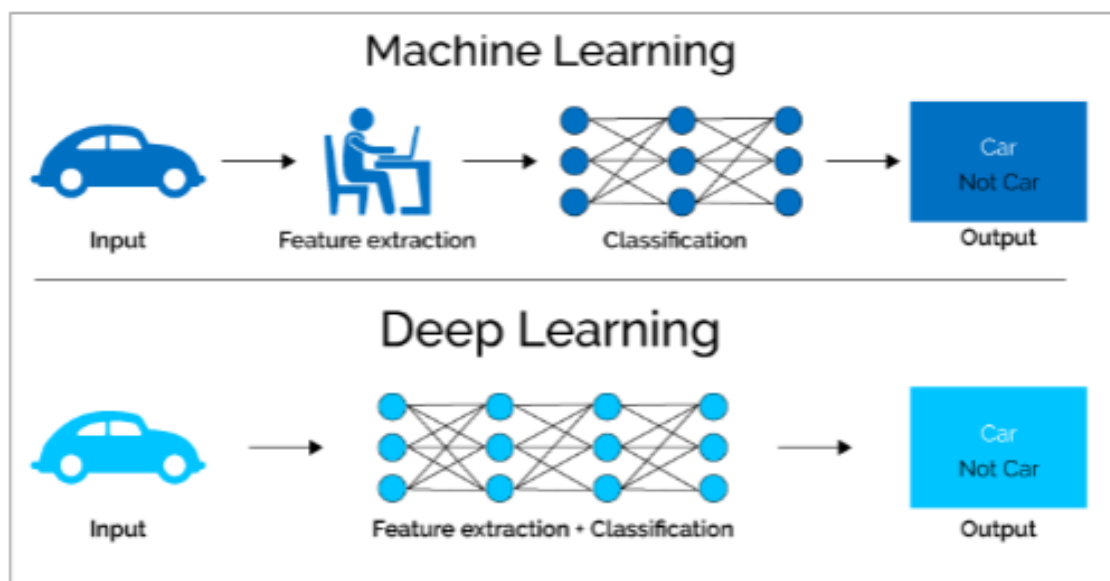
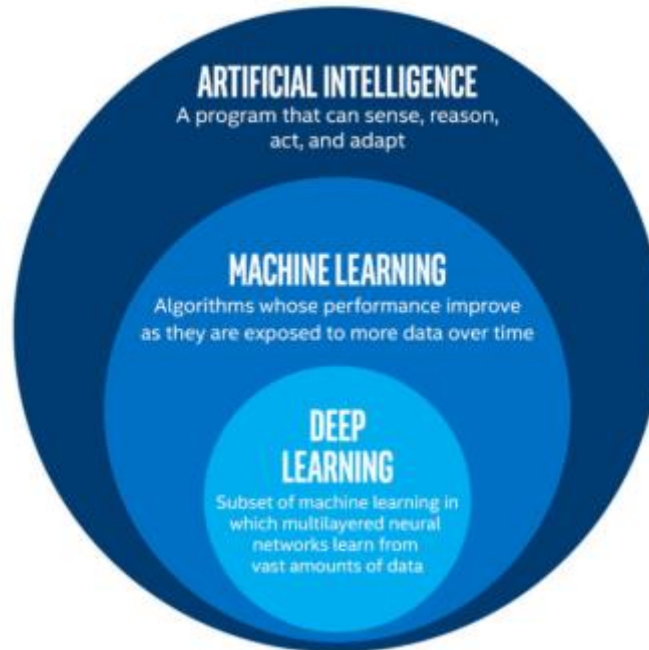


01 INTRODUCTION

MACHINE LEARNING OVERVIEW

Artificial Intelligence Vs Machine Learning



Note:

- Deep learning figure out features for themselves. Humans spend more time building the way the model works.

Different types of learning

- Unsupervised learning
 - Looks if the data has (variations = information)

- No variation, no information in it
 - Variable for variation = variance/standard deviation (SD)
- **Supervised learning** (focus of this course).
- Reinforcement learning

Many different types of Machine Learning aims:

- Clustering (unsupervised)
- Computer vision (detecting cancer in images, self-driving cars)
- Natural Language Processing (Lernout & Hauspi/Alexa/Siri)
- **Classification** (binary/multi-class)
 - Binary classification = focus of this course.
 - Supervised ML (Labels are known in learning phase).
 - Aims to predict labels of new data
 - Matrix format of data (rows x features)
- Recommender systems

Machine Learning example (see canvas notebooks)

- Read in the data → array of RGB colours.
- RGB matrix
- Put all pixels and align them after each other (not smart because it is massive, about 400,000 columns).



Figure: This is a flower

Multiple things can happen:

- Performance is good.
 - The model has learned something,
 - Did it learn the data by heart or did it learn the patterns?
 - Generalized → good.
 - Not generalized → try again.
 - **Overfit model** is a model that has learned the data by heart = when the outcome, model, is very good/If the model is the same as the data that means it is overfit and has learned patterns.

- Performance is bad.
 - The model didn't learn anything (use of all variables = too many).
 - Model does not work, try again (a new approach).

Steps of pipeline:

- Read = you get multiple matrixes
- Test/Train Split
- Data exploration
- Features
- Build the outcome vector (if not provided)

QUIZ NOTES

WHAT IS EXPLORATORY DATA ANALYSIS (EDA)?

Link: <https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15>.

Exploratory Data Analysis (EDA) refers to the critical process of performing initial investigations on data so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

- EDA is about understanding the data first and try to gather as many insights as possible from it.
- EDA is about making sense of data in hand, before getting them dirty with it.

EDA EXPLAINED USING WINE QUALITY DATA SET

STEP 1: Import necessary libraries such as pandas, numpy, matplotlib and seaborn.

```
In [1]: import matplotlib.image as img # To Load the images
import matplotlib.pyplot as plt # To plot the images

import copy # to copy variables
import numpy as np # To do some calculations
import pandas as pd # To work with dataframes (easier matrices)
from sklearn.ensemble import RandomForestClassifier # The machine Learning model
from os import listdir # To get a List of files in a folder
from sklearn.metrics import accuracy_score
```

STEP 2: Take a closer look at the data with the help of “.head()” function, which returns first five observations of the data set. Similarly “.tail()” function returns last five observations of the data set.

```
In [2]: df = pd.read_csv("01 Winequality White.csv", sep = ";")
df.head()
```

Out[2]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.0	0.27	0.36	20.7	0.045	45.0	170.0	1.0010	3.00	0.45	8.8	6
1	6.3	0.30	0.34	1.6	0.049	14.0	132.0	0.9940	3.30	0.49	9.5	6
2	8.1	0.28	0.40	6.9	0.050	30.0	97.0	0.9951	3.26	0.44	10.1	6
3	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6
4	7.2	0.23	0.32	8.5	0.058	47.0	186.0	0.9956	3.19	0.40	9.9	6

STEP 3: Find the total number of rows and columns in the data set using “.shape.”

- Dataset comprises of 4898 observations and 12 characteristics.
- One is dependent variable and the rest, 11, are independent variables – physico-chemical characteristics.

```
In [3]: df.shape
Out[3]: (4898, 12)
```

STEP 4: Know the columns and their corresponding data types, along with finding whether they contain null values or not.

- Data has only float and integer values.
- No variable column has null/missing values.

```
In [5]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4898 entries, 0 to 4897
Data columns (total 12 columns):
fixed acidity      4898 non-null float64
volatile acidity   4898 non-null float64
citric acid        4898 non-null float64
residual sugar     4898 non-null float64
chlorides          4898 non-null float64
free sulfur dioxide 4898 non-null float64
total sulfur dioxide 4898 non-null float64
density           4898 non-null float64
pH                4898 non-null float64
sulphates         4898 non-null float64
alcohol           4898 non-null float64
quality           4898 non-null int64
dtypes: float64(11), int64(1)
memory usage: 459.3 KB
```

STEP 5: Use “.describe()” function to get various summary statistics. This function returns the count, mean, standard deviation, minimum and maximum values, and the quantiles of the data.

```
In [6]: df.describe()
Out[6]:
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.360657	0.994027	3.188267	0.489847	10.514267
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.498065	0.002991	0.151001	0.114126	1.230621
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.000000	0.987110	2.720000	0.220000	8.000000
25%	6.300000	0.210000	0.270000	1.700000	0.036000	23.000000	108.000000	0.991723	3.090000	0.410000	9.500000
50%	6.800000	0.260000	0.320000	5.200000	0.043000	34.000000	134.000000	0.993740	3.180000	0.470000	10.400000
75%	7.300000	0.320000	0.390000	9.900000	0.050000	46.000000	167.000000	0.996100	3.280000	0.550000	11.400000
max	14.200000	1.100000	1.660000	65.800000	0.346000	289.000000	440.000000	1.038980	3.820000	1.080000	14.200000

- Mean value is less than median value of each column, which is represented by 50% (50th percentile) in index column.
- A large difference between 75%tile and max values of predictors “residual sugar,” “free sulphur dioxide,” and “total sulphur dioxide.”
- Thus, observations 1 and 2 suggests that there are extreme values outliers in our data set.

STEP 6: Key insights by looking at dependent variable.

```
In [7]: df.quality.unique()
```

```
Out[7]: array([6, 5, 7, 8, 4, 3, 9], dtype=int64)
```

- Target variable/dependent variable is discrete and categorical in nature.
- “quality” score scale ranges from 1 to 10, where 1 being poor and 10 being the best.
- 1, 2 & 10 “quality” ratings are not given by any observation. Only scores obtained are between 3 to 9.

STEP 7:

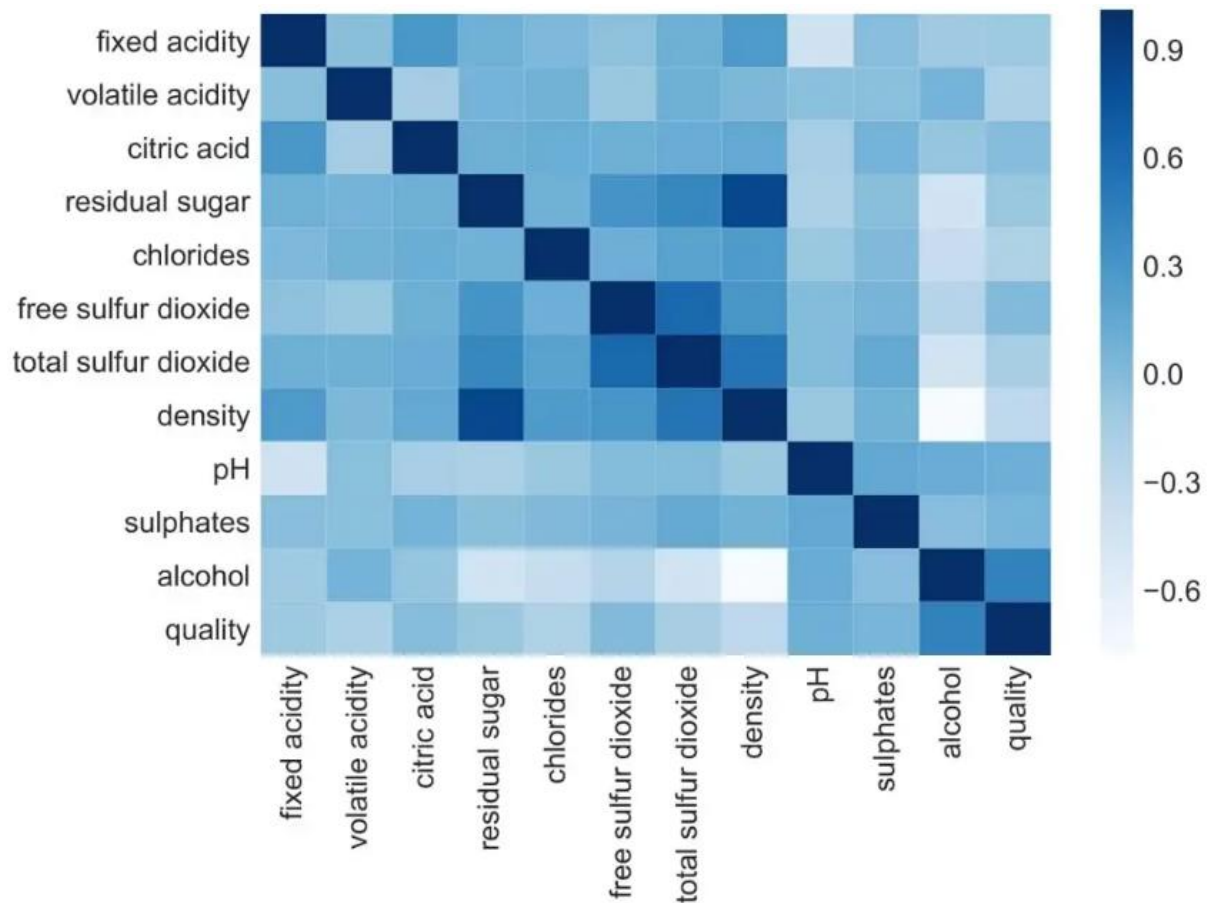
```
In [8]: df.quality.value_counts()
```

