**How would you validate a model you created to generate a predictive model of a quantitative outcome variable using multiple regression?**

At the initial stage, dataset needed to be divided into train and test. In that train data set will have 60 – 80 percent of data. And the remaining data in the test dataset. After splitting the dataset, now we need to build a model on the train dataset my using R commands in R studio and analyzing the train datasets data by using plots. Now our main task is to validate that analysis. To do that we need to predict data for test dataset using confusion matrix and calculating accuracies we can validate the predicted data. That is my approach to validate the model I have created.

**How can you prove that one improvement you've brought to an algorithm is really an improvement over not doing anything?**

The ways to improve our analysis in an algorithm is by gathering large data instead of assumptions and weak dependencies in the datasets attributes, treating missing and noisy data which is the biggest issue in predicting future outcomes and using that data, we need to build a model in such way that it should meet our requirements, we can also use multiple algorithms to get better results because for each algorithm we get the data for different analysis techniques.

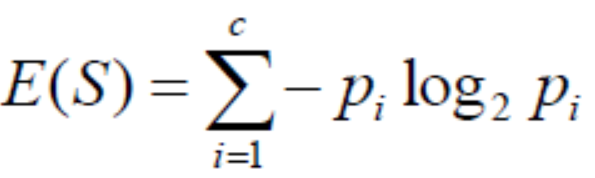
**Pick an algorithm you like and walk us through the math and then the implementation of it, in pseudo-code.**

I have Chosen Decision Tree in that there are few criteria’s like Entropy, information Gain, Gini index, Gain Ration, Reduction in Variance, Chi-Square

These criterions will calculate every value in the attribute. The values are sorted and attributes are also place in the tree by following the sequence based on these criterions to pick the root node, branch nodes and leaf node.

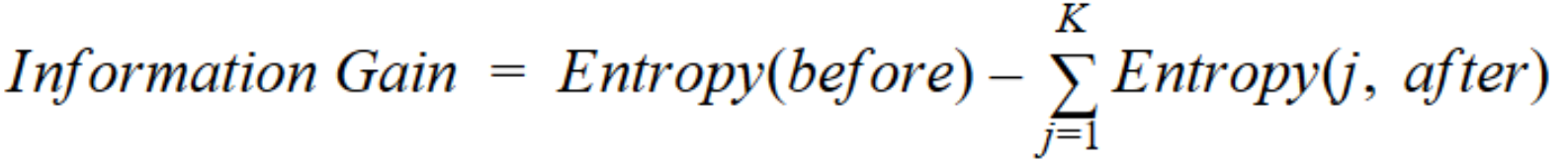
Entropy:

This is a measure of the randomness in the information being processed. The higher the entropy the harder it is to draw a conclusion from the processed information.





Information Gain: This is a statistical property that measures how well a given attribute separates the training examples according to their target classification. The main decision to construct a decision tree is to find that attribute which has the most information gain and less entropy curve.



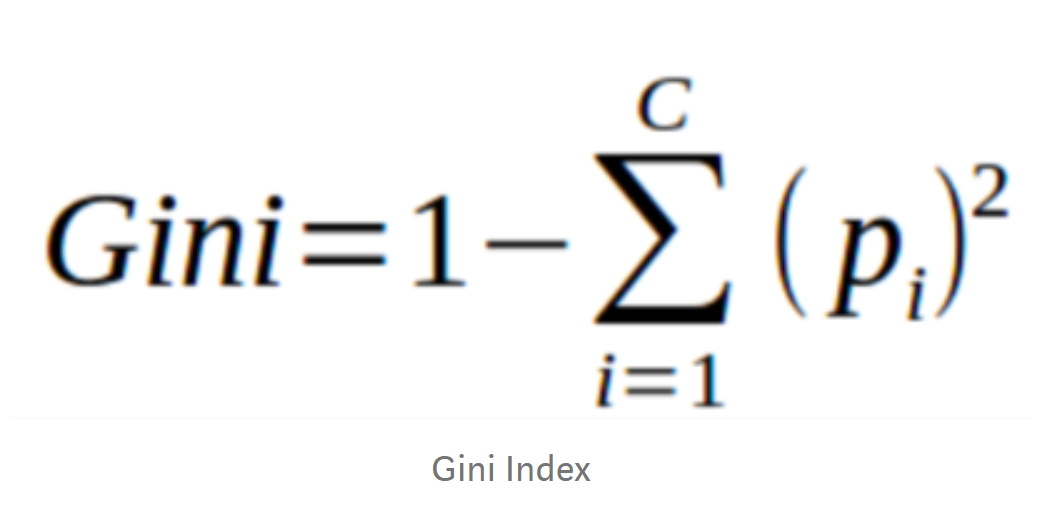
**Gini Index:**

This works with the categorical target variables like success or failure. This gini index perform the binary splits.

