**TREE ENSEMBLES**

**USING MULTIPLE DECISION TREES**

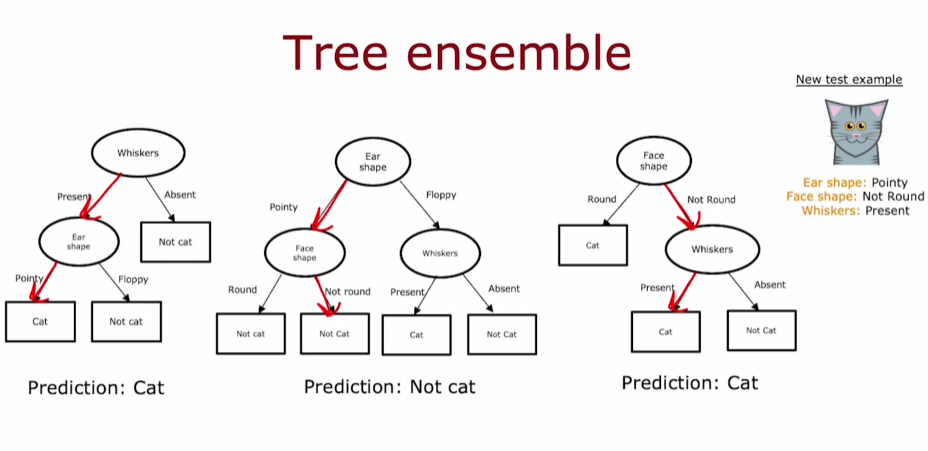
**Understanding Decision Trees**

* **A single decision tree can be highly sensitive to small changes in the training data, leading to different splits and potentially inaccurate predictions.**
* **For example, changing just one training example can result in a completely different decision tree structure.**

****

**Tree Ensembles for Robustness**

* **To improve accuracy, multiple decision trees are built, forming what is known as a tree ensemble, which allows for more reliable predictions.**
* **Each tree in the ensemble makes its own prediction, and the final decision is based on the majority vote among the trees.**

****

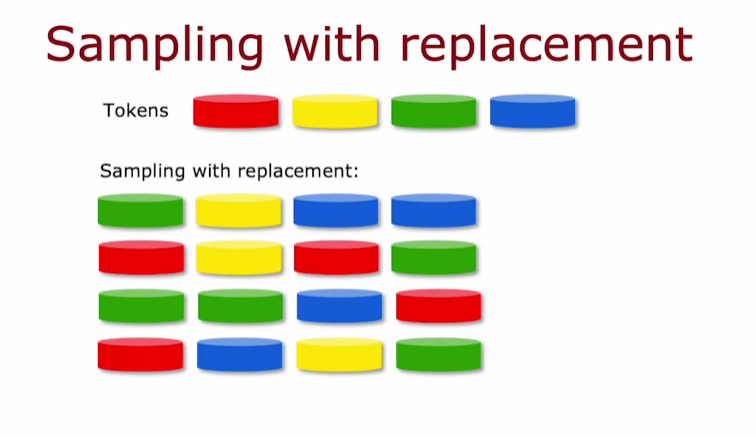
**Voting Mechanism**

* **When a new example is classified, each tree in the ensemble votes on the classification, reducing the impact of any single tree's prediction.**
* **This collective decision-making process enhances the overall robustness of the algorithm, making it less sensitive to individual data points.**