```
Out[8]: 6    2198
        5    1457
        7     880
        8     175
        4     163
        3       20
        9        5
        Name: quality, dtype: int64
```

- Vote count of each “quality” score is descending order.
- “quality” has most values concentrated in the categories 5, 6 and 7.
- Only a few observations made for the categories 3 & 9.

Seaborn is a visualization library that builds on top of matplotlib. Seaborn provides statistical graphs to perform both Univariate and Multivariate analysis.

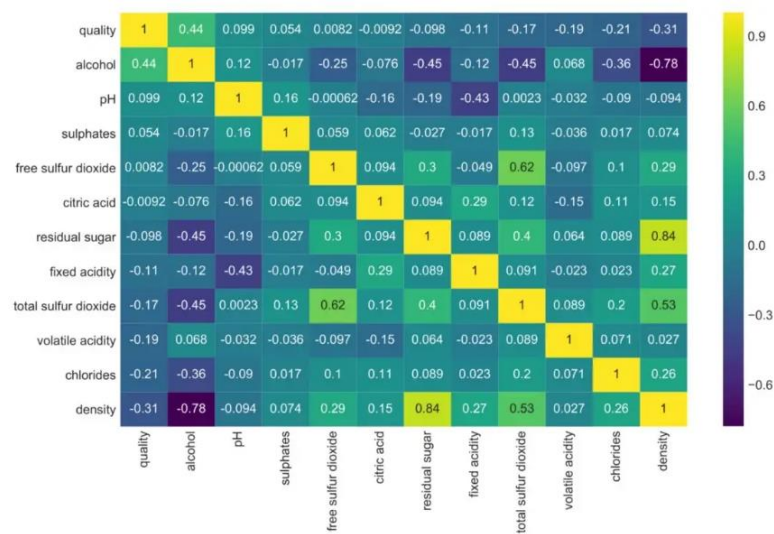
STEP 8: To use linear regression for modelling, remove correlated variables to improve your model. Use “.corr()” function to visualize the correlation matrix using a heatmap in seaborn.



Heatmap

- Dark shades represent positive correlation while lighter shades represent negative correlation.
- Set "**annot=True**" to get values by which features are correlated to each other in grid-cells.

STEP 9: Remove correlated variables during feature selection.



Correlation Matrix

- “density” has strong positive correlation with “residual sugar” whereas it has a strong negative correlation with “alcohol.”
- “Free sulphur dioxide” and “citric acid” has almost no correlation with “quality”.
- Since correlation is 0, we can infer there is no linear relationship between these 2 predictors. It is safe to drop these features in case you are applying Linear Regression model to the dataset.

STEP 10: A **box plot/box-and-whisker plot** shows the distribution of quantitative data in a way that facilitates comparisons between variables.

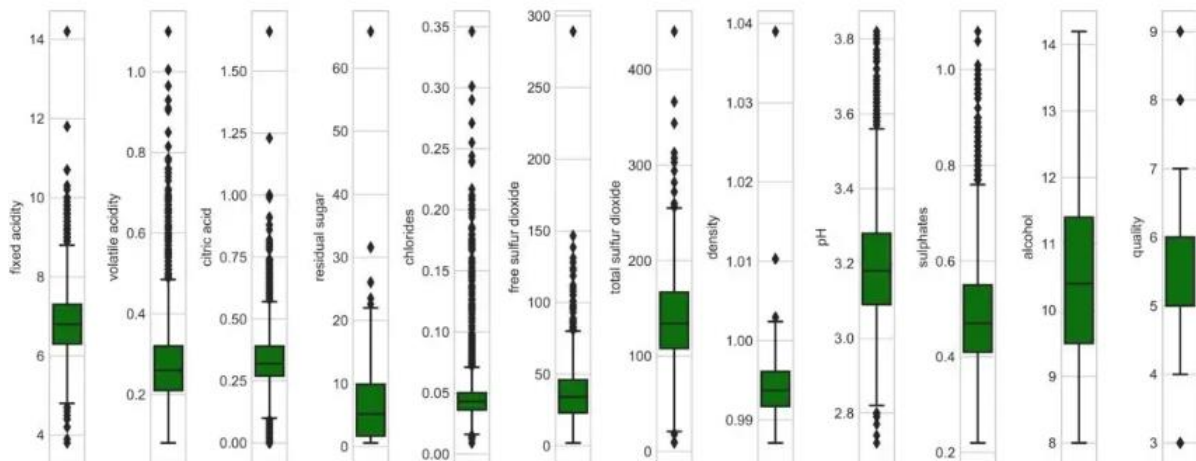
- The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution.

The box plot is a standardized way of displaying the distribution of data based on the 5 number summary:

- Minimum
- First quartile
- Median
- Third quartile
- Maximum

In the simplest box plot the central rectangle spans the first quartile to the third quartile (the interquartile range or IQR)

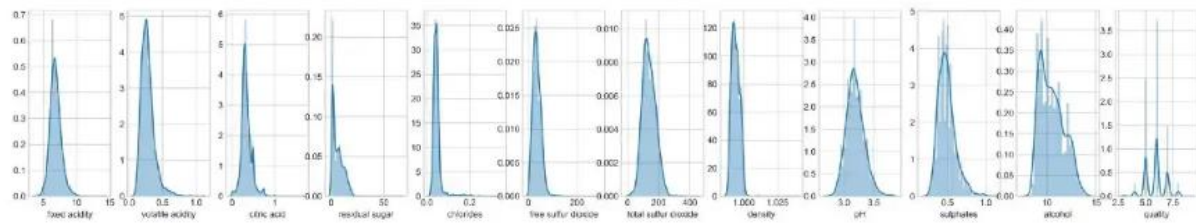
- A segment inside the rectangle shows the median and “whiskers” above and below the box show the locations of the minimum and maximum.



Boxplot

- Outliers are either 1.5xIQR or more above the third quartile or 1.5xIQR or more below the first quartile.
- Except “alcohol” all other features columns show outliers.

STEP 11: Check the linearity of the variables by plotting distribution graph and look for skewness of features. Kernel density estimate (kde) is quite useful tool for plotting the shape of a distribution.



Distribution Plot

- “pH” column appears to be normally distributed.
- Remaining all independent variables are right skewed/positively skewed.

“Exploratory Data Analysis is a philosophical and an artistic approach to gauge every nuance from the data at early encounter.”

WHAT IS EXPLORATORY DATA ANALYSIS (EDA)?

Link: <https://www.itl.nist.gov/div898/handbook/eda/section1/eda11.htm>.

APPROACH

Exploratory Data Analysis (EDA) is an approach/philosophy for data analysis that employs a variety of techniques (mostly graphical) to

- Maximize insight into a data set;
- Uncover underlying structure;
- Extract important variables;
- Detect outliers and anomalies;
- Test underlying assumptions;
- Develop parsimonious models; and
- Determine optimal factor settings.

FOCUS

The EDA approach is precisely that – an approach – not a set of techniques, but an attitude/philosophy about how a data analysis should be carried out.

PHILOSOPHY

EDA is not identical to statistical graphics although the two terms are used almost interchangeable.

- Statistical graphics is a collection of techniques - all graphically based and all focusing on one data characterization aspect.

- EDA encompasses a larger venue:
 - EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model.
 - EDA is not a mere collection of techniques.
 - EDA is a philosophy as to how we dissect a data set, what we look for, how we look, and how we interpret.

EDA heavily uses the collection of techniques that we call “statistical graphics,” but it is not identical to statistical graphics per se.

TECHNIQUES

Most EDA techniques are graphical in nature with a few quantitative techniques. Because, in combination with the natural pattern-recognition capabilities that we all possess, graphics provides unparalleled power to carry this out.

*“The **reason** for the heavy reliance on graphics is that by its nature the main role of EDA is to open-mindedly explore, and graphics gives the analysts unparalleled power to do so, enticing the data to reveal its structural secrets, and being always ready to gain some new, often unsuspected, insight into the data.”*

The graphical techniques employed in EDA are often quite simple, consisting of various techniques of:

- Plotting the raw data such as data traces, histograms, bihistograms, probability plots, lag plots, block plots, and Youden plots.
- Plotting simple statistics such as mean plots, standard deviation plots, box plots, and main effects plots of the raw data.
- Positioning such plots so as to maximize our natural pattern-recognition abilities, such as using multiple plots per page.

EXPLORATORY DATA ANALYSIS

Link: <https://r4ds.had.co.nz/exploratory-data-analysis.html>. (This is in R, ideas still hold for python where you can make similar plots).

7.1: INTRODUCTION

Exploratory Data Analysis (EDA) use visualization and transformation to explore data in a systematic way. EDA is an iterative cycle, you:

- Generate questions about the data.
 - Investigate every idea/question to know the quality of the data
- Search for answers by visualizing, transforming, and modelling your data (= you are deploying the tools of EDA to be able to do data cleaning).
 - Some of the ideas will pan out and some will be dead ends.
- Use what you learn to refine your questions/generate new questions.
 - Home in on a few particularly productive areas that you’ll eventually write up and communicate to others.

7.2: QUESTIONS

The goal during EDA is to develop an understanding of the data by having questions, which are tools to guide your investigation.

- The question focuses your attention on a specific part of your dataset and helps you decide which graphs, models, or transformations to make.
- The key to asking quality questions is to generate a large quantity of questions.
 - Each new question that you ask will expose you to a new aspect of your data and increase your chance of making a discovery.
 - You can drill down into the most interesting parts of your data – and develop a set of thought-provoking questions – if you follow up each question with a new question based on what you find.
- The 2 types of questions will always be useful for making discoveries within your data.
 - What type of variations occur within my variables?
 - What type of covariation occurs between my variables?

A **variable** is a quantity, quality, or property that you can measure.

A **value** is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.

An **observation** is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object).

- An observation will contain several values, each associated with a different variable. Sometimes an observation is referred to as a data point.

Tabular data is a set of values, each associated with a variable and an observation,

- Tabular data is tidy if each value is placed in its own “cell,” each variable in its own column, and each observation in its own row.

7.3: VARIATION

Variation is the tendency of the values of a variable to change from measurement to measurement.

If you measure any **continuous variable** twice, you will get two different results. Even if you measure quantities that are **constant**, like the speed of light, each of your measurements will include a small amount of error that varies from measurement to measurement. **Categorical variables** can also vary if you measure across different subjects, for example, the eye colours of different people, or different times, for example, the energy levels of an electron at different moments.

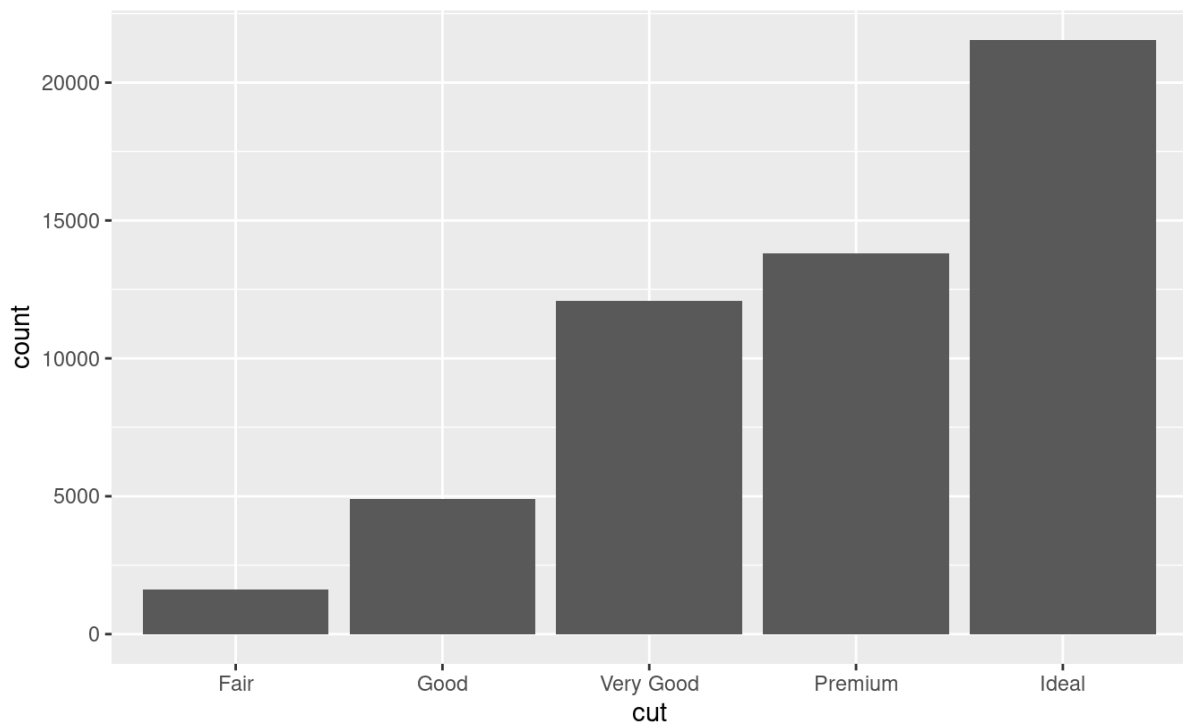
- Every variable has its own pattern of variation, which can reveal interesting information. Visualize the distribution of the variable’s values to understand that pattern.

7.3.1: VISUALIZING DISTRIBUTIONS

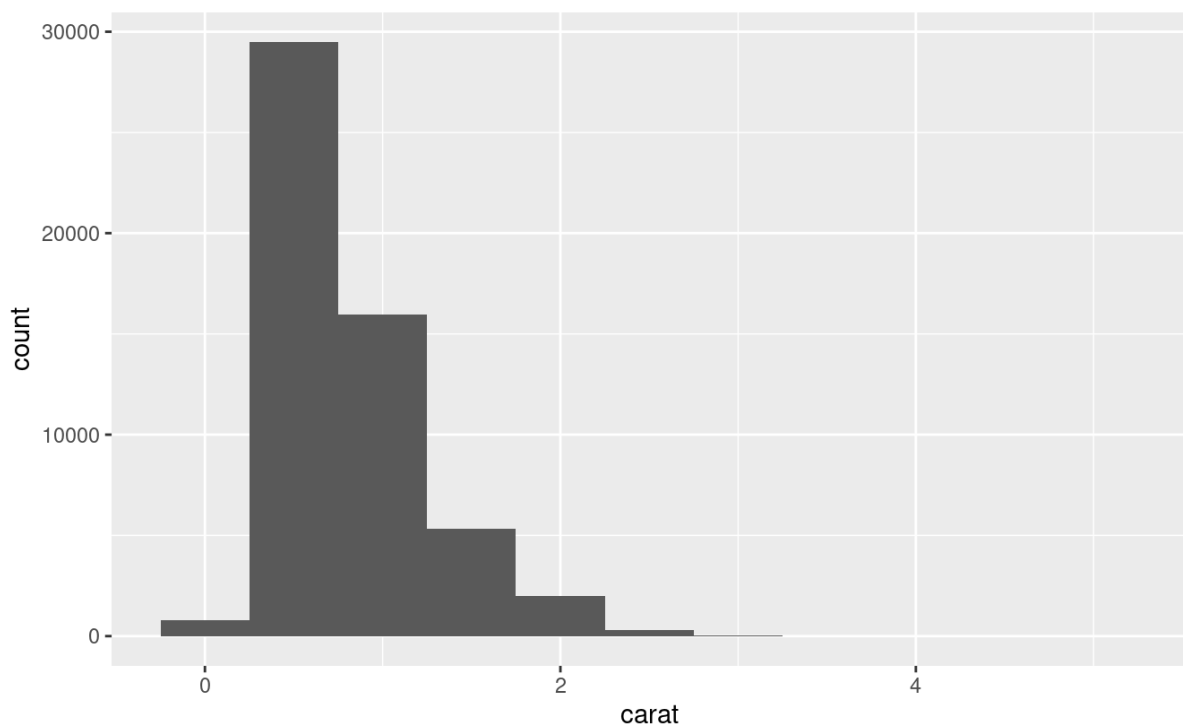
How you visualise the distribution of a variable will depend on whether the variable is categorical or continuous.

A variable is **categorical** if it can only take one of a small set of values. In R, categorical variables are usually saved as factors or character vectors. To examine the distribution of a categorical variable, use a **bar chart**:

- The height of the bars displays how many observations occurred with each x value.

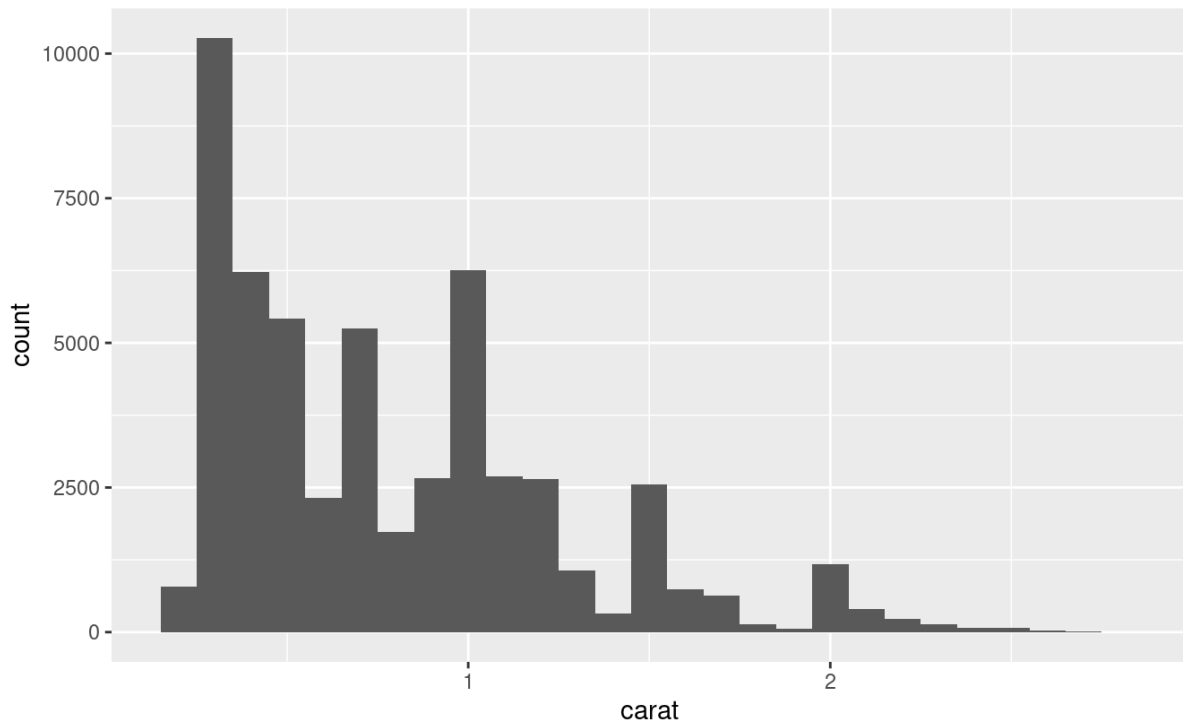


A variable is **continuous** if it can take any of an infinite set of ordered values. Numbers and date-times are two examples of continuous variables. To examine the distribution of a continuous variable, use a **histogram**:

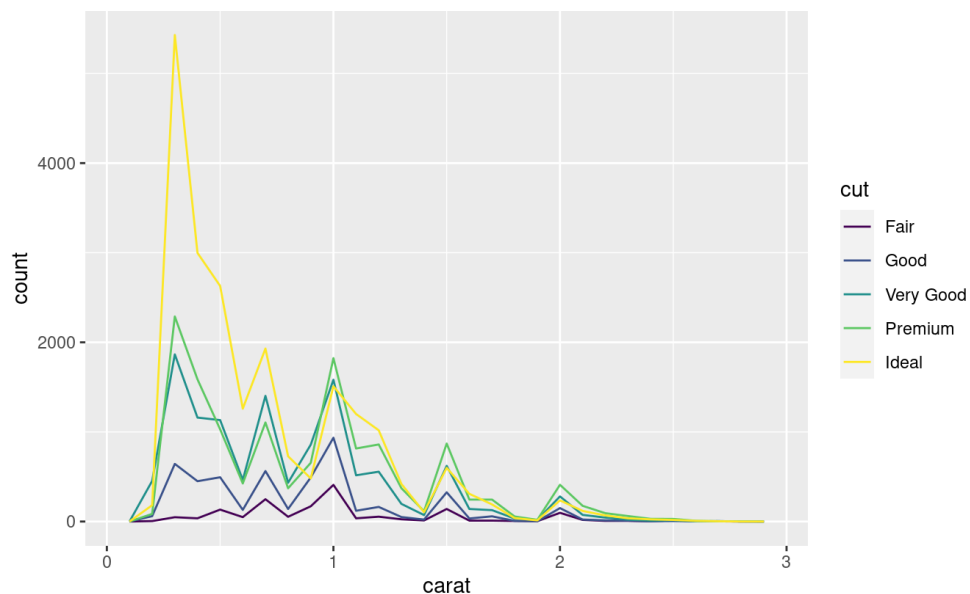


