AI200: APPLIED MACHINE LEARNING

DECISION TREES (CLASSIFICATION)

OVERVIEW & LITERATURE OF MACHINE LEARNING

Machine Learning

1. Spectrum of supervision	Supervised Learning		Unsupervised Learning			Reinforcement Learning
2. Types of problems	Regression	Classification	Clustering	Association	Dimensionality Reduction	
3. Types of algorithm / models	Linear regression Polynomial regression Neural Networks Decision Tree Random Forest XGBoost	Logistic Regression KNN Naïve Bayes SVM Neural Networks Decision Tree Random Forest XGBoost	Neural Networks K-Means Hierarchical Clustering Hidden Markov Models Gaussian Mixture	Apriori Algorithm FP-Growth Algorithm Eclat Algorithm	Neural Networks SVD PCA	
4. Types of applications	Forecasting Predictions Process optimization New Insights	Fraud Detection Image Classification Customer Retention Diagnostics	Recommendations Targeted Marketing Customer Segmentation	Market Basket Analysis	Feature Elicitation Structure Discovery Meaningful Compression Big Data Visualisation	Real-time Decisions Game AI Learning Tasks Skill Acquisition Robot Navigation

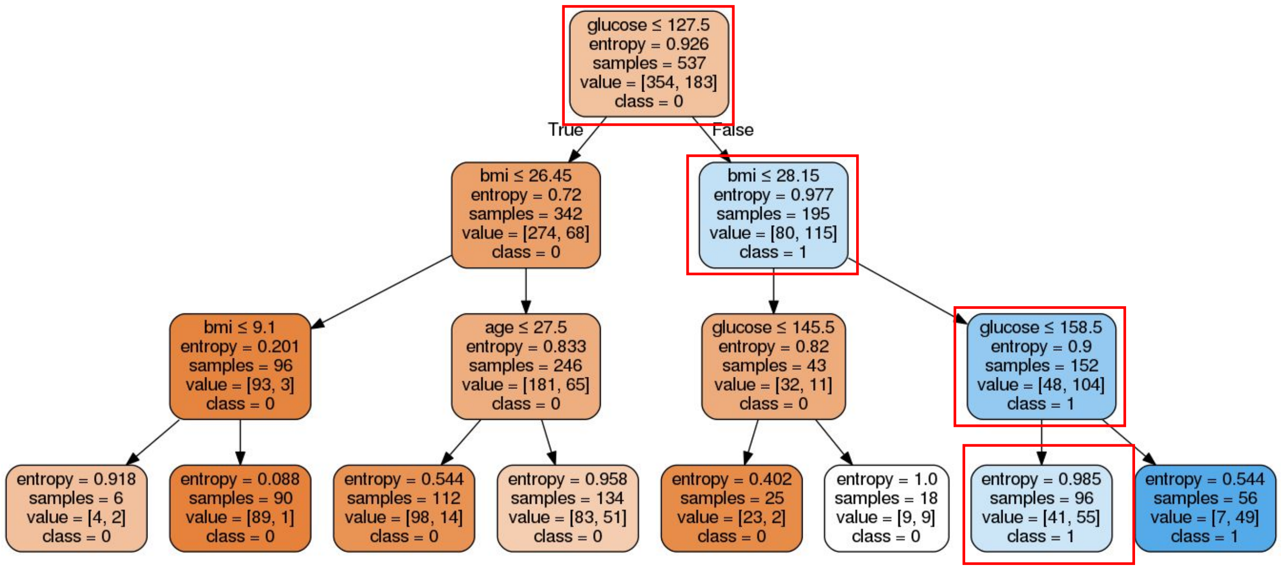
WHAT IS DECISION TREE: MAKING PREDICTIONS WITH A DECISION TREE (CLASSIFICATION)