**SAMPLING WITH REPLACEMENT**

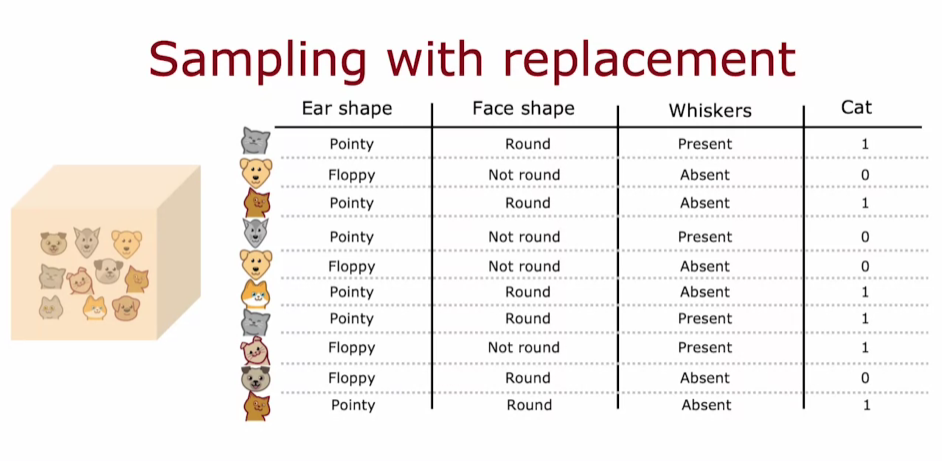
**Understanding Sampling with Replacement**

* **Sampling with replacement involves selecting items from a set, returning them after each selection, allowing for the same item to be chosen multiple times.**
* **An example using colored tokens illustrates this process, where tokens are drawn from a bag, replaced, and drawn again, leading to varied outcomes.**

****

**Application in Building Ensembles**

* **In constructing tree ensembles, multiple random training sets are created from an original dataset by sampling with replacement.**
* **This method allows for the generation of training sets that may not include all original examples and can contain duplicates, fostering diversity in the training data.**

****

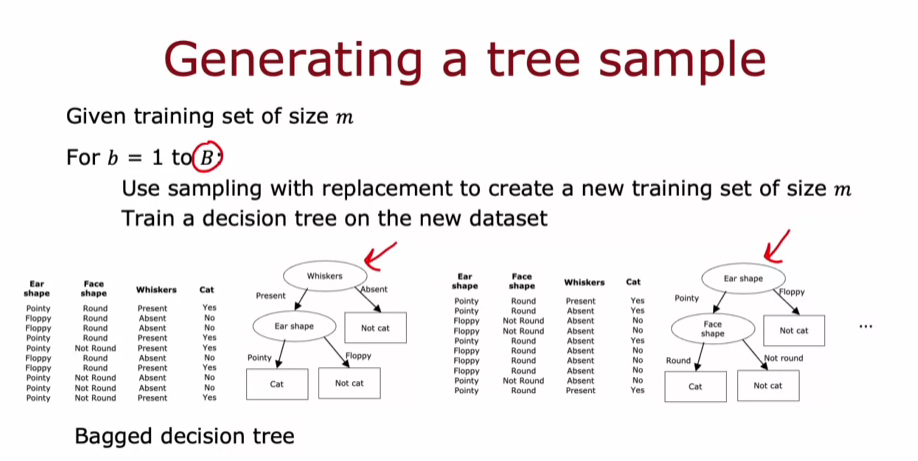
**Importance of the Process**

* **The replacement aspect is crucial; without it, the same set of examples would be used repeatedly, limiting the model's learning potential.**
* **This technique is a foundational step in creating robust machine learning models through ensemble methods.**

**RANDOM FOREST ALGORITHM**

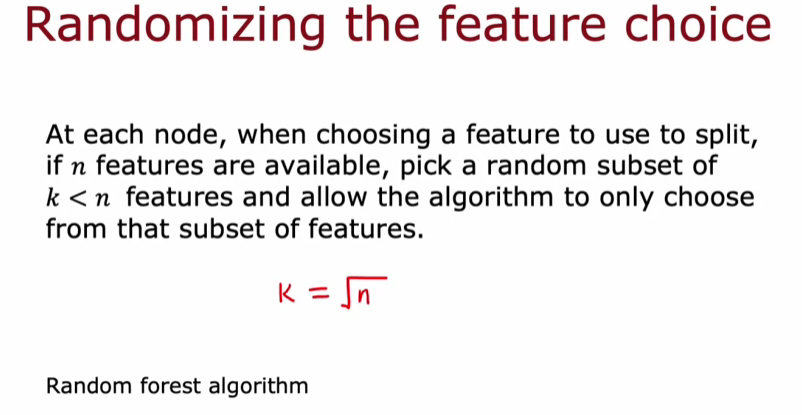
**Building a Random Forest**

* **The random forest algorithm generates multiple decision trees by creating new training sets through sampling with replacement from the original dataset.**
* **Each tree is trained on a slightly different dataset, leading to diverse decision trees that contribute to a more robust overall model.**