A histogram divides the x-axis into equally spaced bins and then uses the height of a bar to display the number of observations that fall in each bin.

- In the graph above, the tallest bar shows that almost 30,000 observations have a carat value between 0.25 and 0.75, which are the left and right edges of the bar.
- You should explore a variety of binwidths when working with histograms, as different binwidths can reveal different patterns.
 - For example, here is how the graph above looks when we zoom into just the diamonds with a size of less than three carats and choose a smaller binwidth.



If you wish to overlay multiple histograms in the same plot, I recommend instead of displaying the counts with bars, use lines instead. It's much easier to understand overlapping lines than bars.



Now that you can visualise variation, what should you look for in your plots? And what type of follow-up questions should you ask?

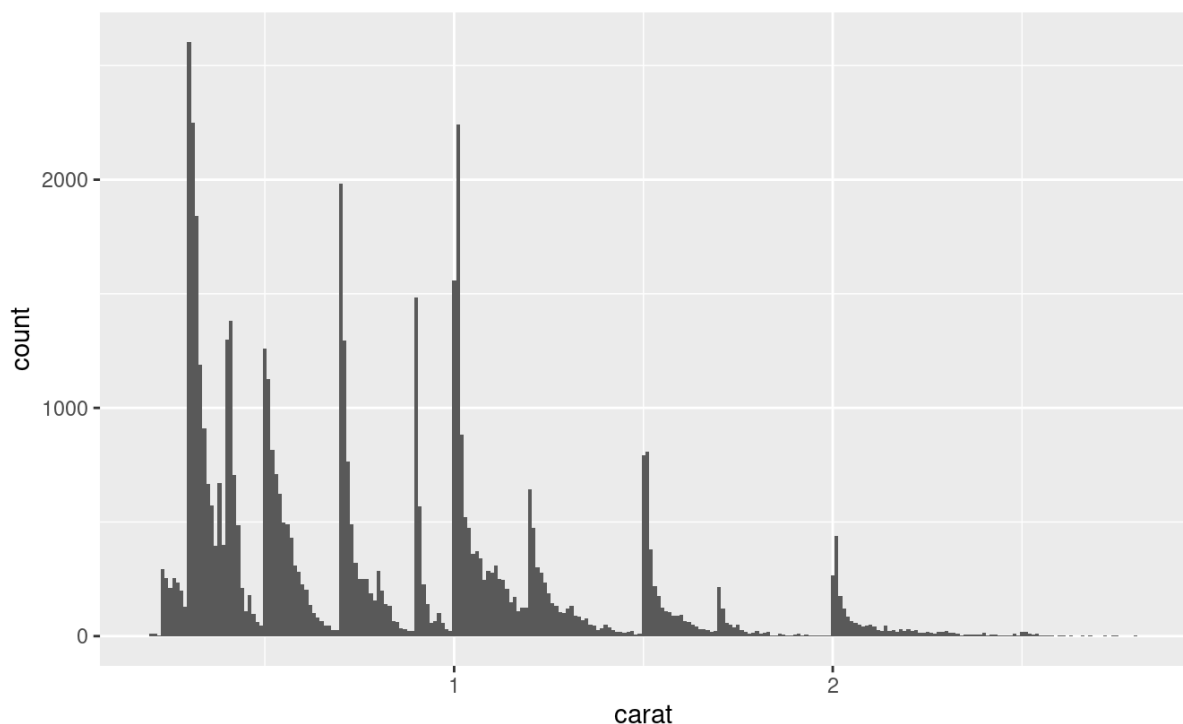
7.3.2: TYPICAL VALUES

In both bar charts and histograms, tall bars show the common values of a variable, and shorter bars show less-common values. Places that do not have bars reveal values that were not seen in your data. To turn this information into useful questions, look for anything unexpected:

- Which values are the most common? Why?
- Which values are rare? Why? Does that match your expectations?
- Can you see any unusual patterns? What might explain them?

As an example, the histogram below suggests several interesting questions:

- Why are there more diamonds at whole carats and common fractions of carats?
- Why are there more diamonds slightly to the right of each peak than there are slightly to the left of each peak?
- Why are there no diamonds bigger than 3 carats?



Clusters of similar values suggest that subgroups exist in your data. To understand the subgroups, ask:

- How are the observations within each cluster similar to each other?
- How are the observations in separate clusters different from each other?
- How can you explain or describe the clusters?
- Why might the appearance of clusters be misleading?

Many of the questions above will prompt you to explore a relationship *between* variables, for example, to see if the values of one variable can explain the behaviour of another variable.

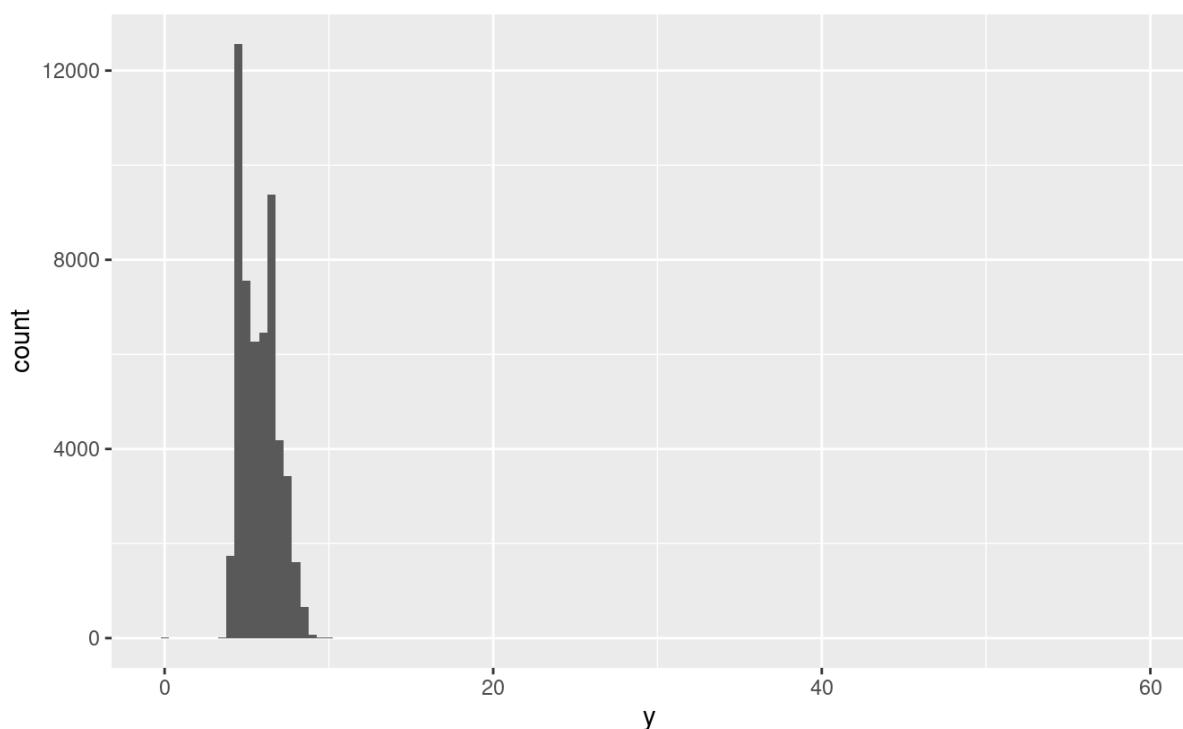
7.3.2: UNUSUAL VALUES

Outliers are observations that are unusual.

- data points that don't seem to fit the pattern.
- Sometimes outliers are data entry errors.
- other times outliers suggest important new science.

When you have a lot of data, outliers are sometimes difficult to see in a histogram. For example, take the distribution of the *y* variable from the diamonds dataset.

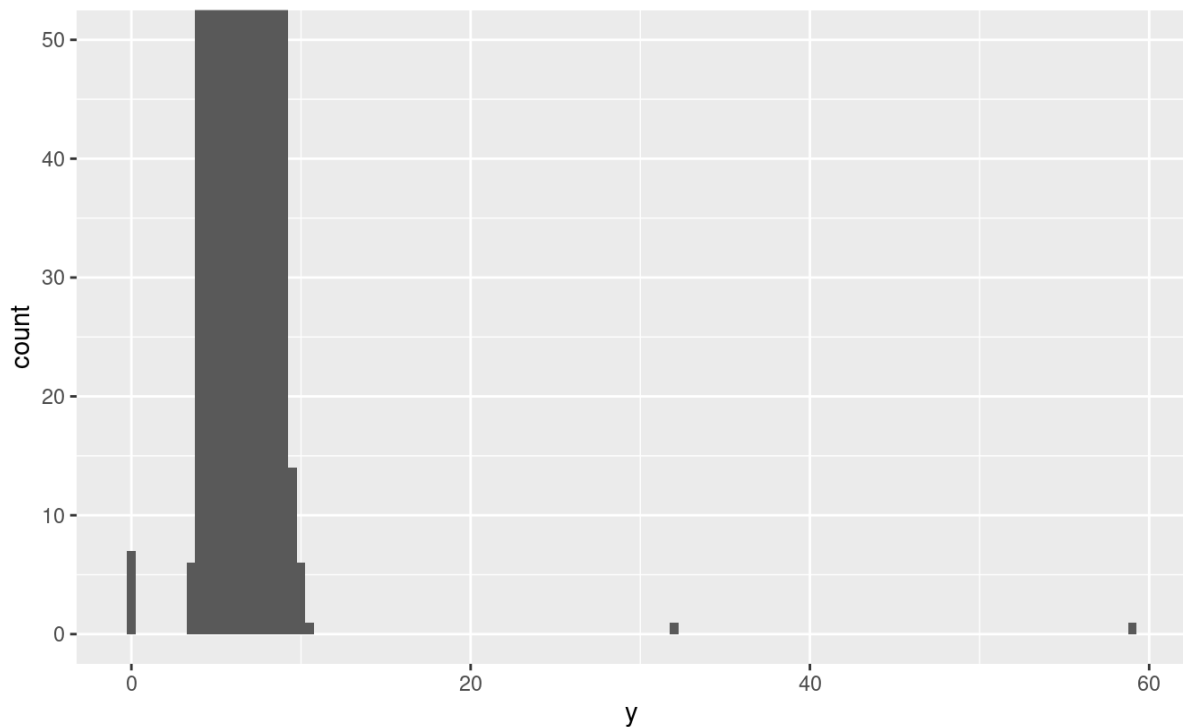
- The only evidence of outliers is the unusually wide limits on the x-axis.



Observations:

- There are so many observations in the common bins that the rare bins are so short that you can't see them (although maybe if you stare intently at 0, you'll spot something).

Zooming in to see the unusual/small values of the y-axis.



Observations:

- This allows us to see that there are three unusual values: 0, ~30, and ~60. We pluck them out.
- The y variable measures one of the three dimensions of these diamonds, in mm. We know that diamonds can't have a width of 0mm, so these values must be incorrect.
- We might also suspect that measurements of 32mm and 59mm are implausible: those diamonds are over an inch long, but don't cost hundreds of thousands of dollars!

It's good practice to repeat your analysis with and without the outliers.

- If they have minimal effect on the results, and you can't figure out why they're there, it's reasonable to replace them with missing values, and move on.
- However, if they have a substantial effect on your results, you shouldn't drop them without justification. You'll need to figure out what caused them (e.g., a data entry error) and disclose that you removed them in your write-up.

7.4: MISSING VALUES

If you've encountered unusual values in your dataset, and simply want to move on to the rest of your analysis, you have two options.

- Drop the entire row with the strange values:
 - Not recommended because just because one measurement is invalid, doesn't mean all the measurements are.
 - Additionally, if you have low quality data, by time that you've applied this approach to every variable you might find that you don't have any data left!
- Instead, replace the unusual values with missing values.

7.5: COVARIATION

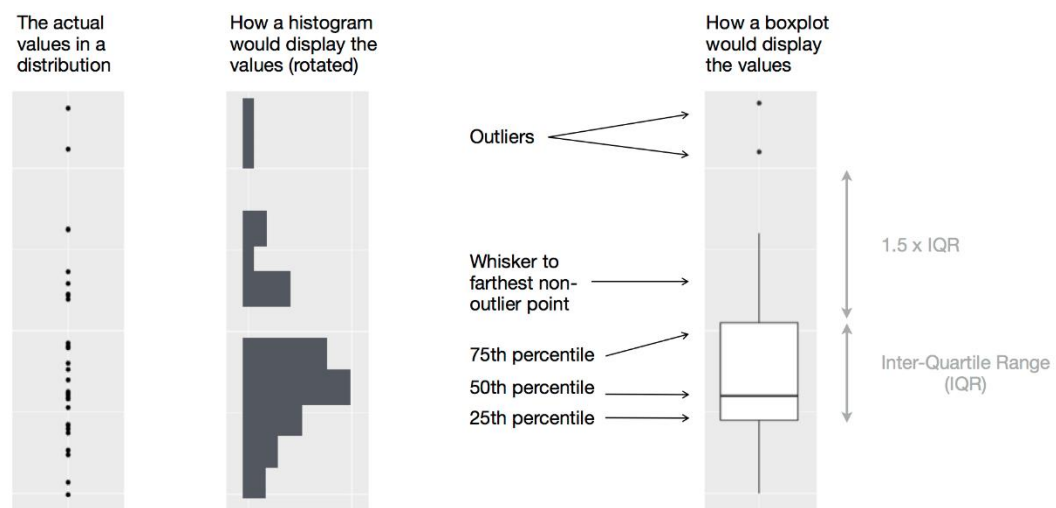
If **variation** describes the **behaviour within a variable**, **covariation** describes the **behaviour between variables**.

Covariation is the tendency for the values of two or more variables to vary together in a related way. The best way to spot covariation is to visualise the relationship between two or more variables. How you do that should again depend on the type of variables involved.

7.5.1: A CATEGORICAL & CONTINUOUS VARIABLE

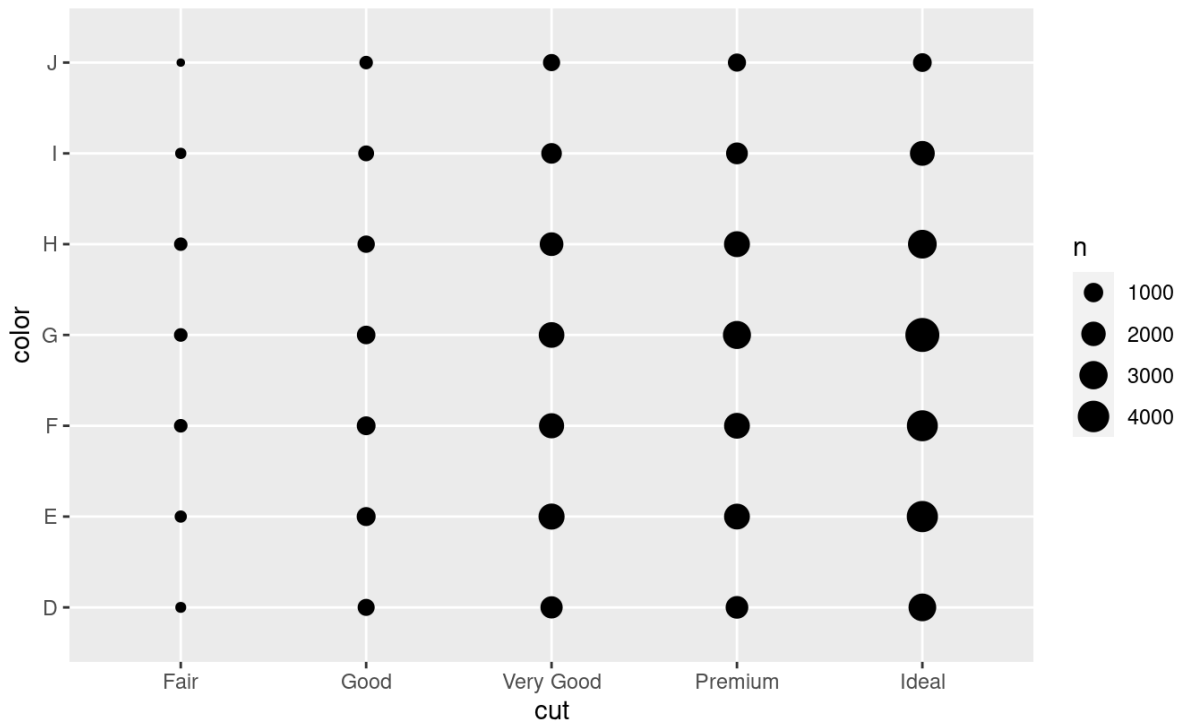
To display the distribution of a continuous variable broken down by a categorical variable is the boxplot. A **boxplot** is a type of visual shorthand for a distribution of values that is popular among statisticians. Each boxplot consists of:

- A box that stretches from the 25th percentile of the distribution to the 75th percentile, a distance known as the interquartile range (IQR). In the middle of the box is a line that displays the median, i.e. 50th percentile, of the distribution. These three lines give you a sense of the spread of the distribution and whether or not the distribution is symmetric about the median or skewed to one side.
- Visual points that display observations that fall more than 1.5 times the IQR from either edge of the box. These outlying points are unusual so are plotted individually.
- A line (or whisker) that extends from each end of the box and goes to the farthest non-outlier point in the distribution.



7.5.2: TWO CATEGORICAL VARIABLES

To visualise the covariation between categorical variables, you'll need to count the number of observations for each combination.

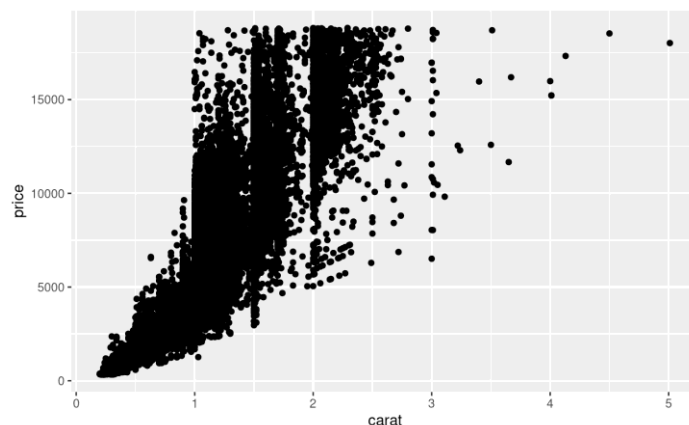


The size of each circle in the plot displays how many observations occurred at each combination of values. Covariation will appear as a strong correlation between specific x values and specific y values.

7.5.3: TWO CONTINUOUS VARIABLES

You've already seen one great way to visualise the covariation between two continuous variables: draw a **scatterplot**.

- You can see covariation as a pattern in the points. For example, you can see an exponential relationship between the carat size and price of a diamond.

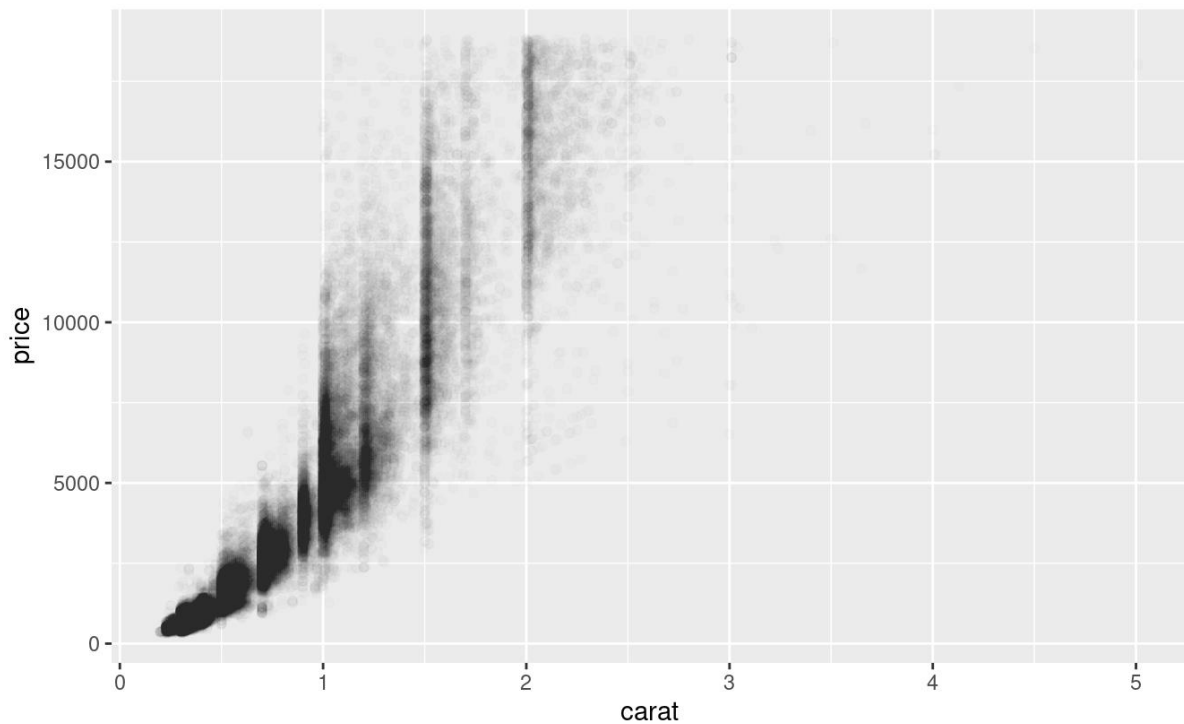


Disadvantage of a scatterplot

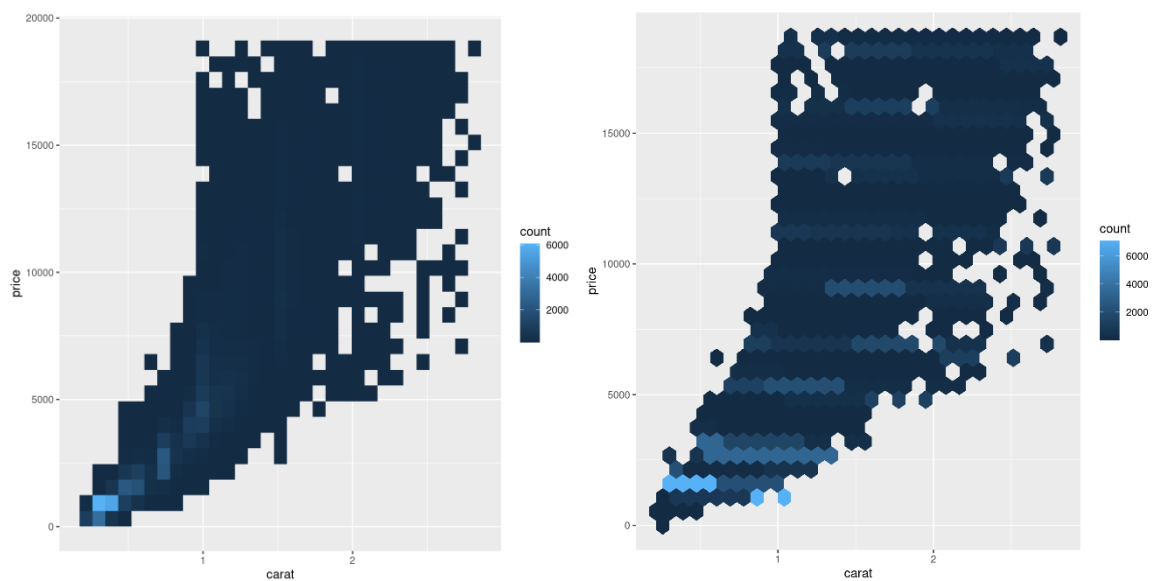
- Scatterplots become less useful as the size of your dataset grows, because points begin to overplot, and pile up into areas of uniform black (as above).

How to fix the disadvantage

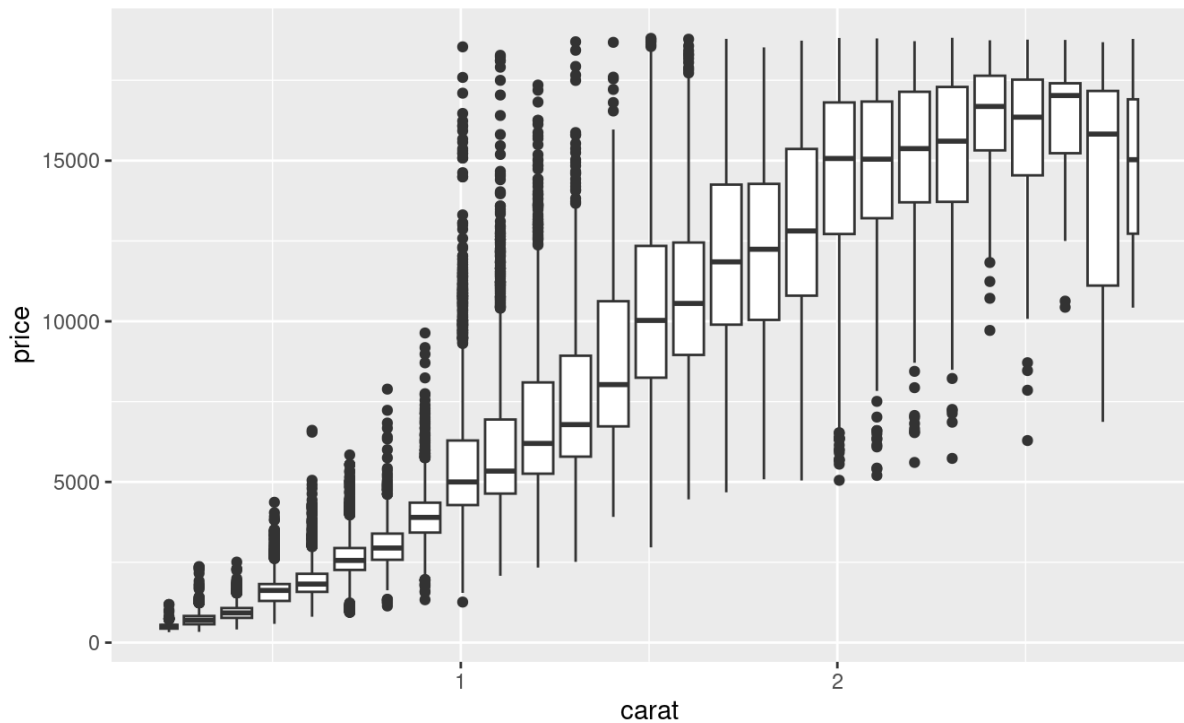
- use the alpha aesthetic to add transparency. But using transparency can be challenging for very large datasets.



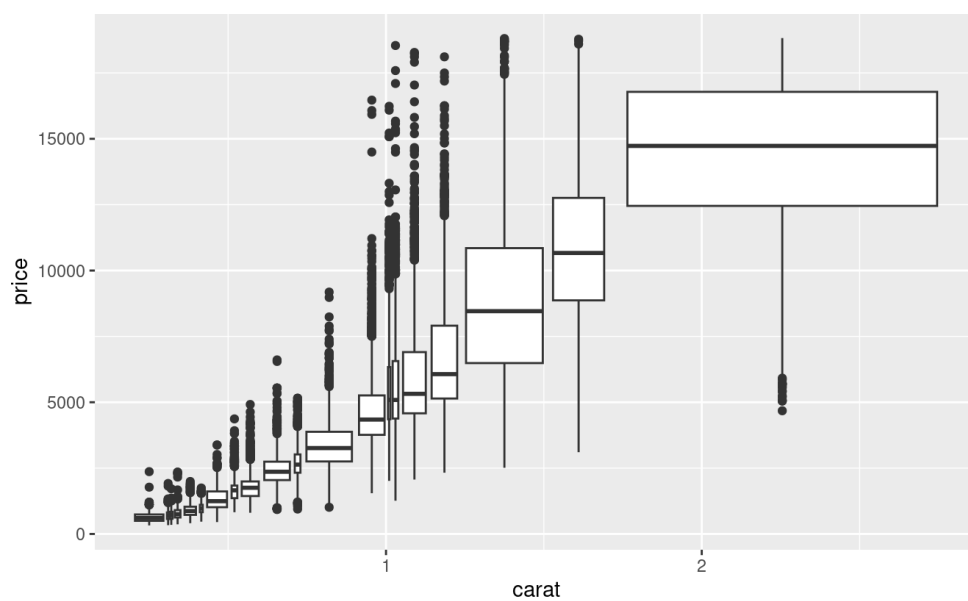
- Another solution is to use bin in two dimensions.



- Another option is to bin one continuous variable, so it acts like a categorical variable. Then you can use one of the techniques for visualising the combination of a categorical and a continuous variable that you learned about. For example, you could bin carat and then for each group, display a boxplot:



- By default, boxplots look roughly the same (apart from number of outliers) regardless of how many observations there are, so it's difficult to tell that each boxplot summarises a different number of points. One way to show that is to make the width of the boxplot proportional to the number of points. Another approach is to display approximately the same number of points in each bin.



7.6: PATTERNS & MODELS

Patterns in your data provide clues about relationships. If a systematic relationship exists between two variables it will appear as a pattern in the data. If you spot a pattern, ask yourself:

1. Could this pattern be due to coincidence (i.e., random chance)?
2. How can you describe the relationship implied by the pattern?
3. How strong is the relationship implied by the pattern?
4. What other variables might affect the relationship?
5. Does the relationship change if you look at individual subgroups of the data?

Patterns provide one of the most useful tools for data scientists because they **reveal covariation**.

- If you think of **variation** as a phenomenon that **creates uncertainty**, **covariation** is a phenomenon that **reduces it**.
 - If two variables covary, you can use the values of one variable to make better predictions about the values of the second.
 - If the covariation is due to a causal relationship (a special case), then you can use the value of one variable to control the value of the second.

Models are a tool for extracting patterns out of data.

02 DATA PROCESSING & EDA

QUIZ 1

What does EDA stand for?