- The idea behind decision tree (classification) is same as what we covered in session 3 for Decision Tree (regression):
    - A tree of rules is constructed based on the data
    - To make a prediction, we just compare the features against the rules to derive at the outcome
    - Let's use an example to see how we make a prediction with an already constructed tree:
        - Assuming we say we want to predict the chances of diabetes for someone with
        - (**BMI=35, glucose=160**)

Features **Outcome**

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
148	72	35	0	33.6	0.627	50	1
85	66	29	0	26.6	0.351	31	0
183	64	0	0	23.3	0.672	32	1
89	66	23	94	28.1	0.167	21	0
137	40	35	168	43.1	2.288	33	1

Plot features and outcome with the decision tree model



WHAT IS DECISION TREE: BROAD MECHANISM BEHIND DECISION TREE (CLASSIFICATION)

- So how did we construct the tree?

- For **Decision Tree (Regression),** recall that we made use of **partitioning** to generate the tree. And the determinant for how we partitioning is based on **RSS**.

- The mechanism for generating a **Decision Tree (Classification)** is similar. Here we use **partitioning** to generate the tree as well. However, we use the **gini impurity** as the determinant for how we partition

PARTITIONING + **RSS** =

PARTITIONING + **GINI IMPURITY** =

All you need to know for now is that Gini impurity calculates the misclassification rate of a split. And all that is calculated for you by the sklearn library's DecisionTreeClassifier!

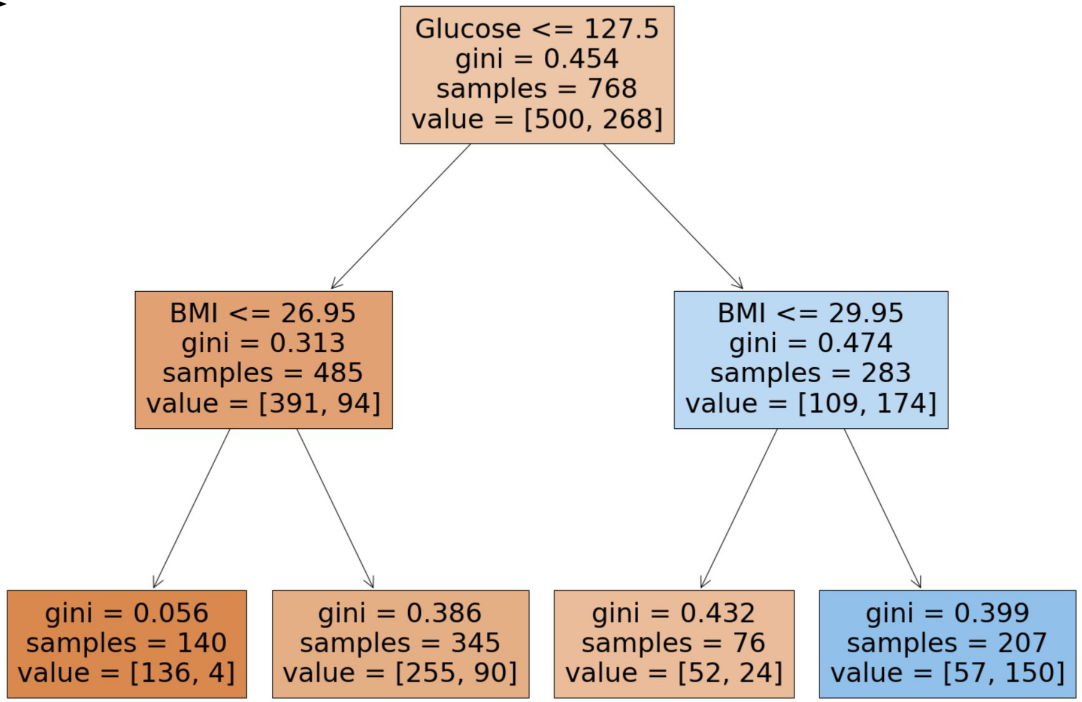
DECISION TREES (CLASSIFICATION)

