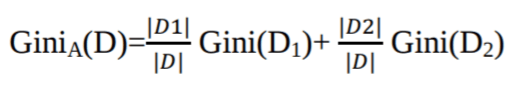
# To split the data the Gini method to create split points.

# It is used to calculate the inequality among values of a frequency distribution

Where, pi is the probability that a tuple in D belongs to class Ci.



The Gini Index considers a binary split for each attribute. You can compute a weighted sum of the impurity of each partition. If a binary split on attribute A partitions data D into D1 and D2, the Gini index of D is:



In case of a discrete-valued attribute, the subset that gives the minimum gini index for that chosen is selected as a splitting attribute.

For a continuous case, we find a splitting attribute by selecting adjacent values as apossible split point with a smaller gini-index.



The attribute with minimum Gini index is chosen as the splitting attribute.

For our dataset, to split the data,

# Split dataset into training set and test set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=1) # 70% training and 30% test

The data is splited by use of a function traint\_test\_split(). We pass the target size and the test size.

Build of decision tree

# Create Decision Tree classifer object

clf = DecisionTreeClassifier()

# Train Decision Tree Classifer

clf = clf.fit(X\_train,y\_train)

#Predict the response for test dataset

y\_pred = clf.predict(X\_test)

Importance of using Gini method

* It is easy to use with a large dataset.
* Gini method is is calculated by subtracting the sum of squared probabilities of each class from one
* Feature with a smaller class Gini index is chosen for a split.

In our dataset;

#The dataset is split in variable of target and variable

feature\_cols = ['pregnant', 'insulin', 'bmi', 'age','glucose','bp','pedigree']

X = df[feature\_cols] # Features

y = df.label # Target variable

In the dataset is split in variable of target and variable