- Exploratory Data Analysis

If you have a variable, what could you do to see whether there are any outliers? Give 2-3 possibilities.

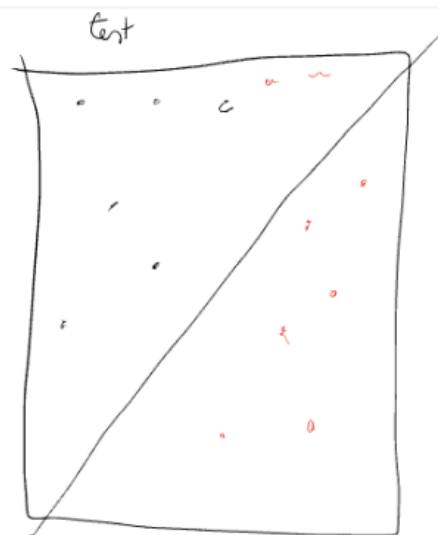
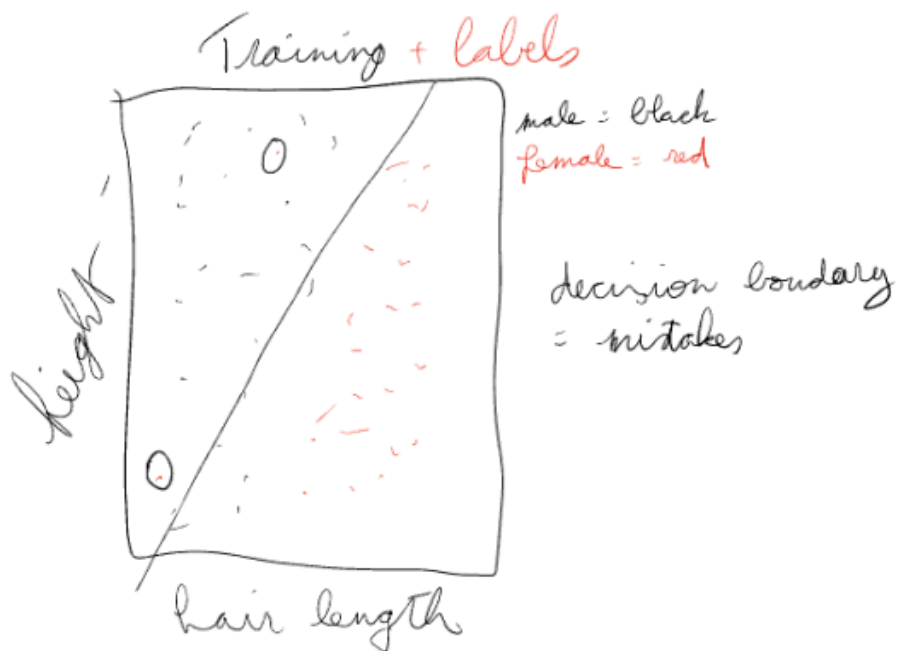
- Calculate the mean and standard deviation of the variable, identify any data points that fall outside of a certain number of standard deviations from the mean. For example, data points that fall more than 3 standard deviations from the mean could be considered outliers.
- Use box plots or scatter plots to visually identify outliers.
- Use the interquartile range (IQR) to identify outliers, which is the difference between the 75th and 25th percentiles. Data points that fall outside of the range of $(Q1 - 1.5 * IQR)$ to $(Q3 + 1.5 * IQR)$ are considered outliers.

What is the difference between correlation and covariation?

- Covariance shows you how the two variables differ, whereas correlation shows you how the two variables are related.

RECAP

- Machine learning default workflow
- Data format:
 - Matrix format
 - No missing value (some ML algorithms can cope with this).
- Data splitting → Train data & Testing data
- Decision boundaries



→ if you fit first, you re-lay the decision lines → it will take the new data as decision line

Why is a model not learning?

- Problem is too difficult for the model I am using.
 - Patterns aren't in the data => clean data
 - Increase model complexity.
- Go back to the beginning.

Why is a model very good?

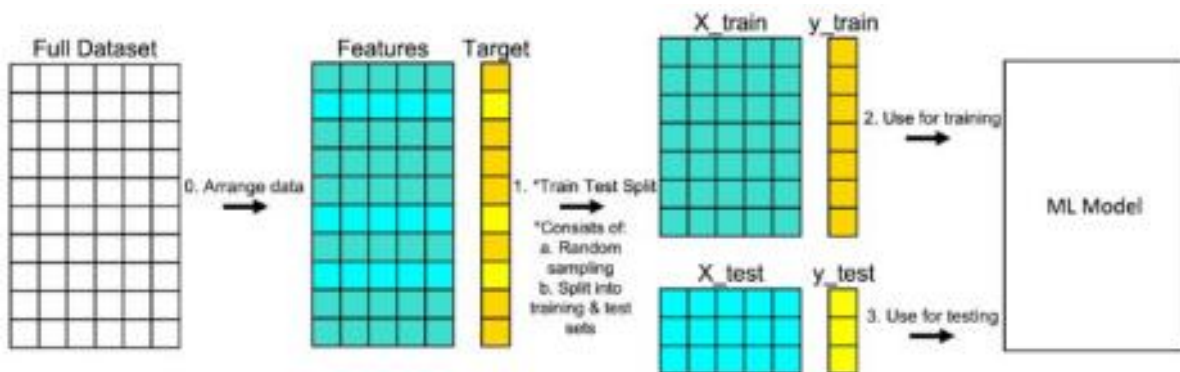
- The data is overfit, the model has learned the data by heart.
 - Check by training and test (if it's good on the training data and bad on tests data, it is overfitted).
 - Don't give it the outcome!
- It has actually learned the patterns that are in the data (the model is generalisable).

Model

- You only fit the model once and make generalizations from there forth.
- **Never fit the test data → you will shift the decision line to the new data.**

ML default pipeline

- Matrix format
- Data splitting → training part and testing part.



EDA

- A bit like an interrogation:
 - You ask question and the data answers.
 - Count (how much of each), shape of the data, correlation, ... → what data do we need to answer the question and is the data clean enough.
 - Ask stupid questions and get stupid answers.
 - Ask the right questions and you get deeper to the truth.

What is the quality of my data? Missing values, faulty data (wrong measurement, ...) and so on → find the good parts.

- Goal is to get to the core of the data.
- The core is dependent on the question we want to solve.
- The core is hidden by problems.
 - Missing values
 - Strange/unknown variables
 - Faulty measurements
 - Correlated variables (not this lesson)

DATA TYPES

MEASUREMENT TYPES

Remember the Data Processing & Analysis course?

- Nominal data
 - is in text: French, Dutch → dummy coding is a solution: one-hot encoding → make a column for each (column French, Dutch) → 1 if you are Dutch, 0 if you are not. But it had a lot of columns.
- Ordinal data
 - Isn't numeric, it is text but there is an order (height: short, average, tall) → you can put this in numbers.
- Interval data
 - A bit less quality, also numbers.
- Ratio data
 - Makes sense, if you are 1m and the other 2m, the other one is twice as high.

MEASUREMENT TYPES - PROCESSING

Not all measurement/data types can be used 'as is' for machine learning. Some need some processing.

- Nominal data is tricky for most ML algorithms (work internally with numbers). One-hot encoding possible solution.
- Ordinal data can be converted to a number.

Interval and Ratio data can often be used to create more data with pre-processing and feature engineering.

MISSING VALUES

MISSING DATA

Different types of missing data

- Missing completely at random (MCAR) → an accident.

- Missing at random (MAR) → underlying process which allows for more values missing but is still ok.
- Missing not at random (MNAR/NMAR) → issue! There is a clear process that stops me from having data there, for example, the range of the sensor cant go so far.



What you can do:

- Carry last value forward → now count how many times you are above the highest value.

MISSING DATA STRATEGIES

Below are some missing data strategies from least to most complex.

- Remove the data (complete case analysis)
 - Remove rows.
 - Remove columns.
- Impute with a default value:
 - 0 (really basic)
 - Mean/median (better in certain situation: there are no extremes)
 - NA
 - 999/-999 (if you ever do this I will come and kill you)
- Impute with a realistic guess!
- Build a machine learning model to impute the values (e.g., missforest).

CONCLUSION

EXERCISE

By next lesson, figure out how to impute data and do complete case analysis in python/pandas.

Impute with a default value:

- 0 (really basic).
- Mean/median (better in certain situation)

- NA
- 999/-999 (if you even do this I will come and kill you)

Also, exercise your python skills.

04/05 PERFORMANCE EVALUATION

Outline

- Train/test
- ML output
- Confusion matrix
- Performance metrics
- Performance plots
- Cross-validation (CV)

TRAIN/TEST

Take a part of our data because we don't want it in our model because we want to evaluate our model on data it hasn't seen before to test its generalisability. Training data has produced a model.

ML OUTPUT

In:

- Test data (10 values)

Out:

- 2 ways it can output (you choose this): 10 labels or 10 numbers between 0 (sure its false) and 1 (sure its true) → it calculates the probability.

T	0.99	T	0.9
F	0.01	F	0.3
F	0.01	F	0.1
T	0.99	T	0.6
T	0.99	T	0.8

CONFUSION MATRIX

		Predicted	
		Positive (1)	Negative (0)
Truth	Positive (1)	True Positive (TP)	False Negative (FN)
	Negative (0)	False Positive (FP)	True Negative (TN)

Table: Confusion matrix outline

model output	by label
H	H
H	H
D	D
H	D
D	D
D	H
H	H

Count each one (predicted & truth)

2 true positive	1 false negative
1 false positive	3 true negative

Sensitivity of model/recall

$$\frac{TP}{P} = \frac{TP}{TP + FN} = \frac{2}{2 + 1} = \frac{2}{3}$$

- Only takes $\frac{2}{3}$ rd into account → how many of the ones we want to detect do we detect?

Selectivity/specificity

$$\frac{TN}{N} = \frac{TN}{TN + FP} = \frac{3}{3 + 1} = \frac{3}{4}$$

- How specific is my model?

Precision

$$\frac{TP}{TP + FP} = \frac{2}{2 + 1} = \frac{2}{3}$$

EXAMPLE

- 10 cancer patients
- 990 healthy

MODEL 1 → ALL HAVE CANCER

10 TP	0 FN
990 FP	0 TN

sensitivity:

$$10/(10+0) = 100\%$$

specificity =

$$0/(0+990) = 0\%$$

precision =

$$10/(10+990) = 1\%$$

MODEL 2 → NO ONE HAS CANCER

MODEL 2 → no one has cancer:

0 TP	10 FN
0 FP	990 TN

sensitivity:

$$0/(0+10) = 0\%$$

specificity =

$$100\%$$

precision =

$$0$$

MODEL 3 → ALL CANCERS AT THE COST OF 300 FP

10 TP	0 FN
300 FP	690 TN

sensitivity:

$$100\%$$

specificity =

$$690/(690+300) = 69\%$$

precision =

$$3.3\%$$

MODEL 4 → 1000 CANCER PATIENT

600 TP	400 FN
300 FP	700 TN

sensitivity =
 $600/(600+400) = 60\%$

specificity =
 $700/(700+300) = 70\%$

precision =
 $600/(600+300) = 66\%$

UNDERSTANDING CONFUSION MATRIX

Link: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.

When we get the data, after data cleaning, pre-processing, and wrangling, the first step we do is to feed it to an outstanding model and of course, get output in probabilities. But hold on!

- How in the hell can we measure the effectiveness of our model. Better the effectiveness, better the performance → where the Confusion matrix comes into the limelight.

The following content answers the following questions:

- What the confusion matrix is and why you need it?
- How to calculate Confusion Matrix for a 2-class classification problem?





WHAT IS CONFUSION MATRIX & WHY YOU NEED IT?

Confusion matrix is a performance measurement for machine learning classification problem where output can be two or more classes.

- It is extremely useful for measuring Recall, Precision, Specificity, Accuracy, and AUC-ROC curves.
- It is a table with 4 different combinations of predicted and actual values.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Let's understand TP, FP, FN, TN in terms of pregnancy analogy.

		Actual Values	
		1	0
Predicted Values	1	TRUE POSITIVE 	FALSE POSITIVE 
	0	FALSE NEGATIVE 	TRUE NEGATIVE 

Note:

- **True positive** = you predicted positive and it's true.
 - You predicted that a woman is pregnant, and she actually is.
- **False Positive** (Type 1 error) = you predicted positive and it's false.
 - You predicted that a man is pregnant but he actually is not.
- **False negative** (Type 2 error) = you predicted negative and it's false.
 - You predicted that a woman is not pregnant but she actually is.
- **True negative** = you predicted negative and its true.
 - You predicted that a man is not pregnant, and he actually is not.

Just remember, we describe:

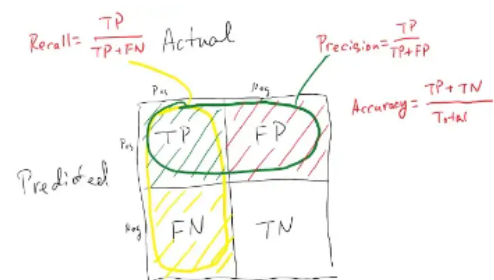
- **Predicted values** as **Positive & Negative**.
- **Actual values** as **True & False**.



HOW TO CALCULATE CONFUSION MATRIX FOR A 2-CLASS CLASSIFICATION PROBLEM?

Let's understand the confusion matrix through math.

y	y pred	output for threshold 0.6	Recall	Precision	Accuracy