Steps to calculate Gini Index for a split:

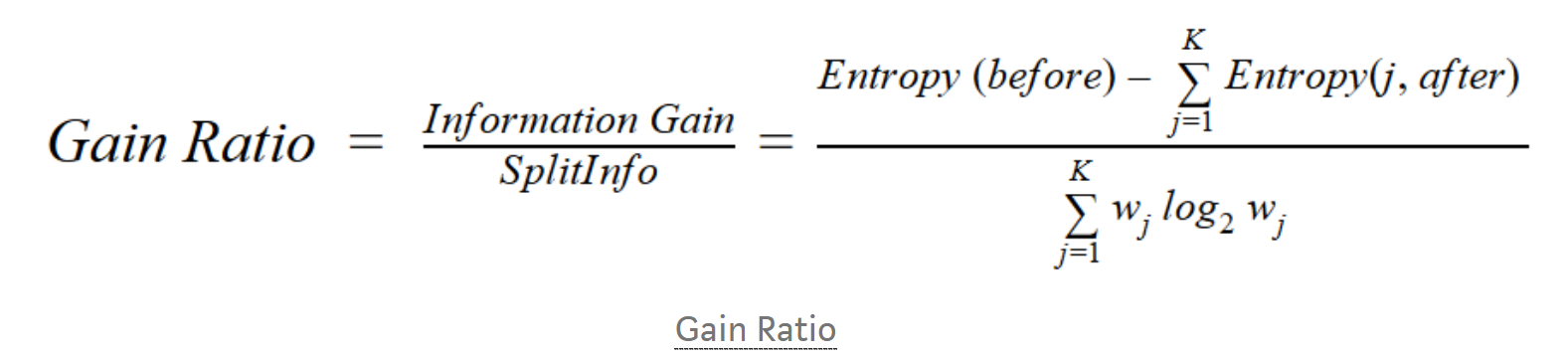
Calculate Gini for sub nodes, using the below formula for success(p) and failure (q) (P2+ q2).

Calculate the gini index for split using the weighted Gini Score of each node of that split.



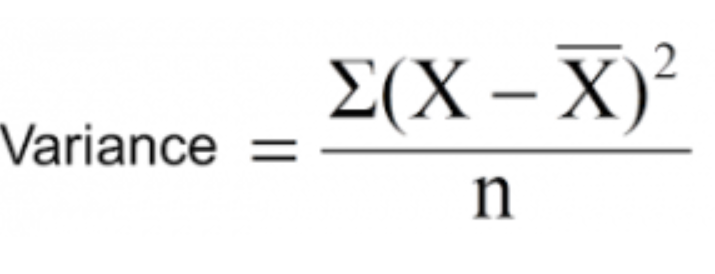
**Gain Ratio:**

This is the modification of information gain which reduce the bias decision and give the best opinion to make decision.



**Reduction in variance:**

This is used for the continuous target variables especially for Regression algorithms.

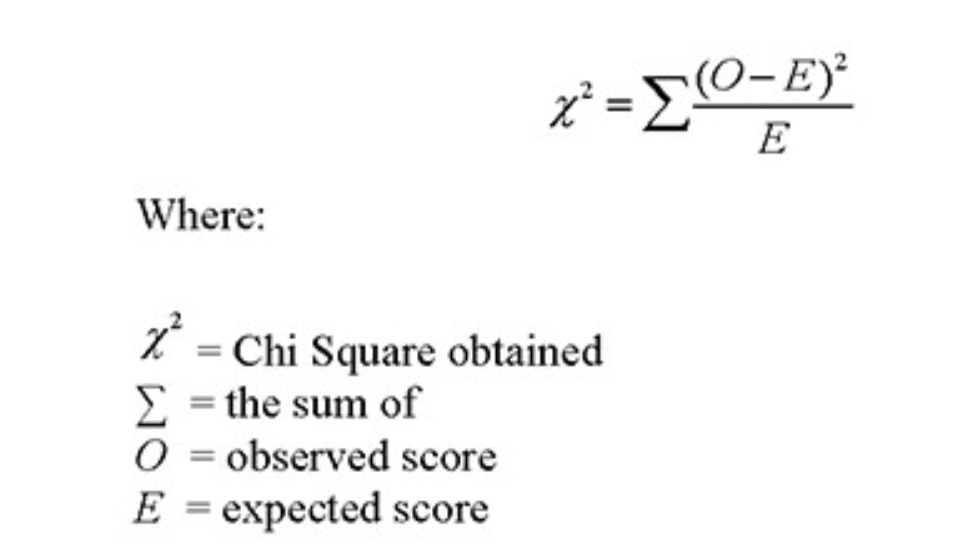


Steps to calculate Variance:

Calculate variance for each node.