MECHANISM BEHIND MODEL: PARTITIONING & GINI IMPURITY

MECHANISM BEHIND MODEL: PARTITIONING

- Let's create a simple decision tree with the diabetes dataset, using only '**Glucose**' and '**BMI**' to predict diabetes **Outcome**
- Using the final generated decision tree, we will show with step-by-step illustration how this classification decision tree was constructed

Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
148	72	35	0	33.6	0.627	50	1
85	66	29	0	26.6	0.351	31	0
183	64	0	0	23.3	0.672	32	1
89	66	23	94	28.1	0.167	21	0
137	40	35	168	43.1	2.288	33	1



MECHANISM BEHIND MODEL: PARTITIONING

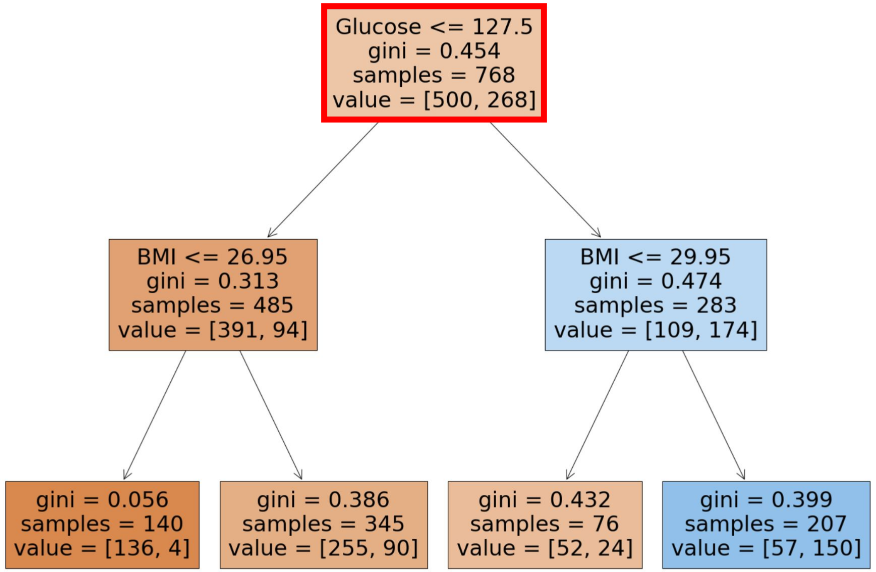
- **Step 1**: At the very initial split, the decision tree algorithm will try various features & values and calculate the impurity for each of the values. Finally it will select the feature & value that gives the lowest impurity