0	0.5	0	1/2	2/3	4/7
1	0.9	1			
0	0.7	1			
1	0.7	1			
1	0.3	0			
0	0.4	0			
1	0.5	0			



Recall = from all the positive classes, how many we predicted correctly.

- Recall should be high as possible.
- To do a better job at recall, is to be more aggressive about saying "pregnant" even for small signs/symptoms that shows a person is pregnant.

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

Precision = from all the classes we have predicted as positive, how many are actually positive.

- Precision should be high as possible.
- To be really precise, is to say "pregnant" when it is absolutely sure that the person is pregnant → raising our classification threshold.

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

Note:

- whenever someone tells you what the precision value is, you need to also ask about the recall value before you can say anything about how good the model is and vice versa.
- Precision and recall are both well defined when there is one specific classification threshold.

Accuracy = from all the classes (positive and negative), how many of them we have predicted correctly.

- Accuracy should be high as possible.
- Accuracy has some key flaws.
 - Breaks down when we have class imbalance, when positives or negatives are extremely rare. For example, use accuracy to assess the quality of a model that is predicting ad click-through rates for display ads. In display ads, our click-through rates are often 1 in 1,000, 1 in 10,000 or even lower.
 - So, a model with absolutely no features in it except for a bias feature that tells it to predict false, always.
 - This predict false always model would have an accuracy of 99.999% in display ads predictions, but would add absolutely no value.

F-measure/F-score helps to measure recall and precision at the same time when it is difficult to compare 2 models with low precision and high recall or vice versa.

- It uses Harmonic Mean in place of Arithmetic Mean by punishing the extreme values more.

$$F - measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

Prediction bias is defined by taking the sum of all of the things that we predict and comparing them to the sum of all the things we observe.

- Preferred that the expected values that we predict to be equal to the observed values.
 - If they are not, we say that the model has some bias.
 - A bias of 0 would show that the sum of the predictions equals the sum of the observations.
- Bias is very simplistic metric in that it's easy to fool.
 - We could have a model that has almost no value to it, it just predicts the mean of all the class probabilities to create a zero bias model. But just having zero bias by itself is not an indicator that the model is perfect, we need to keep looking at other metrics for that.
- However, it's a useful canary because if one or more complicated models does not have zero bias, it means that something is going on. It gives us something to dig into to debug our models.

PERFORMANCE METRICS

Link: https://en.wikipedia.org/wiki/Confusion_matrix.

PERFORMANCE PLOTS

T	0.99	T	0.9
F	0.01	F	0.3
F	0.01	F	0.1
T	0.99	T	0.6
T	0.99	T	0.8

Quantify this performance:

3 TP	0 FN
0 FP	2 TN

=best model → no mistakes

When bar (decision threshold) goes from 0.5 to 0.75

T	0.99	T	0.9
F	0.01	F	0.3
F	0.01	F	0.1
T	0.99	F	0.6
T	0.99	T	0.8

Start at the threshold at 1 and go up. For each threshold calculates the sensitivity, specificity and precision.

RECEIVER OPERATING CHARACTERISTIC

Link: https://en.wikipedia.org/wiki/Receiver_operating_characteristic.

UNDERSTANDING AUC-ROC CURVE

Link: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>.

When we need to check or visualize the performance of the multi-class classification problem, we use the **AUC-ROC** (Area Under The Curve-Receiver Operating Characteristics) OR AUROC (Area Under the Receiver Operating Characteristics) curve.

- Checks any classification model's performance.

The following content answers the following questions:

- What is the AUC-ROC Curve?
- Defining terms used in AUC and ROC Curve.
- How to speculate the performance of the model?
- Relation between Sensitivity, Specificity, FPR, and Threshold.
- How to use AUC-ROC Curve for the multiclass model?

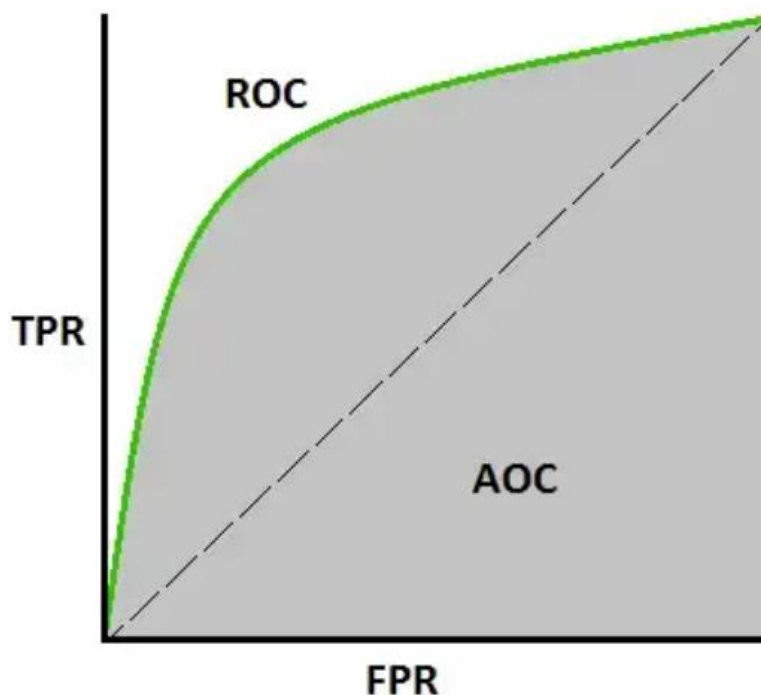
WHAT IS THE AUC – ROC CURVE?

AUC – ROC curve is a performance measurement for the classification problems at various threshold settings. **ROC** is a probability curve and **AUC** represents the degree or measure of separability.

- It tells how much the model is capable of distinguishing between classes.
- Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1.
 - By analogy, the higher the AUC, the better the model is at distinguishing between the patients with the disease and no disease.

ROC curve evaluates the performance of our model across all possible classification thresholds and look at the true positive and false positive rates at that threshold. For instance, if you were to pick a random positive example out of the distribution, and you pick a random negative example, what is the probability that the model will correctly assign a higher score to the positive than it does the negative? In a sense, what's the probability it gets that little pairwise order incorrect? → turns out that probability is exactly equal to the probability value of the area under the ROC curve. For example, if you see a value of 0.9 area under ROC, that's the probability that you will get that pairwise comparison correct.

The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.



DEFINING TERMS USED IN AUC & ROC CURVE

TPR (True Positive Rate)/Recall/Sensitivity

$$\text{TPR / Recall / Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Specificity

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

FPR (False Positive Rate)

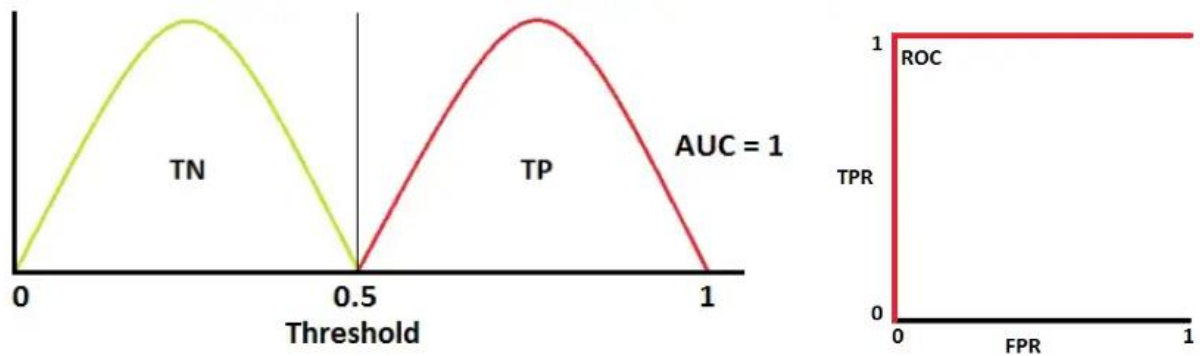
$$\begin{aligned}\text{FPR} &= 1 - \text{Specificity} \\ &= \frac{\text{FP}}{\text{TN} + \text{FP}}\end{aligned}$$

HOW TO SPECULATE ABOUT THE PERFORMANCE OF THE MODEL?

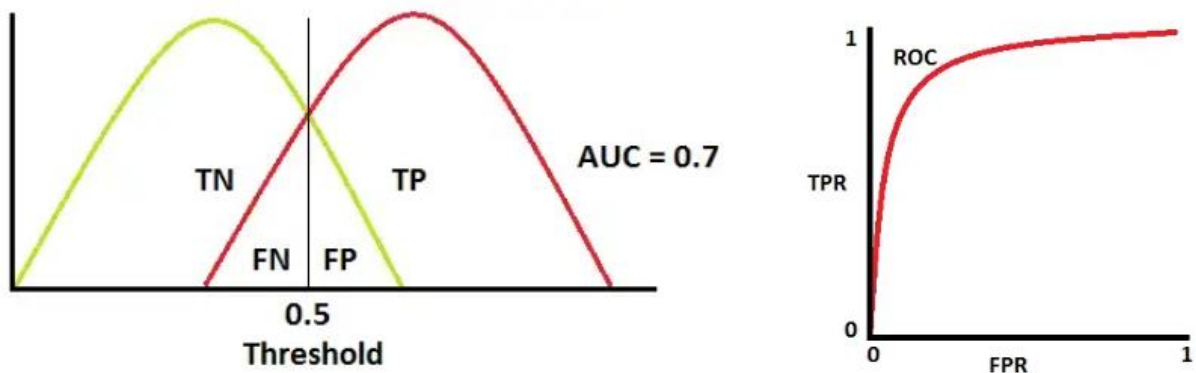
- An **excellent model** has AUC near to the 1 which means it has a good measure of separability.
- A **poor model** has an AUC near 0 which means it has the worst measure of separability.
 - In fact, it means it is **reciprocating the result**. It is predicting 0s as 1s and 1s as 0s.
- And when AUC is 0.5, it means the model has **no class separation capacity** whatsoever.

Let's interpret the above statements - As we know, ROC is a curve of probability. So, let's plot the distributions of those probabilities:

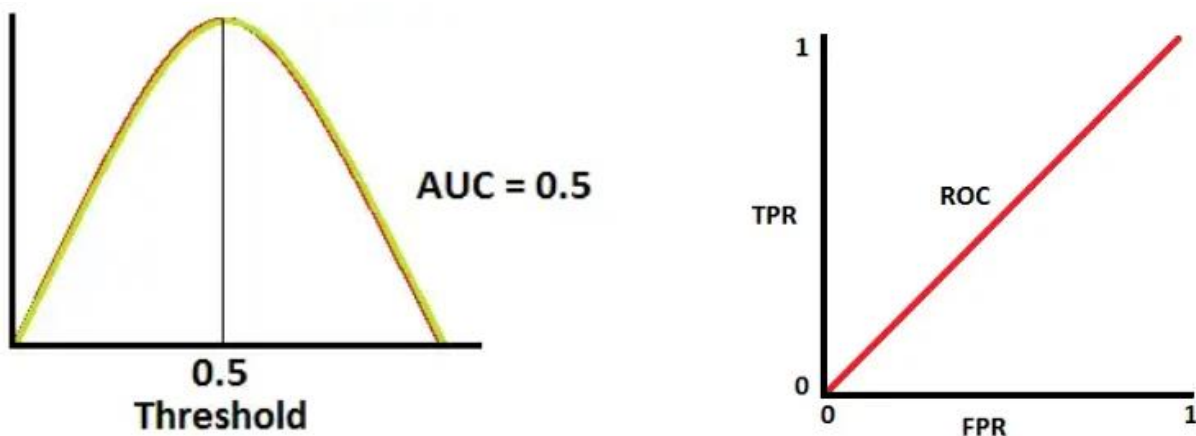
- Red distribution curve is of the positive class (patients with disease) and the green distribution curve is of the negative class (patients with no disease).
 - **Ideal situation** = when 2 curves don't overlap at all means model has an ideal measure of separability. It is perfectly able to distinguish between positive class & negative class.



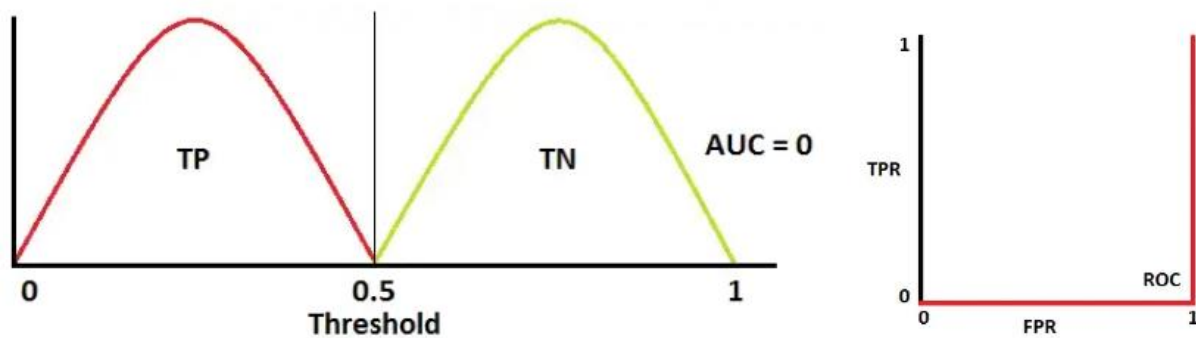
- When 2 distributions overlap, we introduce **type 1 and type 2 errors**. Depending upon the threshold, we can minimize or maximize them.
 - When AUC is 0.7, it means there is a 70% chance that the model will be able to distinguish between positive class and negative class.



- This is the **worst situation**. When AUC is approximately 0.5, the model has no discrimination capacity to distinguish between positive class and negative class.



- When AUC is approximately 0, the model is actually **reciprocating the classes**. It means the model is predicating a negative class as a positive class and vice versa.



THE RELATION BETWEEN SENSITIVITY, SPECIFICITY, FPR, & THRESHOLD

Sensitivity and Specificity are **inversely proportional** to each other.

- So, when we **increase Sensitivity**, **Specificity decreases**, and vice versa.

Sensitivity **↑**, Specificity **↓**
 Sensitivity **↓**, Specificity **↑**

When we **decrease the threshold**, we get **more positive values** thus it **increases the sensitivity** and **decreasing the specificity**.

When we **increase the threshold**, we get **more negative values** thus we get **higher specificity** and **lower sensitivity**.

As we know FPR is 1 – specificity. So, when we **increase TPR**, **FPR also increases** and vice versa.

TPR **↑**, FPR **↑** and TPR **↓**, FPR **↓**

HOW TO USE THE AUC ROC CURVE FOR THE MULTI-CLASS MODEL?

In a **multi-class model**, we can plot the N number of AUC ROC Curves for N number classes using the **One vs All methodology**.

- For example, if you have 3 classes named X, Y, AND Z, you will have
 - one ROC for X classified against Y and Z,
 - another ROC for Y classified against X and Z, and
 - the third one of Z classified against Y and X.

ROC & AUC, CLEARLY EXPLAINED

Link: <https://www.youtube.com/watch?v=4jRBRDbJemM>.

CLASSIFICATION

Link: <https://developers.google.com/machine-learning/crash-course/classification/video-lecture>.

ROC VS PR

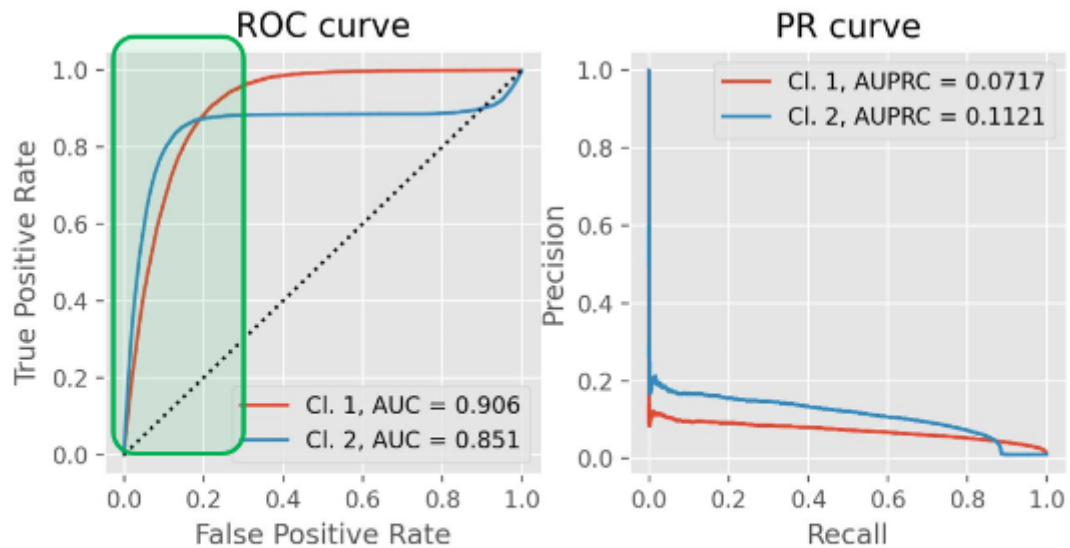
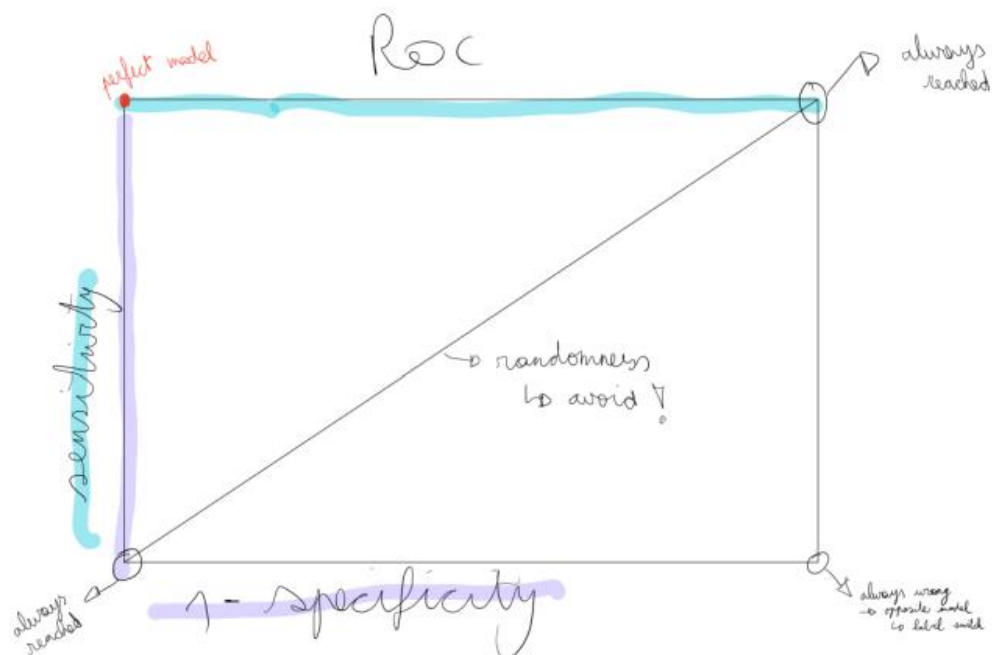


Figure: Image by Fabio Sigrist;

<https://towardsdatascience.com/demystifying-roc-and-precision-recall-curves-d30f3fad2cbf>