****

**Enhancing Randomness in Feature Selection**

* **To further improve the model, a random subset of features is selected at each node when splitting, rather than using all available features.**
* **This technique helps to ensure that the trees in the ensemble are more varied, which enhances the accuracy of predictions when the trees vote on the final output.**

****

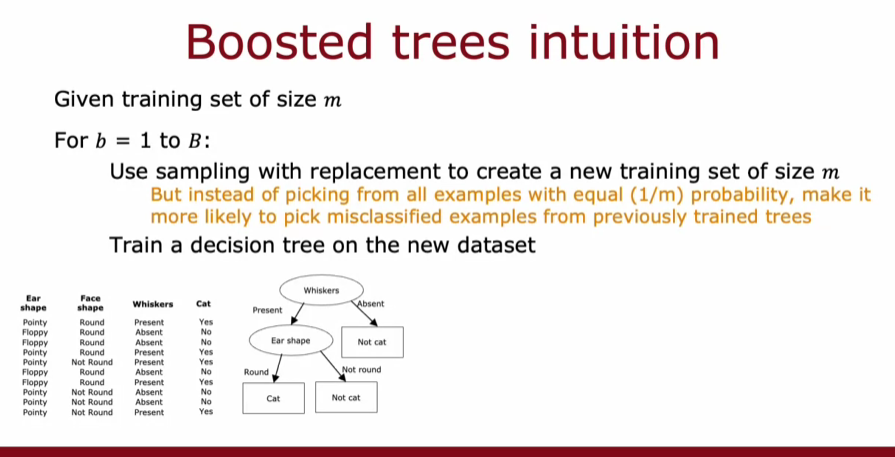
**Benefits of Random Forest**

* **The random forest algorithm is more robust than a single decision tree because it averages the predictions from multiple trees, reducing the impact of any single training set's peculiarities.**
* **It is particularly effective in handling larger datasets with many features, making it a valuable tool in machine learning.**

**XGBOOST**

**Understanding XGBoost**

* **XGBoost stands for Extreme Gradient Boosting and is widely used in machine learning competitions and commercial applications due to its speed and performance.**
* **The algorithm improves upon traditional decision tree methods by focusing on misclassified examples during training, similar to the concept of deliberate practice in learning.**