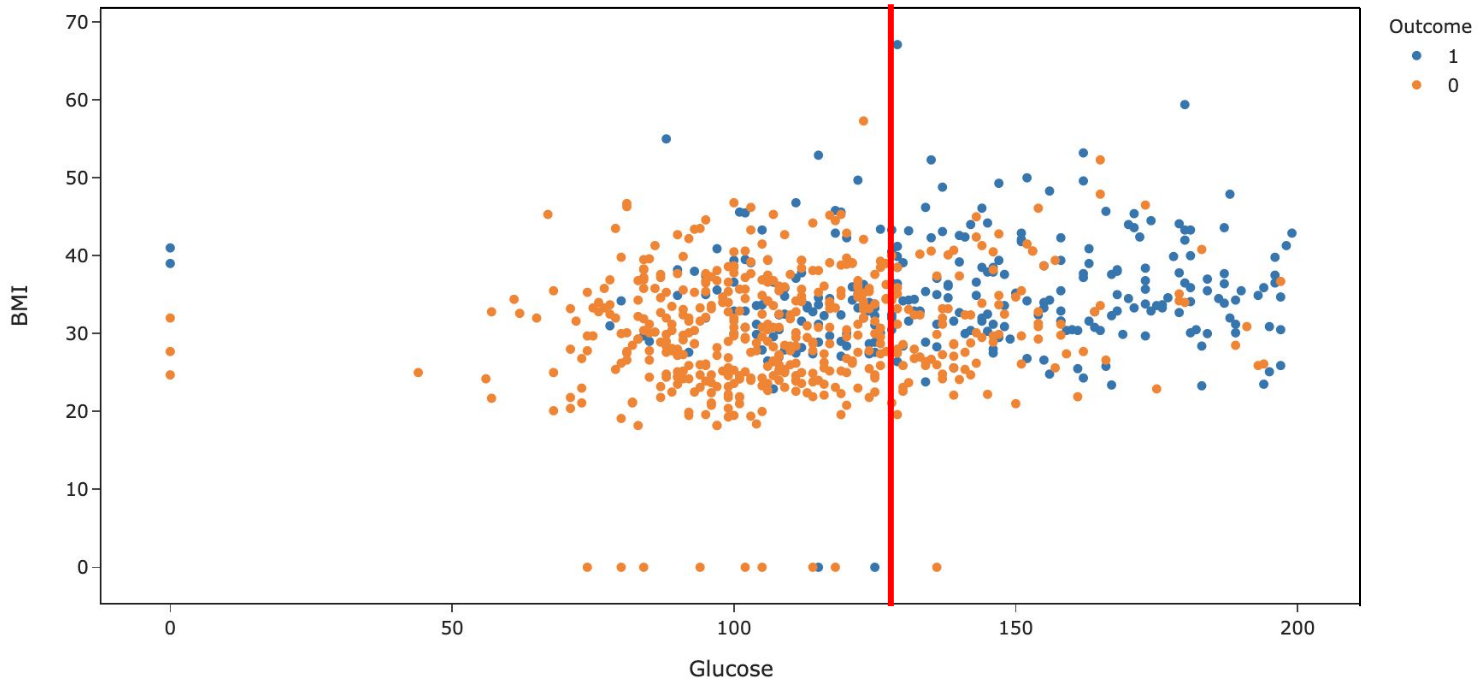
$$\text{Gini}(K) = \sum_{i \in N} P_{i,K}(1 - P_{i,K}) = 1 - \sum_{i \in N} P_{i,K}^2$$

- N is the list of classes (In this case $N = \{\text{'Yes'}, \text{'No'}\}$)
- K is the category
- $P_{i,K}$ is the probability of category K having class i