True positive rate = sensitivity = recall.

False positive rate = 1 – specificity.



CROSS VALIDATION (CV)

- Multiple train/test split (multiple model(?))
- Optimal use of data
- K-fold (Stratified), Leave-one-out (LOO)

MACHINE LEARNING FUNDAMENTALS – CROSS VALIDATION

Link: <https://www.youtube.com/watch?v=fSyztzGwwBVw>.

Cross validation allows us to compare different machine learning methods and get a sense of how well they will work in practice.

1. We want to use the variables (Chest Pain, Good Blood Circulation, etc..) to predict if someone has heart disease.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
...

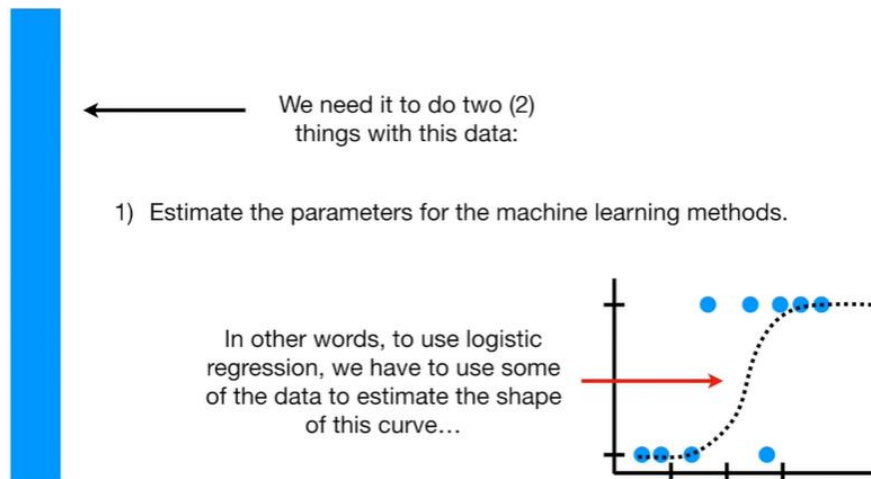
2. Then, when a new patient shows up, we can measure these variables and predict if they have heart disease or not.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	

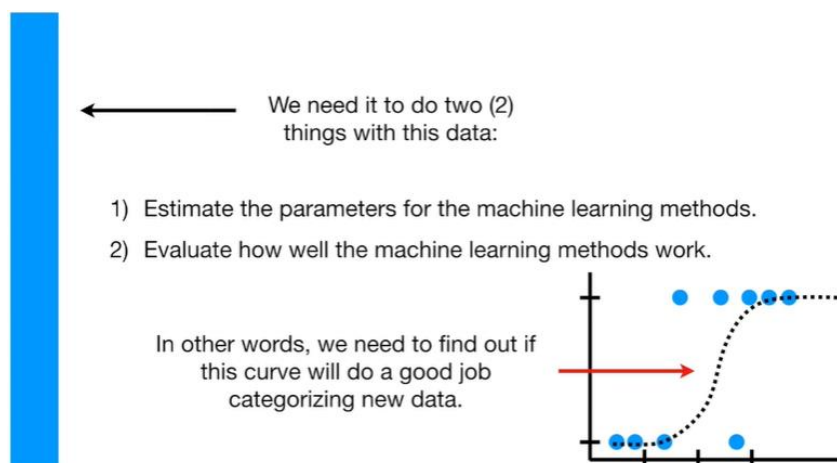
3. However, first we must decide which machine learning method would be best. We could use Logistic Regression, K-nearest neighbours or support vector machines (SVM) and many more machine learning methods. How do we decide which one to use?
4. Cross validation allows us to compare different machine learning methods and get a sense of how well they will work in practice.
5. Using machine learning lingo, we need the following explained data to



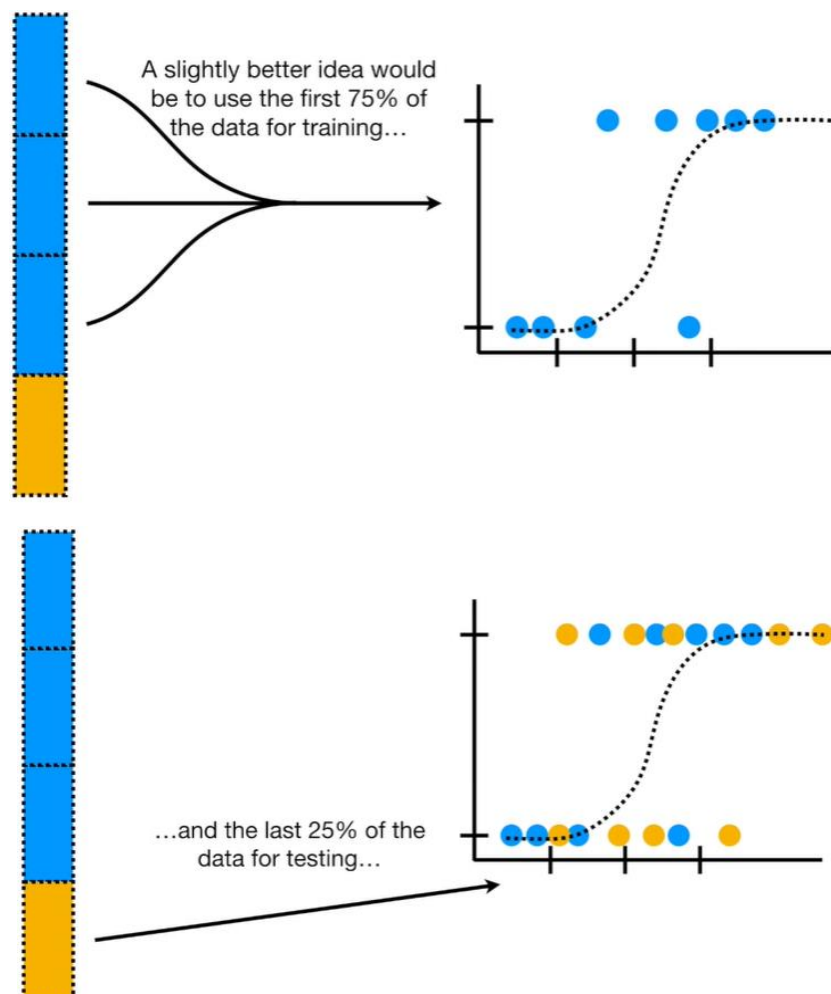
- a. **Train** the machine learning methods. In machine learning lingo, **estimating parameters** is called "**training the algorithm**."



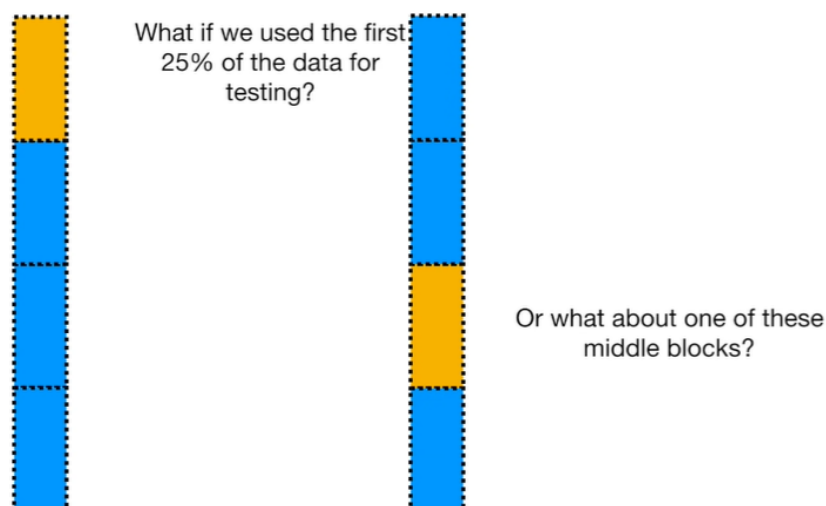
- b. **Test** the machine learning methods. In machine learning lingo, **evaluating a method** is called "**testing the algorithm**."



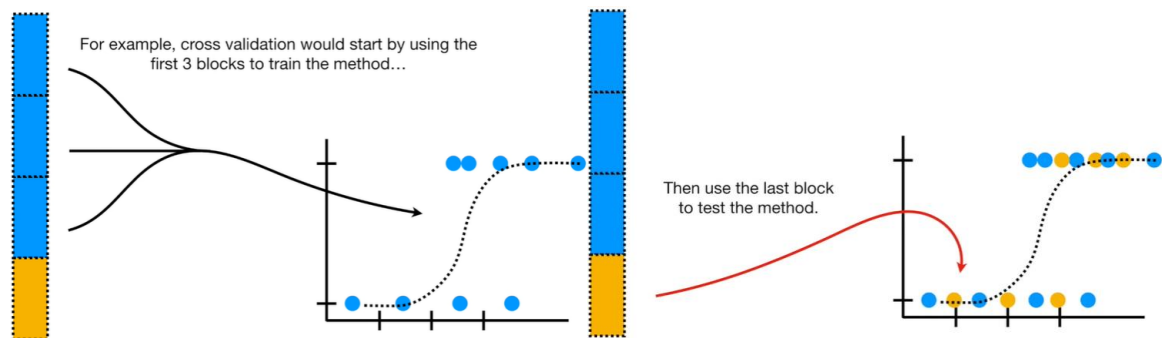
6. Reusing the same data for both training and testing is a bad idea because we need to know how the method will work on data it wasn't trained on.
7. A slightly better idea would be to use the first 75% of the data for training and the last 25% of the data for testing.



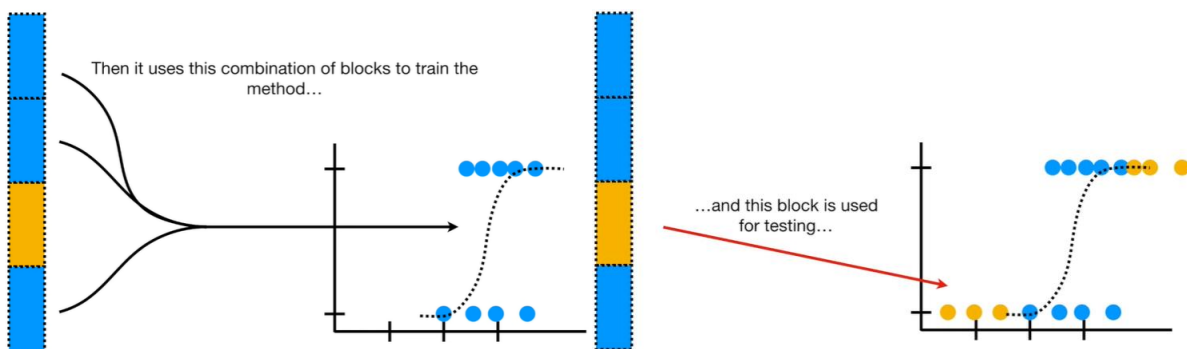
8. We could then compare method by seeing how well each one categorized the test data.
9. But how do we know that using the first 75% of the data for training and the last 25% of the data for testing is the best way to divide up the data?



10. Rather than worry too much about which block would be best for testing, cross validation uses them all, one at a time, and summarizes the results at the end.



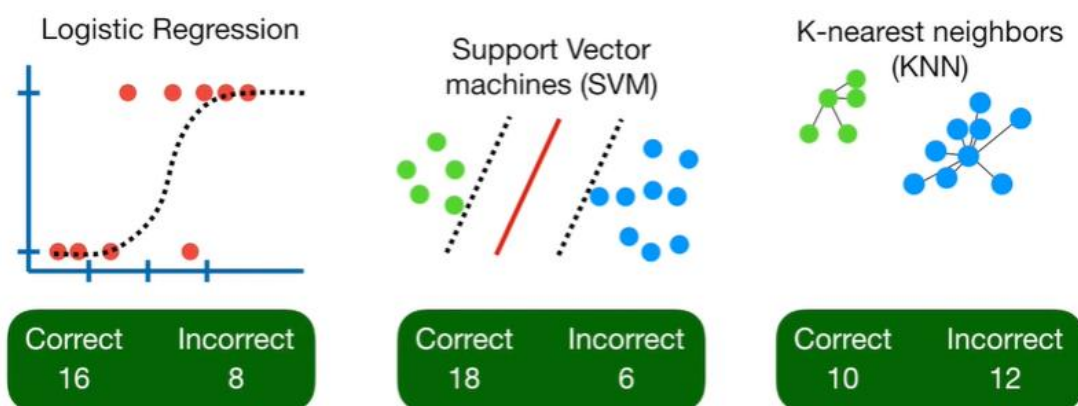
And then it keeps track of how well the method did with the test data.



And then it keeps track of how the method did with the test data.

Etc.

11. In the end, every block of data is used for testing, and we can compare methods by seeing how well they performed. In this case, since the support vector machine did the best job classifying the test datasets, we'll use it!



12. **NOTE:** In this example, we divided the data into 4 blocks. This is called **Four-Fold Cross Validation**. However, the number of blocks is arbitrary.
- In an extreme case, we could call each individual patient (or sample) a block = "**Leave One Out Cross Validation**." Each sample is tested individually.

- b. That said, in practice, it is very common to divide the data into 10 blocks = **Ten-Fold Cross Validation**.
13. We want to use a method that involved a “tuning parameter” – a parameter that isn’t estimated, but just sort of guessed. For example, Ridge Regression has a tuning parameter. Then we could use 10-fold cross validation to help find the best value for that tuning parameter.

CROSS VALIDATION EXPLAINED – EVALUATING ESTIMATOR PERFORMANCE

Link: <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85>.

The ultimate goal of a Data Scientist is to develop a Model in order to get Predictions of New Data or Forecast some events for future on Unseen data.

- A **good model** is not the one that gives accurate predictions on the known data or training data but the one which gives good predictions on the new data and avoids overfitting and underfitting.

You will know:

- Why to use **cross validation** is a procedure used to estimate the skill of the model on new data.
- There are common tactics that you can use to select the value of k for your dataset.
- There are commonly used variations on cross-validation such as stratified and LOOCV that are available in scikit-learn.
- Practical implementation of K-Fold Cross Validation in Python.

To derive a solution, we should first understand the problem. Before we proceed to understand cross validation, we should first understand overfitting and underfitting.

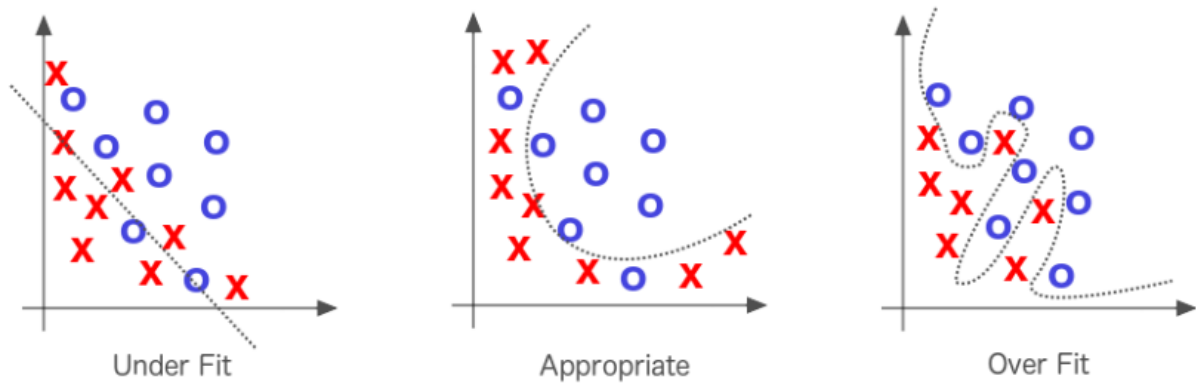
UNDERSTANDING UNDERFITTING & OVERFITTING

Overfit model = overfitting occurs when a statistical model or machine learning algorithm captures the noise of the data. Intuitively, overfitting occurs when the model or the algorithm fits the data too well.

- Overfitting a model result in good accuracy for training data set but poor results on new data sets.
 - Such a model is not of any use in the real world as it is not able to predict outcomes for new cases.

Underfit model = underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough.

- Underfitting is often a result of an excessively simple model. Simple = means that the missing data is not handled properly, no outlier treatment, removing of irrelevant features or features which do not contribute much to the predictor variable.



HOW TO TACKLE PROBLEM OF OVERFITTING

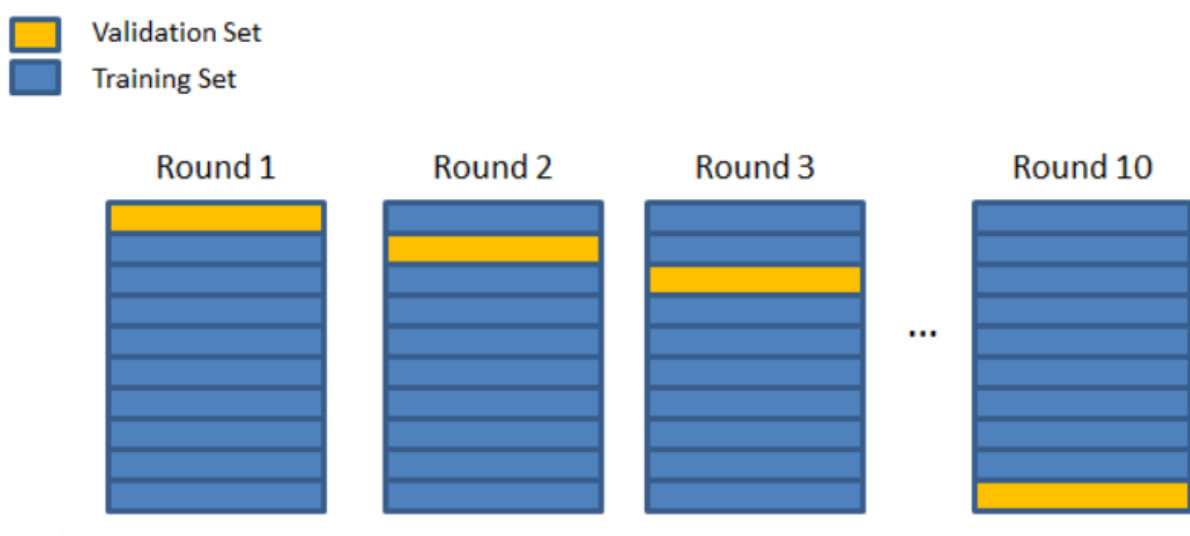
The answer = Cross Validation.

A key challenge with overfitting, and with machine learning in general, is that we can't know how well our model will perform on new data until we actually test it.

- To address this, we split our initial dataset into separate training and test subsets.

There are different types of cross validation techniques, but the overall concept remains the same,

- To partition the data into a number of subsets.
- Hold out a set at a time and train the model on remaining set
- Test model on hold out set
- Repeat the process for each subset of the dataset



the process of cross validation in general

TYPES OF CROSS VALIDATION

- K-Fold cross validation
- Stratified K-fold cross validation
- Leave one out cross validation

K-FOLD CROSS VALIDATION



The procedure, k-fold cross validation, has a single parameter called **k** that refers to the number of groups that a given data sample is to be split into.

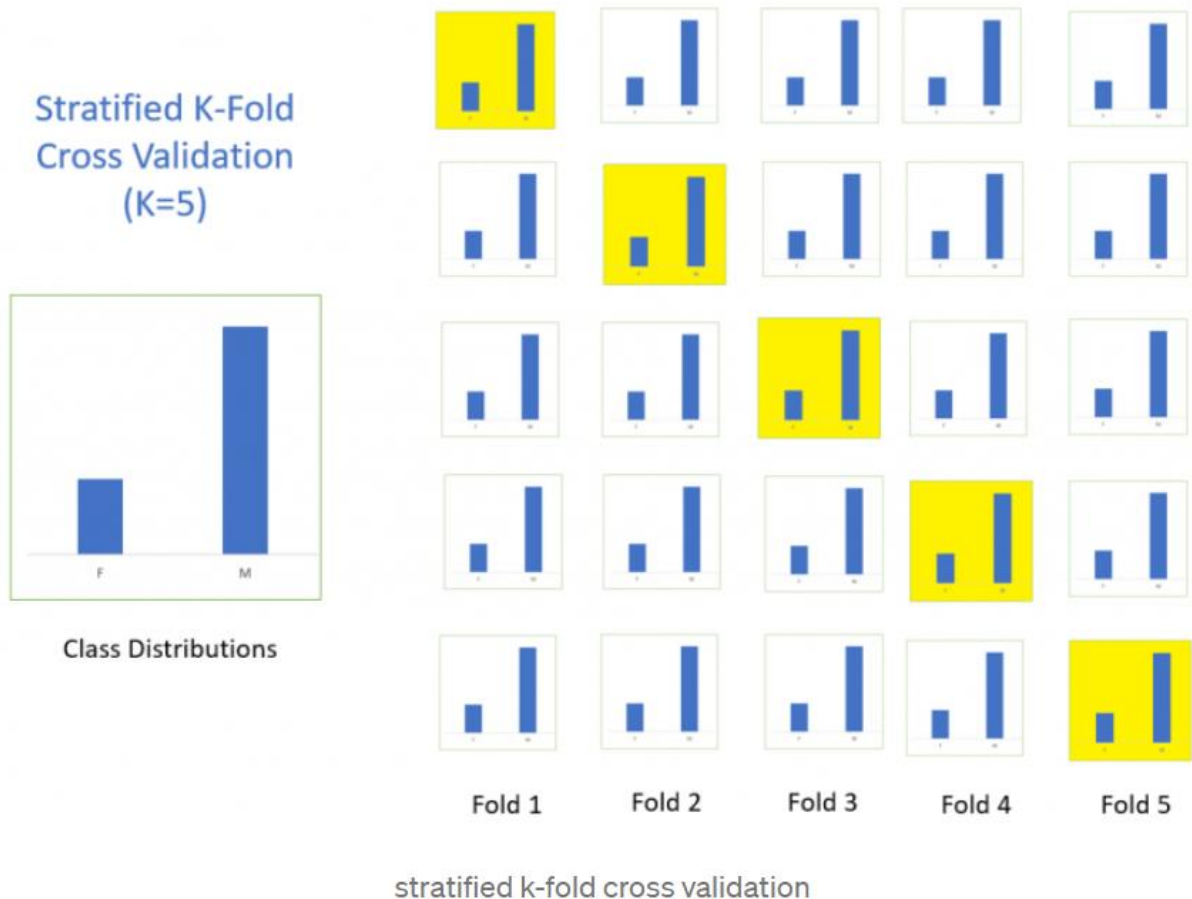
- When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k = 10 becoming 10-fold cross validation.
 - If k = 4, the dataset will be divided into 5 equal parts and the below process will run 5 times, each time with a different holdout set.
 - Take the group as a holdout or test data set.
 - Take the remaining groups as a training data set.
 - Fit a model on the training set and evaluate it on the test set.
 - Retain the evaluation score and discard the model
 - At the end of this process, summarize the skill of the model using the sample of model evaluation scores.
- How to decide the value of k?
 - The value of k is chosen such that each train/test group of data samples is large enough to be statistically representative of the broader dataset.
 - A value of k = 10 is very common in the field of applied machine learning and is recommend if you are struggling to choose a value for your dataset.
 - if a value for k is chosen that does not evenly split the data sample, then one group will contain a remainder of the examples.
 - It is preferable to split the data sample into k groups with the same number of samples, such that the sample of model skill scores are all equivalent. m

STRATIFIED K-FOLD CROSS VALIDATION

- Same as K-Fold cross validation, just a slight difference.

The splitting of data into folds may be governed by criteria such as ensuring that each fold has the same proportion of observations with a given categorical value, such as the class outcome value.

In below image, the stratified k-fold validation is set on basis of Gender whether M or F.



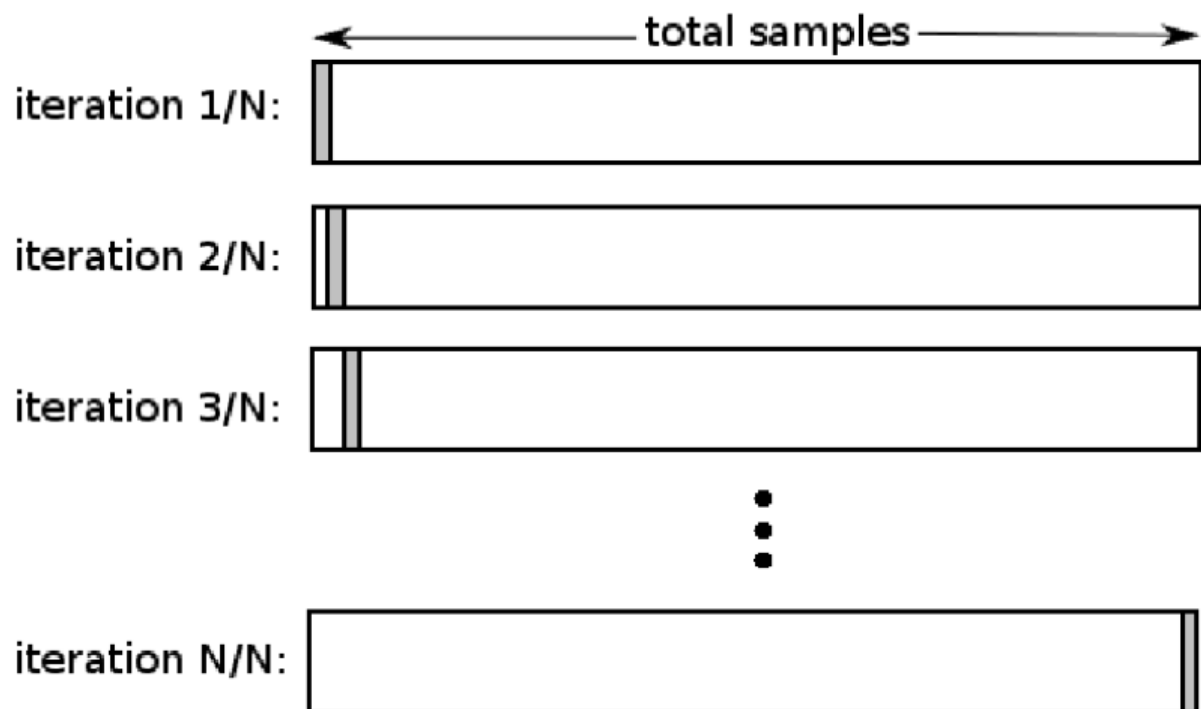
LEAVE ONE OUT CROSS VALIDATION (LOOCV)

This approach leaves 1 data point out of training data, i.e., if there are n data points in the original sample, then $n - 1$ sample are used to train the model and p points are used as the validation set.

- This is repeated for all combinations in which the original sample can be separated this way, and then the error is averaged for all trials, to give overall effectiveness.
 - The number of possible combinations is equal to the number of data points in the original sample or n .

Note:

- Cross validation is a very useful technique for assessing the effectiveness of your model, particular in cases where you need to mitigate over-fitting.



representation of leave one out cross validation

IMPLEMENTATION OF CROSS VALIDATION IN PYTHON

We do not need to call the fit model separately while using cross validation, the "cross_val_score" method fits the data itself while implementing the cross validation on data.

Below is the example for using k-fold cross validation.

