**Decision Trees:**

Slide 2:

Okay, so we’re going to first look at a very simple machine learning model, called decision trees.

Slide 3:

But before that, here is the dataset that we will be training our model on later. Each observation is a patient with general health, checkup, exercise, sex, age category and so on as its features. The target variable is heart disease, i.e. we’re trying to predict whether a patient has heart disease or not. Note that this dataset can also be used to predict whether a patient has skin cancer, another type of cancer, depression, diabetes or arthritis.

Slide 4:

A decision tree is made up of 3 types of nodes, a root node, internal nodes and leaf nodes. The root node is the starting point and represents the entire dataset. Internal nodes are points where observations are split based on their features. Leaf nodes are the end points and represent the final predictions.

Slide 5:

Here is a real life example, where an observation is a day and where wind and wind direction are its features. In this case, we’re trying to predict if a day is suitable for surfing or not. We can see that the 1st level of internal nodes splits the dataset based on an observation’s wind and the 2nd level of internal nodes splits the dataset based on an observation's wind direction. Any questions so far?

**Choosing The Best Feature at Each Node:**

Slide 6:

Okay, so how do we choose the best feature to split on at a node?

Slide 7:

Well, first we need to understand what entropy is. Entropy measures the impurity or uncertainty in a dataset or in a set of observations. Entropy is 0 whenall observations belong to the same class, i.e. when there is max certainty of what class an observation belongs to. Entropy is highestwhen each class contains the same number of observations, i.e. when there is max uncertainty of what class an observation belongs to.

Slide 8:

Here is an example on how to calculate entropy.

Slide 9:

Now that we know what entropy is, we can talk about information gain. Information gain measures the difference in entropy (uncertainty) after splitting the observations based on a feature at a node. The feature with the highest information gain will produce the best split. This is a common metricused to choose which feature to split on at a node. The most common metric used is Gini impurity.

Slide 10:

Here is an example on how to calculate information gain.

Slide 11:

Now that we know what information gain is, we can build a decision tree. Any questions?

**Advantages & Disadvantages of Decision Trees:**

Slide 12:

Okay, so what are the advantages and disadvantages of decision trees?

Slide 13:

Decision trees are easy to interpret, as they’re white box models. This means that their inner workings and decision-making process are transparent and easily understandable, allowing us to see how the model arrives at its predictions.  Unlike neural networks, as they’re black box models. This means that their inner workings and decision-making process are mysterious and difficult to understand. Decision trees also require little to no data preparation, as they can handle discrete, continuous and missing values. Decision trees can also be used for classification and regression. However, today we will only be looking at decision trees for classification.

Slide 14:

Decision trees are prone to overfitting and don’t generalize well to new and unseen data. Decision trees also have a high variance, as small variations within the dataset can produce a very different decision tree. Decision trees can also be more expensive to train compared to other algorithms, as they use a greedy search algorithm for training.

Slide 15:

Okay, so we’ve covered the building block of the model that we’re going to be building later. We’re now going to take a quick break and play a Kahoot.

**[Play Kahoot]**.

Kahoot Link:

<https://create.kahoot.it/share/decision-tree-kahoot/01e96ddb-5901-4039-a18c-3d850de7c984>

**Boosting:**

Slide 16:

Okay, let’s now take a look at boosting.

Slide 17:

Boosting is an ensemble method that can be used to improve the performance of decision trees. Boosting combines multiple individual weak trees to form a strong learner. Each weak learner is trained sequentially to correct the errors made by the previous models.

Slide 18:

Boosting improves accuracy. Boosting is also robust to overfitting. Boosting can also handle imbalanced datasets well,  as it prioritizes misclassified points. Note that an imbalanced dataset is a dataset where the classes are not represented equally. Boosting also has better interpretability, as its sequential nature helps break down decision-making. Any questions?

**Gradient Boosting:**

Slide 19:

Okay, let’s now take a look at a type of boosting called gradient boosting.

Slide 20:

Gradient Boosting trains the base learner, i.e. the 1st weak learner, on the original data X and its labels y. The base learner makes predictions which are then used to compute the residuals r1. Note that the residuals are the difference between the labels and the predictions.

Slide 21:

Gradient Boosting trains the 2nd weak learner on the original data X and the residuals r1. In other words, the weak learner is trained to predict the errors of the previous weak learner. This process continues for all the trees in the ensemble.

Slide 22:

Once all trees are trained, predictions are made by summing the contributions of all the trees. Each tree's predictions are shrunk by multiplying them with the learning rate η. Any questions?

**XGBoost:**

Slide 23:

Okay, so the model that we’re going to be building today is called XGBoost.

Slide 24:

XGBoost is an optimized and scalable implementation of gradient boosting. XGBoost allows for parallel and distributed computing, as trees can be built in parallel. XGBoost also uses a cache-aware prefetching algorithm, which helps reduce the runtime for large datasets. XGBoost also has built-in regularization, this is a technique used to prevent overfitting by penalizing complexity. XGBoost can also handle missing values.

Slide 25:

XGBoost is used in many disciplines, such as healthcare, IT and finance. XGBoost is typically used in healthcare to predict cardiovascular risk factors and cancer patient survival rates. XGBoost is typically used in It for page rankings in search engines. XGBoost is typically used in finance for fraud detection, pricing analysis and more.

Slide 26:

Okay, so we’ve covered the model that we’re going to be building later. We’re now going to take a quick break and play a Kahoot.

**[Play Kahoot]**.

Kahoot Link:

<https://create.kahoot.it/share/xgboost-kahoot/48ec0bc1-0f09-4a0c-aeb9-ec37f99e47b2>

**Recap:**

Slide 27:

So, to recap what we’ve learnt.

Slide 28:

We’ve learnt what how a decision tree works. We’ve learnt how to choose the best feature to split on at a node. We’ve learnt what boosting, gradient boosting and XGBoost is. And we’ve learnt what the possible applications of XGBoost are.

**Workshop Info:**

Slide 29:

Now onto the fun part, let’s build an XGBoost model.

Slide 30:

Here are 2 QR codes. The left QR code will take you to the dataset on Kaggle. The right QR code will take you to the code on GitHub.

Slide 31:

Okay, so we’re going to try and train an XGBoost model to predict if a patient has a heart disease or not. We’re using the cardiovascular disease dataset, that I used earlier as an example, to do this.

Datset Link:

<https://www.kaggle.com/datasets/alphiree/cardiovascular-diseases-risk-prediction-dataset?select=CVD_cleaned.csv>