Out of N samples, how many are wrongly classified

This is the split with the lowest Gini Impurity: 0.454



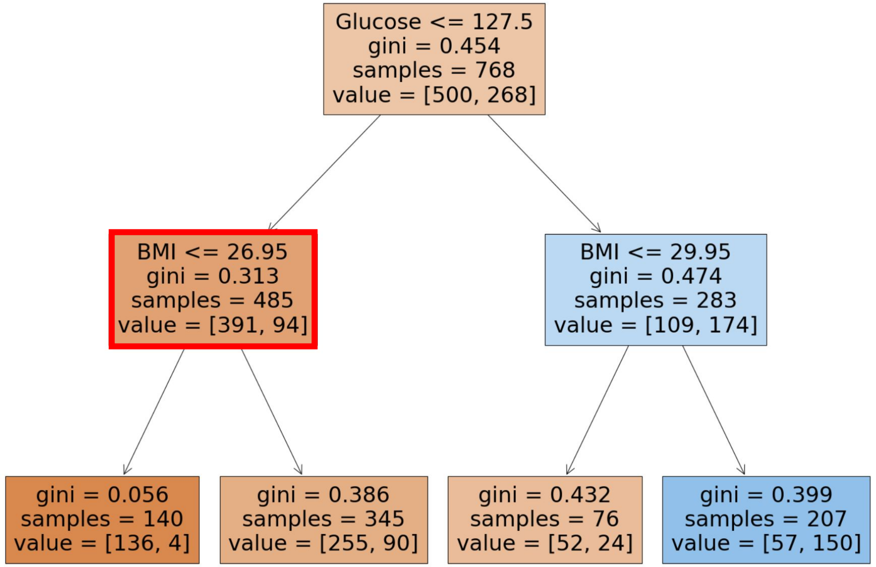


MECHANISM BEHIND MODEL: PARTITIONING

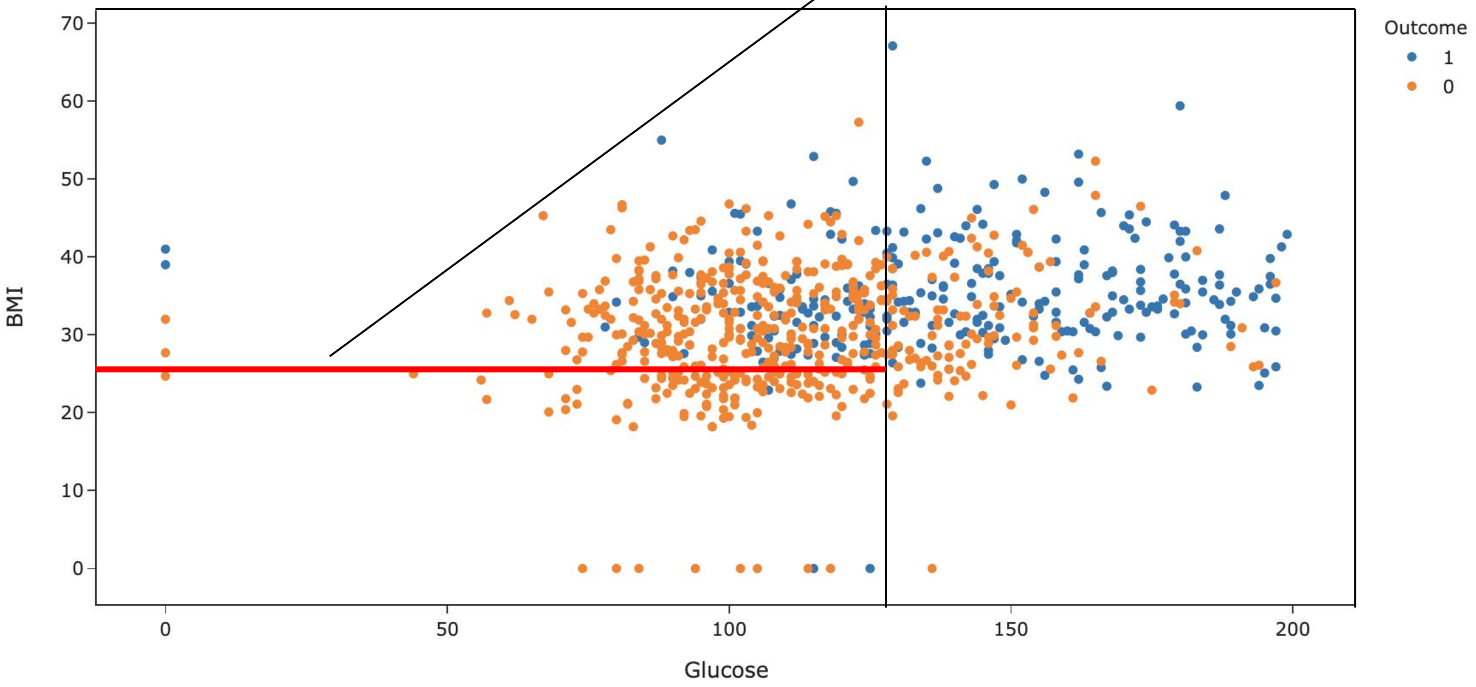
- **Step 2**: Building on the previous split, the decision tree algorithm would again try various features & values and calculate the Gini Impurity for each of the values. Finally it will select the feature & value that gives the lowest Gini Impurity

$$\text{Gini}(K) = \sum_{i \in N} P_{i,K}(1 - P_{i,K}) = 1 - \sum_{i \in N} P_{i,K}^2$$