```
import pandas as pd
import numpy as np
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm
from sklearn.model_selection import cross_val_score
#read csv file

data = pd.read_csv("D://RAhil//Kaggle//Data//Iris.csv")

#Create Dependent and Independent Datasets based on our Dependent #and
Independent features

X = data[['SepalLengthCm','SepalWidthCm','PetalLengthCm']]
y= data['Species']

model = svm.SVC()

accuracy = cross_val_score(model, X, y, scoring='accuracy', cv = 10)
print(accuracy)

#get the mean of each fold
print("Accuracy of Model with Cross Validation is:",accuracy.mean() *
100)
```

Output:

```
[0.93333333 0.93333333 1.         1.         0.93333333 0.86666667
 0.93333333 0.93333333 1.         1.         ]
Accuracy of Model with Cross Validation is: 95.33333333333334
```

Note:

- The accuracy of the model is the average of the accuracy of each fold.

Specifically, you learned:

- That cross validation is a procedure used to avoid overfitting and estimate the skill of the model on new data.
- There are common tactics that you can use to select the value of k for your dataset.
- There are commonly used variations on cross validation, such as stratified and repeated, that are available in scikit-learn.

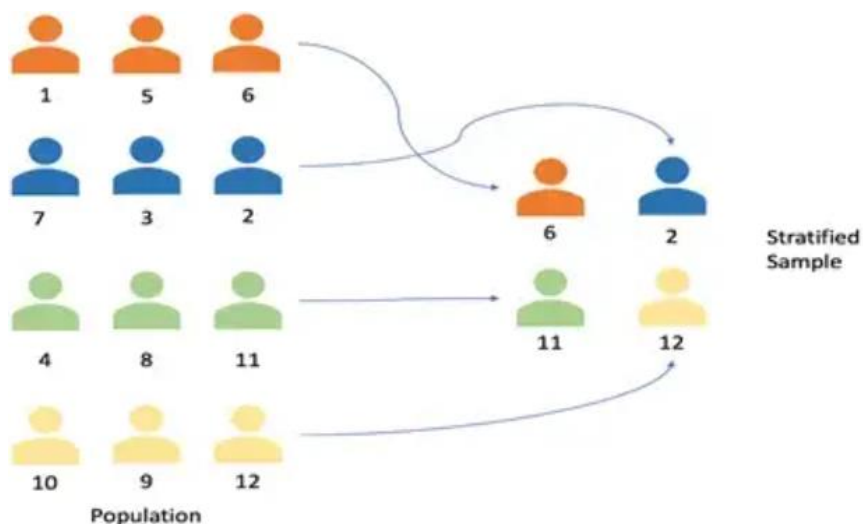
WHAT IS MEANT BY ‘STRATIFIED SPLIT’?

Link: <https://medium.com/@analyttica/what-is-meant-by-stratified-split-289a8a986a90>.

Stratified Split (Py) helps us split our data into 2 samples (i.e., Train Data & Test Data), with an additional feature of specifying a column for stratification.

Example: We mention the variable Age as the column for stratification then the division is done in such a way that each unique age value comes in both the Data Sets (Train/Test)

- In general, it is a way to make subsets from several strata of the main data.



We use the Train Data for Model Building and the Test Data for Model Testing/Validation.

- This function creates 2 new tables where the data is bifurcated on the basis of the parameters specified by the user.

Application:

- It is used to split our data into 2 sets (Train Data & Test Data).
- Train Data should contain 60-80% of total data points.
- Test Data should contain 20-30% of total data points.

	CustomerId	Age	Item_Outlet_Sales	Current_Balance	Vis_rate	postdelinquency	DefaultRate
1	15565251	42	94	9555	5.5555	75	0
2	15565555	42	104	5455	5.5555	55	0
3	15567325	42	95	9255	5.5555	0	0
4	15570051	43	92	5755	5.55553	0	0
5	15570554	43	94	9575	5.5555	543	0
6	15570559	50	95	9545	5.5555	0	0
7	15572547	50	95	9545	5.5555	49	0
8	15574572	43	95	5754	5.5555	0	0
9	15575745	42	104	5755	5.5555	0	0

Example: Consider the above Data Set. It has 10 samples so when we use the Stratified Split (Py) function we need to specify 2 parameters.

- Train size = suppose we mention 0.7 i.e., 70%
- Stratification variable

So the function executes and creates 2 new tables containing 70% & 30% data respectively.

Input:

- In ATH, to run Stratified Split (Py) select the columns of the data and then use the path: Data Management → Data Sampling/Subsetting → Stratified Split (Py).
 - In the 'Train size': Enter a suitable Value between 0 – 1.
 - In the 'Stratification Variable': Select The Variable for stratification.

Output and Interpretation:

1. Train Data 70% of actual data points –

1	15565251	42	94	9555	5.5555	75	0
2	15565555	42	104	5455	5.5555	55	0
3	15567325	42	95	9255	5.5555	0	0
4	15570051	43	92	5755	5.55553	0	0
5	15570554	43	94	9575	5.5555	543	0
6	15570559	50	95	9545	5.5555	0	0

2. Test Data 30% of actual data points –

7	15572547	50	95	9545	5.5555	49	0
8	15574572	43	95	5754	5.5555	0	0
9	15575745	42	104	5755	5.5555	0	0

ASSIGNMENT

- Build + plot PR and ROC for train and test (Flowers/Cars)
- Build cars model for 50/50 split => plot ROC
- CV for Cars => ROC

06 RANDOM FOREST & FEATURE SELECTION

RANDOM FOREST

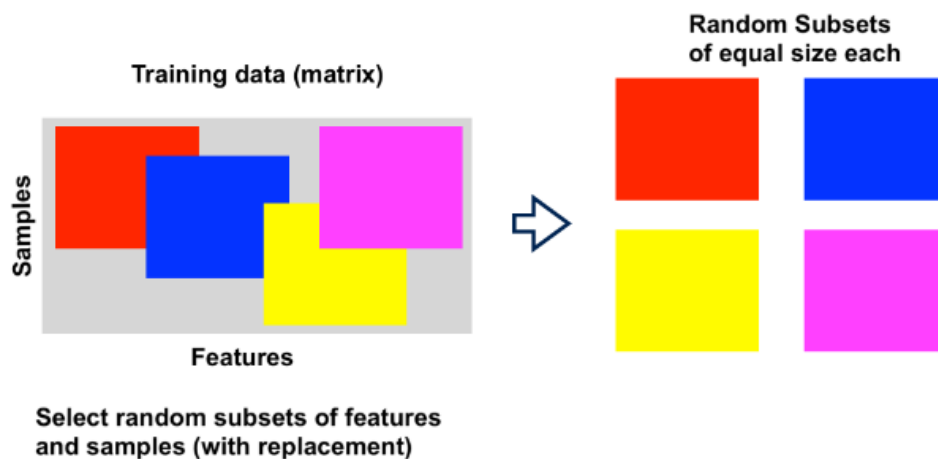
- How does a decision tree work?
- How does a random forest work? (Building process, decision process, output).
- What are the main parameters and what do they do?

3 crucial concepts:

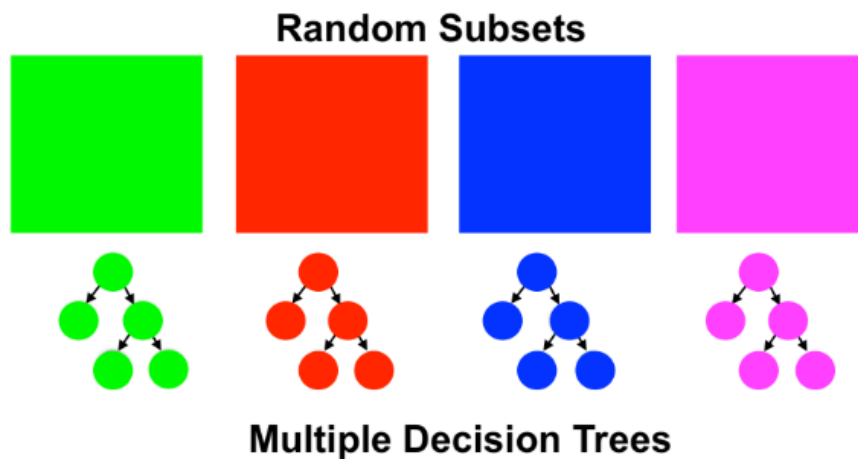
- Random subsets of data
- Multiple trees
- Voting to get result

RANDOM FOREST STEPS

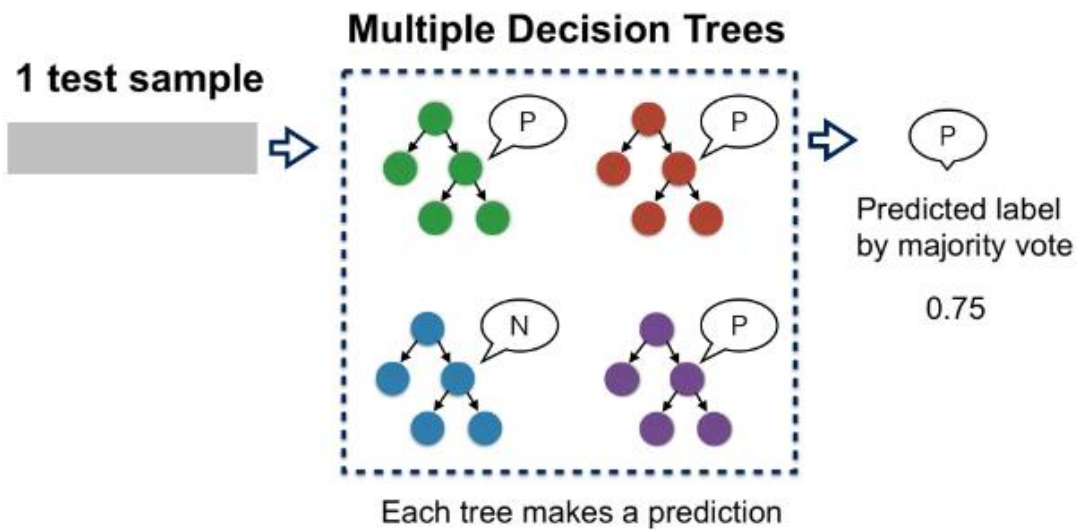
1. Step 1



2. Step 2



3. Step 3

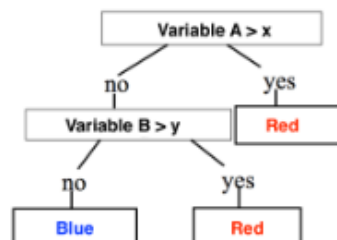
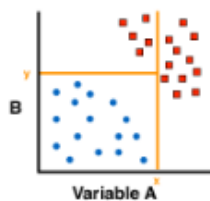
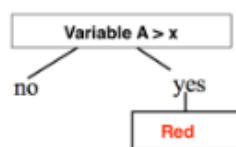
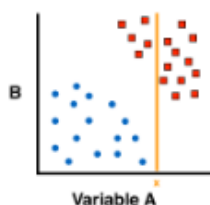


Sources:

Link: <https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5>.

Link: <https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147>.

DECISION TREE



1. Select the feature with the best separation between classes (how to quantify 'best'?)
2. Use this feature to make the next nodes (leaves) in the tree
3. Repeat process in each node until all samples have been classified (only one class at each endpoint)

CALCULATE SEPARATION QUALITY

Quantify the **Gini impurity** in each node:

$$i = 1 - p_p^2 - p_N^2$$

- p_p is the proportion (fraction) of positives and vice versa for the negatives.

Use a feature to determine a split in the node and calculate the decrease in Gini impurity δi :

$$\delta i = i_{parent} - i_{child1} * f_{child1} - i_{child2} * f_{child2}$$

- f is the fraction of parent samples in that child.

Gini impurity

- Lower Gini impurity signifies better separation between classes.
- If only a single class occurs and the other is absent (perfect separation), the Gini impurity will be 0 ($= 1 - 1 - 0$).
- δi will be 0 if there is no improvement in the separation.

DECISION TREE EXAMPLE

Heart	Smoking	Gender
TRUE	TRUE	M
FALSE	FALSE	F
TRUE	FALSE	M
TRUE	TRUE	F
FALSE	FALSE	M
FALSE	TRUE	F
TRUE	TRUE	M
FALSE	FALSE	F
TRUE	TRUE	M
TRUE	TRUE	F

1. Make tree to classify heart disease
2. First node in tree: smoking or gender?

DECISION TREE (ADVANTAGES

- Intuitive approach, easy to implement and understand
- Easy to check why a given sample is classified a certain way (so not a black box)

DECISION TREE (DIS)ADVANTAGES

- Approach is 'greedy' and will continue until all samples are correctly classified (long and overfit trees).
- Tend to be very dependent on the input data and small changes can lead to very different trees.

CLASS NOTES

- The first one above x

- Calculating impurities
- You get points if you build nodes that are pure, but it also gets bigger
- Reward for a decreasing in impurity
- The model doesn't apply relationships
- Statistical feature selection doesn't work in machine learning because its 1 dimensional
- There is no problem with throwing features away after

FEATURE SELECTION

- Manual feature selection
- Statistical feature selection
- Model based feature selection
- Boruta feature selection

BORUTA FEATURE SELECTION

Link: <https://towardsdatascience.com/boruta-explained-the-way-i-wish-someone-explained-it-to-me-4489d70e154a>.

08 DATA LEAKAGE

Data leakage happens when there is information leaking into the model which should not be there. The result ranges from overly optimistic performance results...

There are 2 main types of data leakage:

- Target leakage
- Train/test contamination

TARGET LEAKAGE

Target Leakage is the type of data leakage with the most obvious effect on performance. It can really destroy a model's performance. This also makes it **easier to identify**.

Problem	AUC	Real performance
Target (outcome) variable included in training phase.	Perfect	Error
Variable only known after event occurrence are included (chronology)	Perfect/Good	No error but low performance
Time series values after event are included (windmill failure detection)	Problem specific	Model too slow (late prediction)

Question: What happens when there is a variable which is highly correlated to the outcome variable, but which is known at the time of prediction (so no chronology issues). Is there data leakage. Do you include it yes/no?

TRAIN/TEST CONTAMINATION

This type is more subtle, it causes smaller problems, but these are therefore harder to detect; consequently, these occur very often in the real-world. It happens most often in the pre-processing phase.

Problem	AUC	Real performance
Bad train/test split (or fold selection). There is test set information in the training data (patients/windmills).	Optimistic	Lower
Feature selection before train/test split	Optimistic	Lower
Imputation/scaling before train/test split	Optimistic	(slightly) lower

HOW TO DETECT

Sometimes these are signs that you had a data/information leak in your pipeline. Below are some warnings and red flags:

6. Model gives error in production (missing a feature)
7. (near) perfect performance for test set (>95% AUC be very cautious).
8. Highly/perfectly correlated features (check if chronology is ok)
9. (small) difference in performance between test/validation and real-world implementation.

VALIDATION SET

Often in data science courses/trainings they recommend using a train/test/validation setup. Whereby you take a validation dataset at the very beginning which you only use at the very end before going into production.

This obviously has some drawbacks:

- You lose part of your data.
- This might not detect certain data leaks:
 - Chronology problems with variables.
 - Chronology problems with time series.
 - Bad split e.g., with patients or windmills being in this set and in the training set.

SOURCES & MATERIALS

DATA LEAKAGE & ITS EFFECT ON THE PERFORMANCE OF A ML MODEL

Link: <https://www.analyticsvidhya.com/blog/2021/07/data-leakage-and-its-effect-on-the-performance-of-an-ml-model/>.

DATA LEAKAGE IN MACHINE LEARNING

Link: <https://machinelearningmastery.com/data-leakage-machine-learning>.

CLASS NOTES

- Sensitivity/recall = Every case you detect, it goes up by 10%
- Precision
 - If its down = huge class imbalance

09 LESSON

INTRODUCTION

Time-Series Forecasting is widely used in business contexts for predictions purposes.

- Sales
- Customers
- Stock
- Server load
- Anomaly detection
- Etc

DIFFERENCE WITH ML

ML:

- $N \times M$ matrix format data + outcome vector y (length N).
- We use M variables to predict the outcome vector y for unknown rows N^* .

Time-series forecasting:

- We have a time series $y(t)$, which stops at time $t = 0$.
- We want to predict $y(t > 0)$ and for this we use $y(t \leq 0)$.
- i.e., the variable we want to predict is also the one we use to fit (train in ML).

THE PROBLEM

