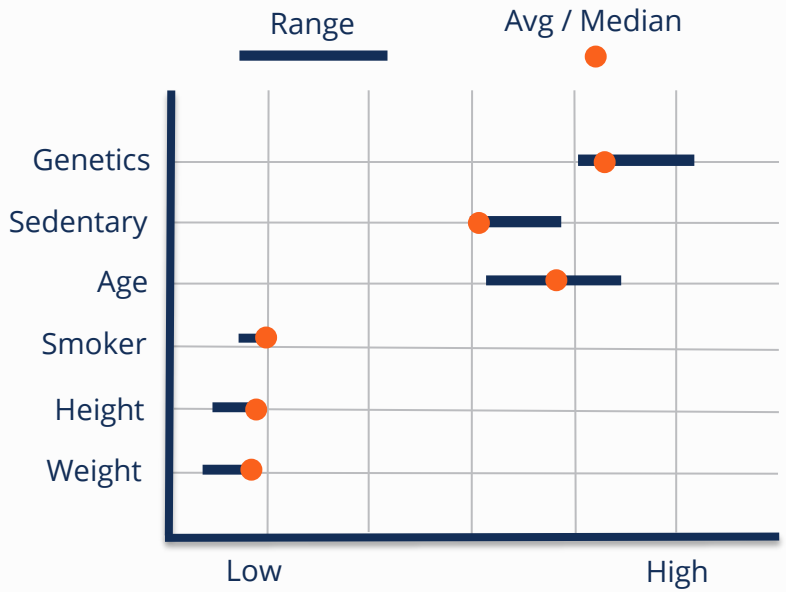
Reinforces accountability and audit.

Black Box Models make it **difficult to pin-point causality** for a particular outcome.

Less helpful for accountability or audit.

Introduction to Feature Importance

- **Feature importance** helps determine how well each of the individual features helps predict the **overall outcome**.
- The **higher** the total importance, the **more influence** that feature has on what we want to predict.
- This technique tends to be used in tree-based algorithms such as **Decision Trees** or **Random Forests** but can also be used more generally.
- There are different types of feature importance depending on **how we define importance** e.g. *number of times a feature is used to split a branch in a decision tree.*



Feature importance
(across model iterations)

We can see **genetics**, **age** and **being sedentary** have high importance when **predicting whether a tumor is malignant.**

Strengths

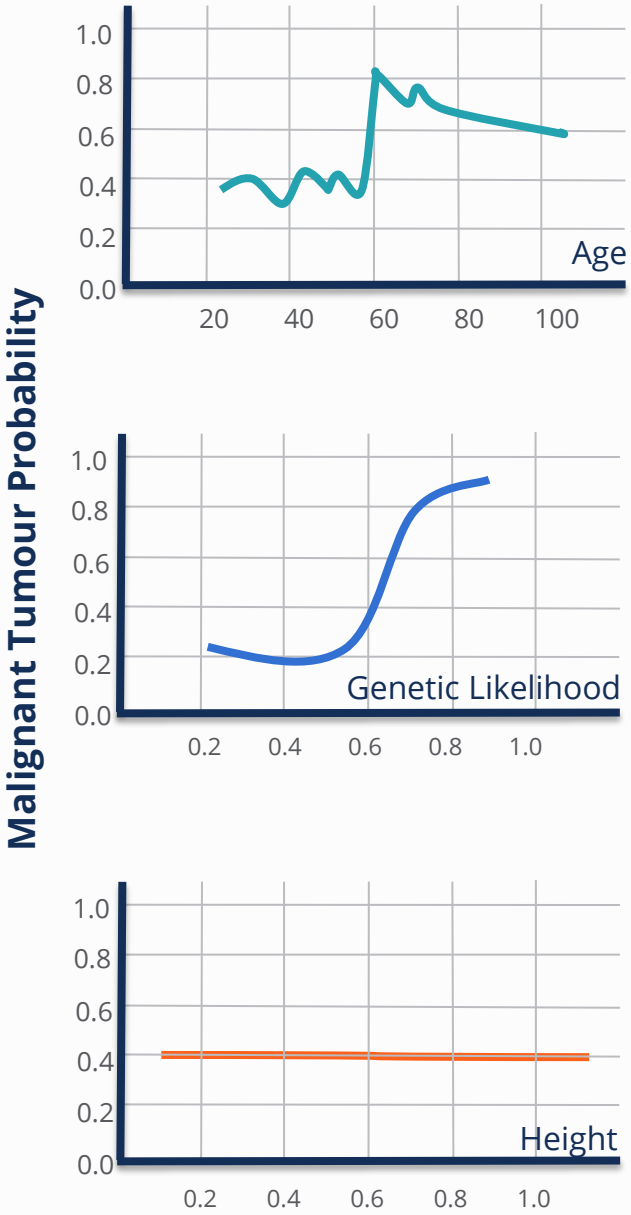
- Model learns better by prioritising important features
- Training time and memory requirements reduces