- N is the list of classes (In this case $N = \{\text{'Yes'}, \text{'No'}\}$)
- K is the category
- $P_{i,K}$ is the probability of category K having class i



Ok this is the next split with the lowest Gini Impurity: 0.313

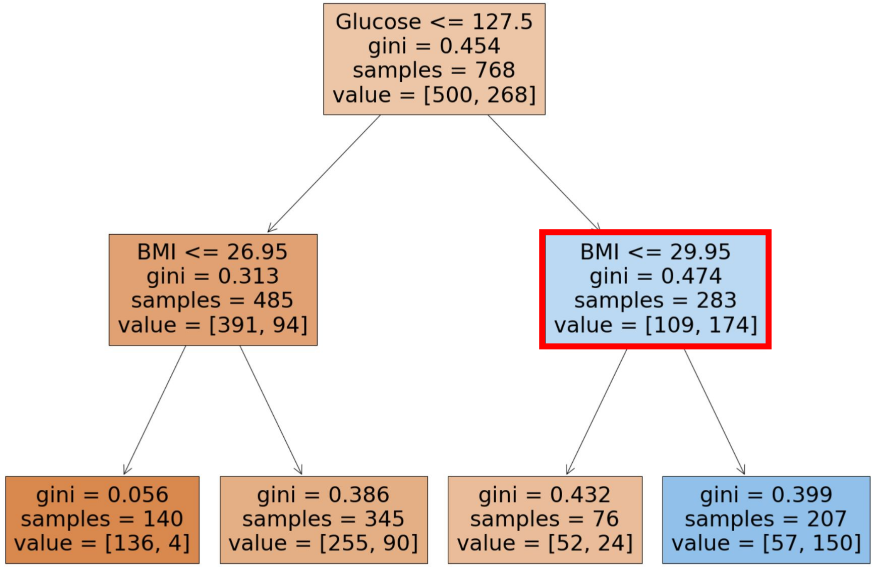


MECHANISM BEHIND MODEL: PARTITIONING

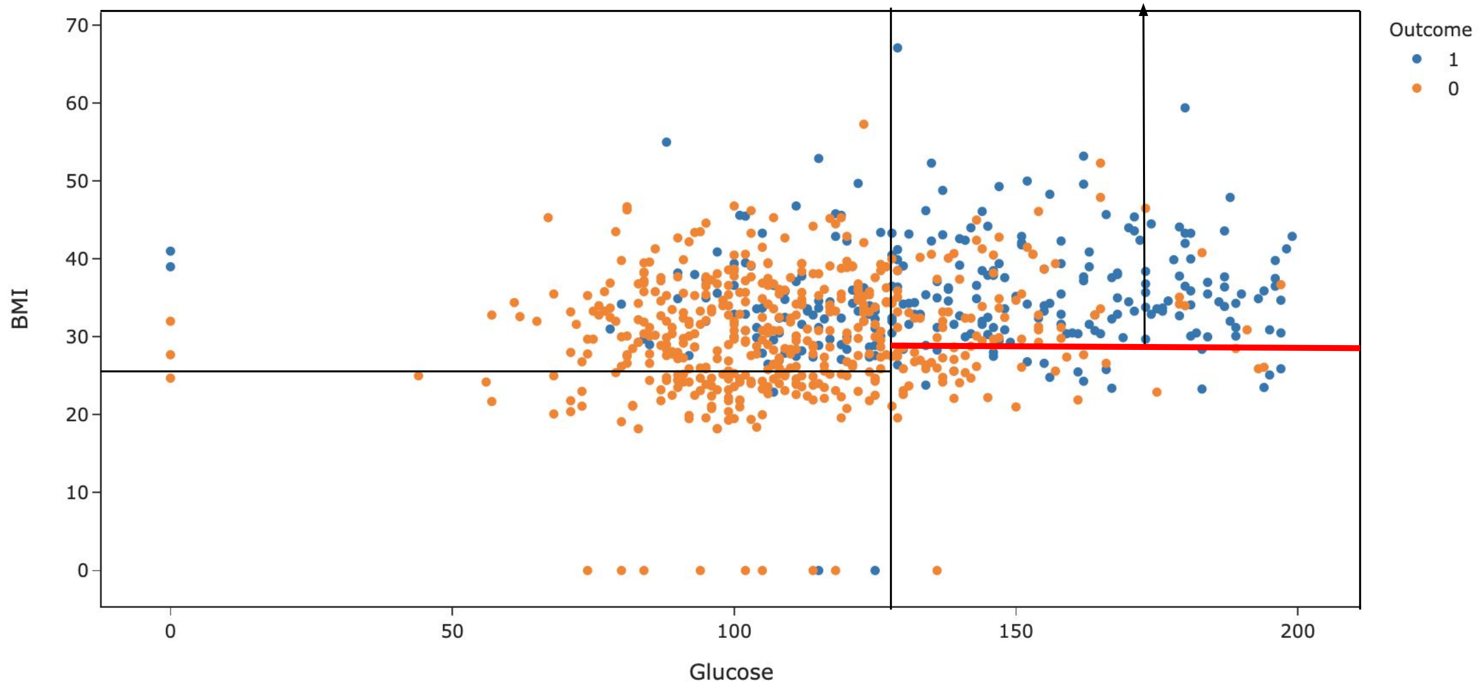
- **Step 2**: Repeat **step 2** to continue splitting until the splits no longer generates improvement in the Gini Impurity