Time-Series Forecasting is difficult. And people with good forecasting skills are hard to find.

The reason is mostly down to the algorithms and theory behind it being rather difficult to automate, tune or even set up.

The result is that in many companies the need for forecasts greatly outweighs the rate at which they can be produced.

FACEBOOK PROPHET

The Prophet algorithm was developed by Facebook (<https://facebook.github.io/prophet/>) so that their data scientist could do many of the forecasts themselves, without needing to go to a forecasting expert every time.

The aim of the algorithm is not to solve each forecasting problem, rather it can be used for specific problem but with great flexibility. Some problems for which it can work (spot the common theme):

- predicting airline passengers over time.
- Predicting crowds over time.
- Predicting sales over time.
- Predicting server load over time.

The common theme is really people interacting with a certain service. So, Facebook Prophet is really only suited for predictions in a business context.

It works well for data with strong cycles or seasonalities (daily/weekly/yearly), with certain outlier events known in advance (holiday events, new product launches), and natural growth limits (e.g., product market saturation).

Note that forecasting experts will generally produce better forecasting results with the classical tools, but this takes time.

HOW DOES IT WORK

The time-series forecasting problem whereby time is explicitly modelled to a simply a fitting problem. The following function is fit to the data $y(t)$:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

With

- g the trend component,
- s the periodic changes (seasonality),
- h the holiday effects and
- ϵ the error component.

Important to note that this model is additive, i.e., the components are simply added together and can thus be interpreted independently. But also, more can be added if needed (extra seasonality) but more on that later.

THE TREND COMPONENT

The trend component is fitted automatically. There are 2 trend types available (problem specific):

- Piecewise linear (changepoints selected manually or automatic).
- Saturating growth model (product market saturation).

SEASONALITY

There are different seasonalities available. This is useful because patterns can occur at multiple scales. For example:

- Yearly seasonality are patterns which occur on a yearly repeating scale but not necessarily on the same date e.g., holiday periods like easter or the start of the new school year.
- Weekly seasonalities capture patterns which occur every week scale e.g., weekday-weekend differences.
- (Optionally) daily seasonalities captures pattern occurring each day. This becomes tricky as this is on top of the weekly patterns and should therefore be somewhat consistent for each consecutive day.

These seasonalities are fitted with a Fourier Series (https://en.wikipedia.org/wiki/Fourier_series).

HOLIDAYS

Holidays are things which can have a tremendous effect on business time series. Why?

Holidays and events sometimes do not occur every year at the same time (i.e., every 76th day of the year, but are determined by lunar/religious/etc rules. The assumption Prophet makes is that the effect of the event is usually the same year over year, but the date is not.

SOURCES

- Paper: Forecasting at Scale by SJ Taylor and B Letham
- Link: <https://facebook.github.io/prophet/>.
- Link: <https://www.youtube.com/watch?v=pOYAXv15r3A>.
- By next lesson:
 - Watch the video.
 - Quiz on FB Prophet (begin of next lesson)
 - Find a time series dataset you want to predict and set it up in a notebook (load it in and visualise the signal).

10 TIME-SERIES FORECASTING 2

MULTIPLICATIVE/ADDITIVE

Something we skipped last lesson: difference between additive and multiplicative model.

Additive:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Multiplicative:

$$\hat{y} = g(t) + g(t) \cdot s(t) + g(t) \cdot h(t) + \epsilon_t$$

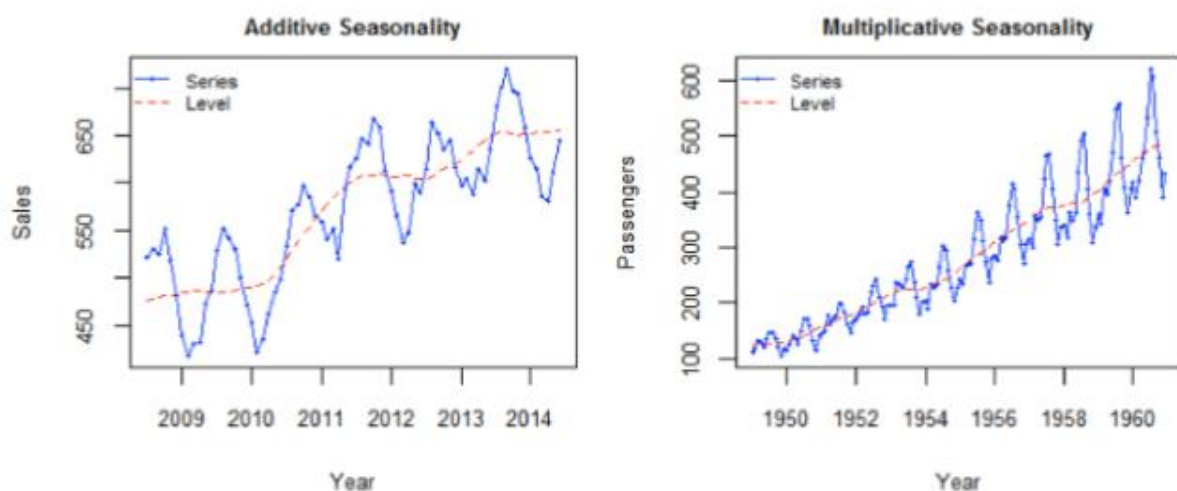
Or:

$$\hat{y} = g(t) \cdot (1 + s(t) + h(t)) + \epsilon_t$$

Again with

- g the trend component,
- s the periodic changes (seasonality),
- h the holiday effects and
- ϵ the error component.

Multiplicative or Additive – spot the difference.



PERFORMANCE EVALUATION

How good is my forecast?

Different data to be used (think of ML):

- Metrics calculated on the training data performance (goodness of fit).
- Metrics calculated on a test set (end of data, remember data leakage).

PERFORMANCE EVALUATION – TRAINING DATA

By checking the performance on the training data we can evaluate a number of things:

- Get a feel for the model: how do the model components look like, do these make sense or are they overfitting?
- Are we missing things: by looking at the residuals we can check for missed (periodic) patterns.

PERFORMANCE EVALUATION – TEST DATA

The best quantification whether the model is actually able to predict things is by looking at the predictions on the test data.

- Root Mean Square Error/Deviation:

$$RMSE/RMSD = \sqrt{\frac{\sum_i^N (\hat{y}_i - y_i)^2}{N}}$$

- Mean Absolute Error:

$$MAE = \frac{\sum_i^N |\hat{y}_i - y_i|}{N}$$

- Mean Absolute Percentage Error/Deviation:

$$MAPE/MAPD = \frac{1}{N} \sum_i^N \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

Besides the default performance metrics there are also more visual variant:

- Plot of data vs predictions
- Distribution plot of residuals
- Correlation plot
- Rolling correlation plot (N-day or monthly): time series of the correlation over time.

DIY

Use the Auckland cycle data and do the following.

- Load in the data and select only the cyclists on Tamaki Drive (choose one way for your convenience).
- Select a test set of a full year.
- Train a basic model and evaluate the performance both numerically and visually.
- Add holidays and evaluate the performance.
- Add weather and evaluate the performance.

12 NEURAL NETWORKS

The (possible) future of ML & AI

INTRODUCTION

MACHINE LEARNING

- Pre-process data
- (Build features)
- Build model.
- Evaluate performance.

HOW DO YOU SEE THE WORLD?

link: <https://www.theguardian.com/education/2012/nov/12/improbable-research-seeing-upside-down>.

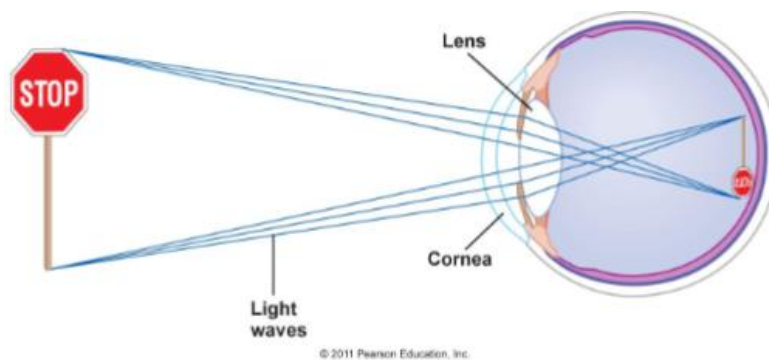


Figure: You already see the world upside down.

The brain

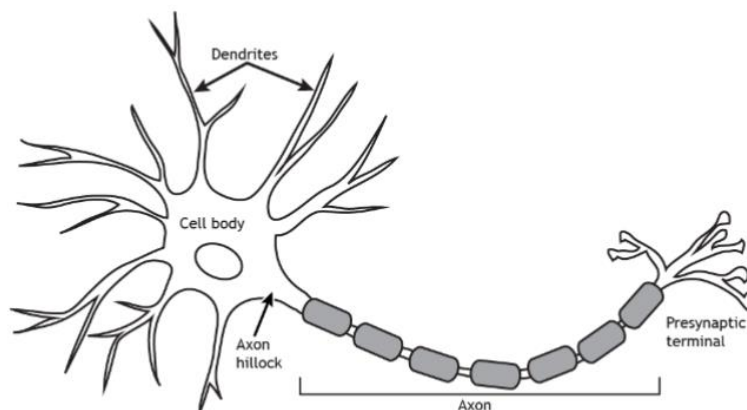
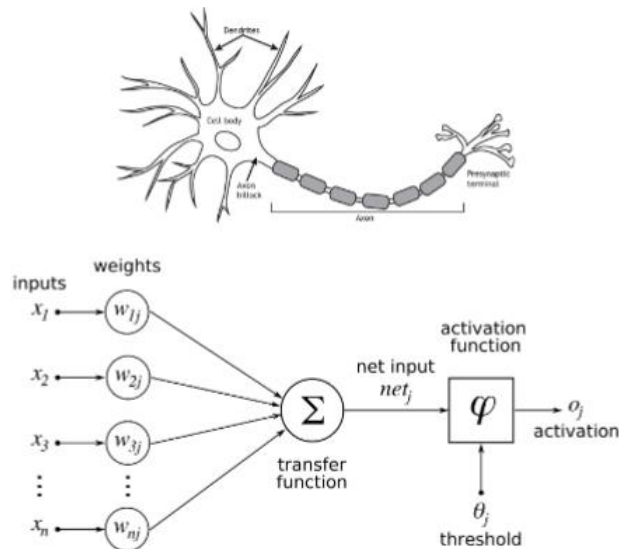


Figure: 'Neuron' by Casey Henley

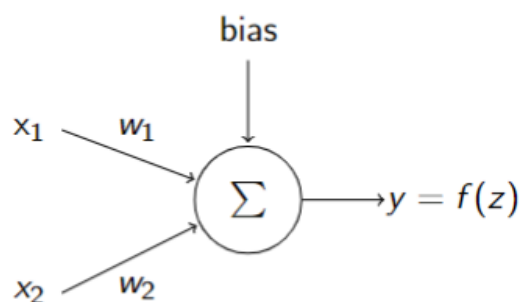
(ARTIFICIAL) NEURONS



- Inputs = features
- Transfer function = $\sum x_i \cdot w_i$.
- Activation: net input \geq threshold.

ARTIFICIAL NEURON IN 2D FEATURE SPACE

Example, data with 2 features x_1 and x_2 :



Activation functions:

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

$$f(z) = ReLU = \max(0, z)$$

$$z = bias + \sum_{i=1}^2 x_i \cdot w_i$$

This boils down to a decision line in 2D feature space!

NEURAL NETWORK: INTRODUCTORY PROBLEM

MNIST dataset



How would you train a model for this?

- Pre-processing?
- Model input/features?

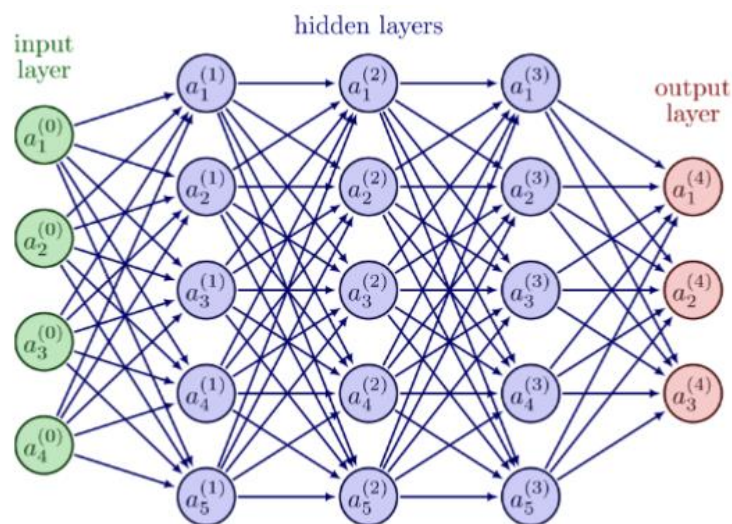


Figure: Note this is a specific type of neural network called a fully connected neural network.

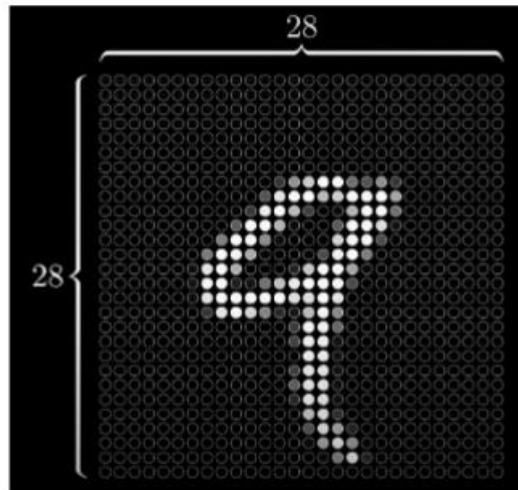


Figure: Screenshot from 3Blue1Brown (see sources).

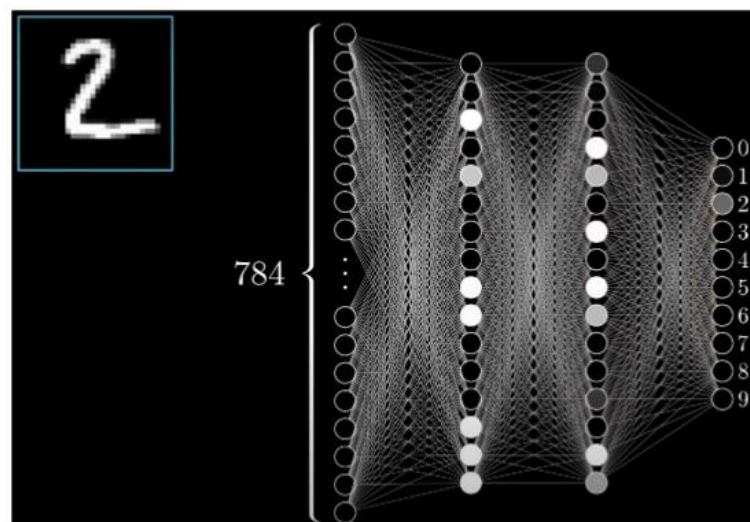


Figure: Screenshot from 3Blue1Brown (see sources).

NEURAL NETWORK – LAYERS & PARAMETERS

MNIST example with shown network:

- 4 layers:
 - 1 input layer with 784 input nodes (grayscale pixels)
 - 2 hidden layers with 16 nodes each (arbitrary number in this case)
 - 1 output layer with as many nodes as there are classes.
- 784 weights for each node in the second layers
- 1 bias per node
- Node activation is determined by:
 - Input nodes
 - Weights
 - Bias
 - Activation function (sigmoid, ReLu, step, etc)

NEURAL NETWORK - TRAINING

1. Give the model a training instance (image, ...)
2. Let the model give the output.
3. Tell the model how bad it is.
4. Give another instance.

HOW TO TELL THE MODEL HOW BAD IT IS

Defining of Loss and Cost function

- The loss function determines the difference between what it should be (the truth) and what the model output for one instance.
- The cost function summarise the loss over the full training set (optionally with a model complexity penalty).

Objective is to minimise the Cost, and thus the Loss. This is done in an optimisation procedure. Often this is Gradient Descent, but many others exist.

Gradient Descent

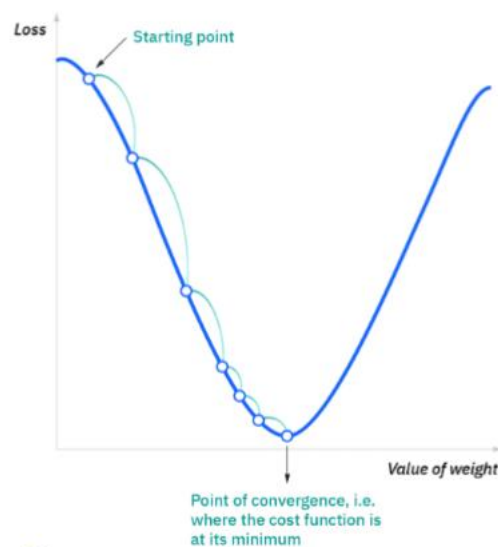


Figure: Source: ibm.com/cloud/learn/gradient-descent

SOFTWARE 2.0

Introduced by Andrej Karpathy: <https://karpathy.medium.com/software-2-0-a64152b37c35>.



Figure: <https://xkcd.com/1838/>

Artificial Neural Networks (and accompanying deep learning) are not just other algorithms which you choose instead of e.g. a Random Forest. This is possible of course, but the possible disruptive nature of this type of learning is much more than that.

- Provide data (labels, augmentation)
- Define learning framework/architecture (layers, size, etc) determined by available performance.
- Train model
- Evaluate
- Deploy

Difference with ML as we have seen.

SOFTWARE 2.0 VS ML

- A lot more data needed:
 - Data augmentation (make more data from the data you have) gathering of labels.
- Model determines what is important, not humans (features are not constructed by people but by computers)
- Humans provide a learning framework (Data, model architecture, compute power).
- (Sometimes) low model explainability.
- More compute power necessary.

EXAMPLES OF THE FUTURE TODAY

- ChatGPT
(https://chat.openai.com/?__cf_chl_tk=t4cY5S4gp6fWb7TWpwF9bMTxe7k224xVgHsPWZP6ALs-1674451428-0-gaNycGzNCJE).
- DALL·E (<https://openai.com/blog/dall-e/>).
- Tesla/Google/others autonomous driving.
- GitHub Copilot
- Google Deepmind research (<https://www.deepmind.com/research>).

SOURCES & MATERIALS

Included in the stuff you need to know for the exam:

BUT WHAT IS A NEURAL NETWORK? CHAPTER 1, DEEP LEARNING

Link: <https://www.youtube.com/watch?v=aircAruvnKk&feature=youtu.be>.

GRADIENT DECENT, HOW NEURAL NETWORKS LEARN | CHAPTER 2, DL

Link: <https://www.youtube.com/watch?v=IHZwWFHwa-w>.

Extra: only if you are interested:

- What is backpropagation really doing? .
- Andrej Karpathy: Tesla AI, Self-Driving, Optimus, Aliens, and AGI:
<https://www.youtube.com/watch?v=cDiD-9MMpb0>.
-

END.