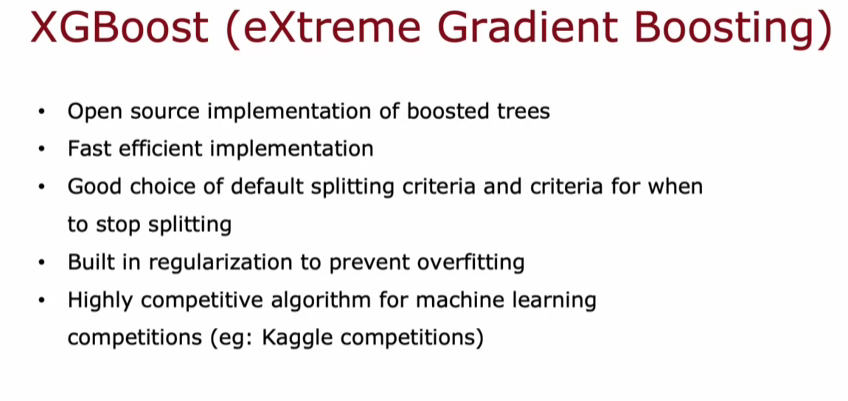
****

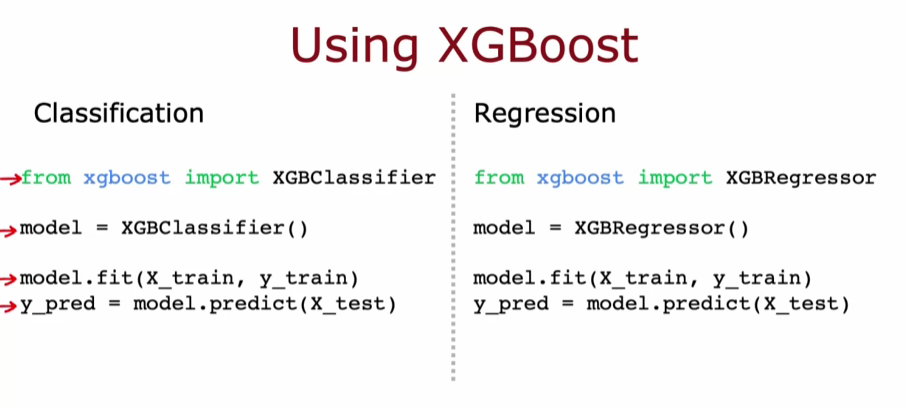
**Boosting Process**

* **In the boosting process, each new decision tree is trained on a modified training set that emphasizes examples the previous trees misclassified.**
* **This iterative approach allows the ensemble of trees to learn more effectively and improve overall accuracy.**

**Implementation and Usage**

* **XGBoost can be easily implemented using open-source libraries, allowing users to initialize a model and make predictions with minimal effort.**
* **It includes built-in regularization to prevent overfitting, making it a robust choice for various applications, including regression and classification tasks.**

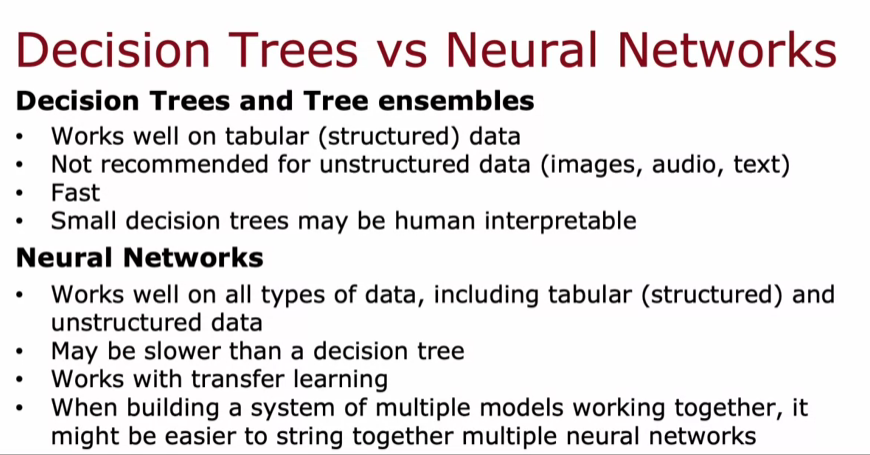
****

****

**WHEN TO USE DECISION TREES**

**Decision Trees and Tree Ensembles**

* **Decision trees excel with tabular or structured data, such as datasets resembling spreadsheets, making them suitable for tasks like housing price prediction.**
* **They are fast to train, allowing for quicker iterations in model development, and small decision trees can be interpretable, providing insights into decision-making processes.**

****

**Neural Networks**

* **Neural networks are versatile, performing well on both structured and unstructured data, including images, audio, and text, making them the preferred choice for unstructured tasks.**
* **However, they may require longer training times compared to decision trees, but they benefit from transfer learning, which enhances performance with smaller datasets.**

**Choosing the Right Algorithm**

* **For structured data, both decision trees and neural networks can be competitive, but for unstructured data, neural networks are generally favored.**
* **If using decision trees, XGBoost is recommended for most applications, while neural networks are advantageous for building systems with multiple models due to their training efficiency.**