$$\text{Gini}(K) = \sum_{i \in N} P_{i,K}(1 - P_{i,K}) = 1 - \sum_{i \in N} P_{i,K}^2$$

- N is the list of classes (In this case $N = \{\text{'Yes'}, \text{'No'}\}$)
- K is the category
- $P_{i,K}$ is the probability of category K having class i



Ok this is the next split with the lowest Gini Impurity: 0.474

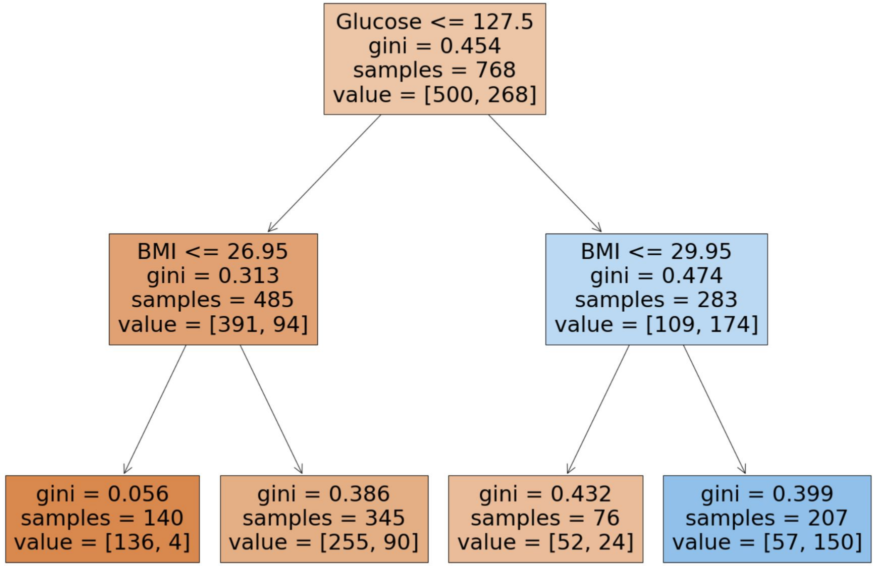


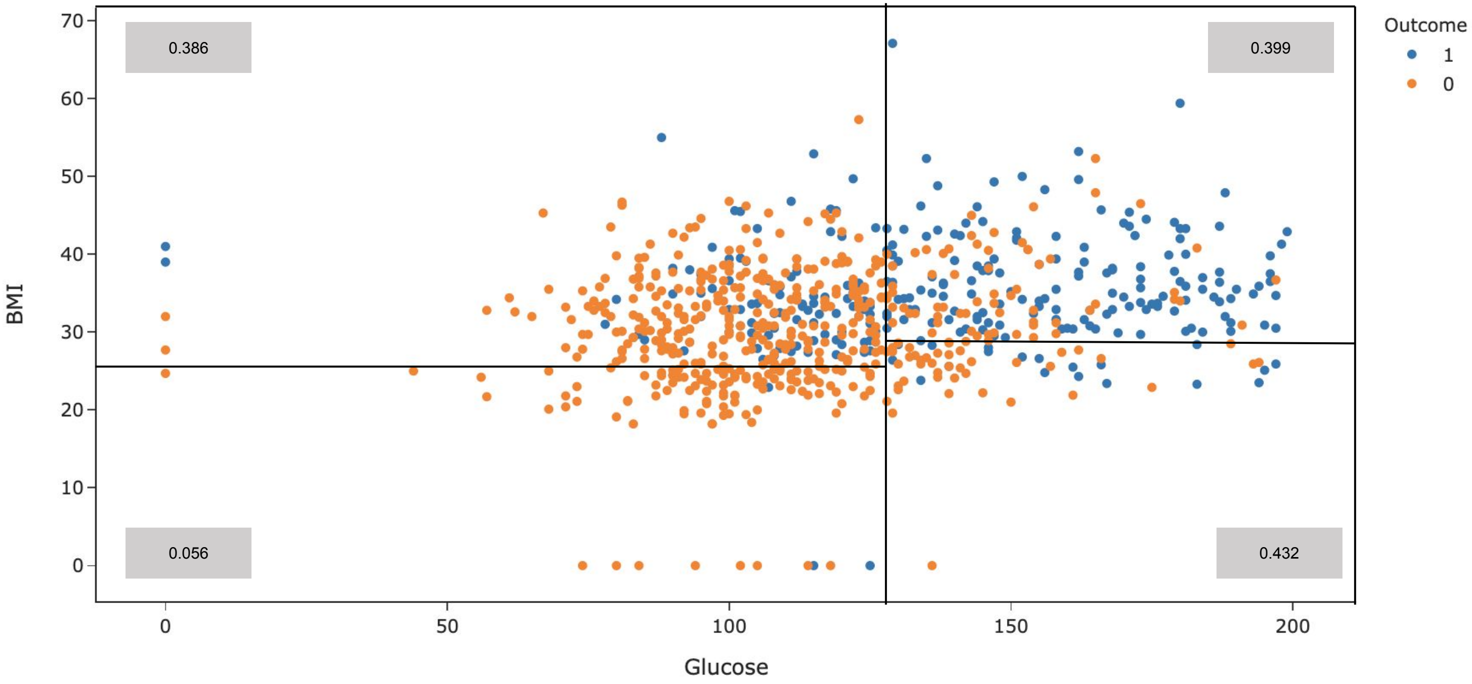
MECHANISM BEHIND MODEL: PARTITIONING

- **Step 3**: Repeat **step 2** to continue splitting until the splits no longer generates improvement in the Gini Impurity
- **Step 4**: From here on, the algorithm deems that any further splits would not result in any significant improvements in Gini Impurity, and hence it terminates the process for further splits

$$\mathrm{Gini}(K) = \sum_{i \in N} P_{i,K}(1 - P_{i,K}) = 1 - \sum_{i \in N} P_{i,K}^2$$

- N is the list of classes (In this case $N = \{\text{'Yes'}, \text{'No'}\}$)
- K is the category
- $P_{i,K}$ is the probability of category K having class i





MECHANISM BEHIND MODEL: PARTITIONING

- Let's say we want to use the newly constructed tree to do some prediction:
  - **BMI = 50, Glucose = 150**
  - **Diabetes Outcome = 1**

Probability of diabetes
= 150 / 207
= 0.72 (rounded to 1)