Weaknesses

- Mostly rely on approximations other than for linear models
- Most methods cannot deal with high-dimensional data

Introduction to Partial Dependence Plots (PDP)

- **Partial Dependence Plots** (PDP) show the **marginal effect** of independent feature(s) on **model predictions**
- They shows whether the relationship between the outcome and predictor variable is **linear**, **monotonic** or **more complex**
- **Straight flat PDP** indicates that the feature is not important.
- When predicting malignant tumours – **increase in age (>55)** and **genetic likelihood score (>0.5)** means higher likelihood of tumour being malignant



Strengths

- The interpretation of the plots is intuitive and easy to understand

Weaknesses

- Assumes independence in the features
- Limited interpretability with more than 2 features

Introduction to SHapley Additive exPlanations (SHAP)

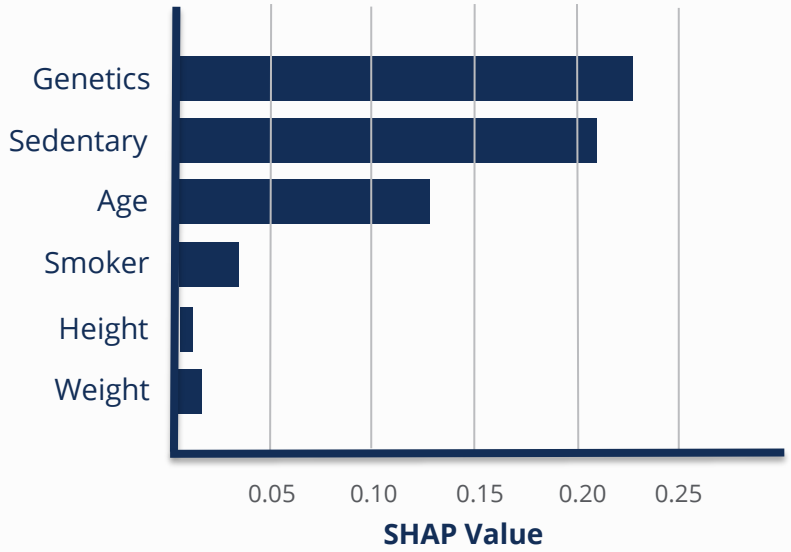
- **SHAP values** help explain the **contribution** each feature is making on **an individual prediction.**
- When we are talking about individual observations, we are talking about **local explainability**.

Instead of asking:

- How much is the prediction of a tumour being malignant driven by a persons age?

We would ask:

- How much is the prediction of this patients tumour being malignant driven by the fact that she is over 65 years old?



Genetics was the highest driver behind this positive prediction.

Strengths

- Makes black-box models easy to explain to all audiences
- Allows for decomposing of each prediction by all the features

Weaknesses

- SHAP can be slow to plot
- It is possible to create intentionally misleading interpretations that can hide biases