Calculate variance for each split as the weighted average of each node variance.

**Chi-square:** This works with categorical target variables like success or failure. Higher the value of chi-square higher the statistical significance of difference between sub node and parent node.



Steps to calculate Chi-square for a split:

Calculate chi-square for an individual node by calculating the deviation for success and failure both.

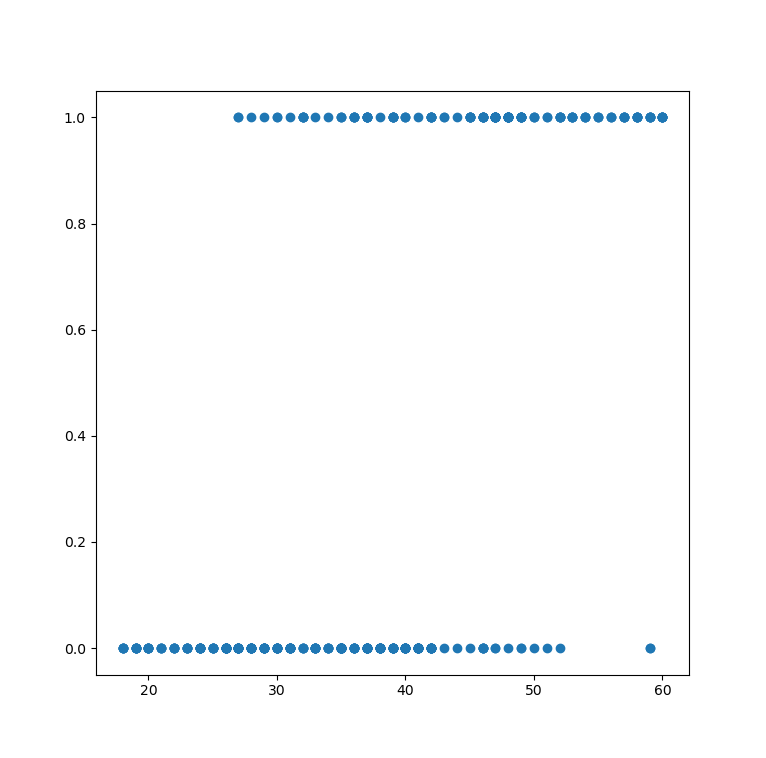
Calculated Chi-square of split using sum of all chi-square of success and failure of each node of the split.

**Discuss how you go about feature engineering (look for both intuition and specific evaluation techniques).**

Feature Engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. These features are used to improve the performance of the machine learning techniques. These features can be considered as the applied machine learning itself.

**How can you build a simple logistic regression model in Python?**

import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model\_selection import train\_test\_split  
from math import exp  
  
plt.rcParams["figure.figsize"] = (10, 6)  
data = pd.read\_csv("Social\_Network\_Ads.csv")  
data.head()  
plt.scatter(data['Age'], data['Purchased'])  
plt.show()  
x\_train, x\_test, y\_train, y\_test = train\_test\_split(data['Age'], data['Purchased'], test\_size=0.20)  
  
  
def normalize(X):  
 return X - X.mean()  
  
  
def predict(X, b0, b1):  
 return np.array([1 / (1 + exp(-1 \* b0 + -1 \* b1 \* x)) for x in X])  
  
  
def logistic\_regression(X, Y):  
 X = normalize(X)  
  
 b0 = 0  
 b1 = 0  
 L = 0.001  
 epochs = 150  
  
 for epoch in range(epochs):  
 y\_pred = predict(X, b0, b1)  
 D\_b0 = -2 \* sum((Y - y\_pred) \* y\_pred \* (1 - y\_pred))  
 D\_b1 = -2 \* sum((Y - y\_pred) \* y\_pred \* (1 - y\_pred) \* X)  
 b0 = b0 - L \* D\_b0  
 b1 = b1 - L \* D\_b1  
 return b0, b1  
 b0, b1 = logistic\_regression(x\_train, y\_train)  
 Print(b0, b1)



**References:**

1. <https://www.youtube.com/watch?v=l8VEth6leXA>
2. <https://www.kaggle.com/rakeshrau/social-network-ads/data>
3. <https://www.google.com/search?q=feature+engineering&oq=feature+&aqs=chrome.3.69i57j0l4j69i60l3.4383j0j4&sourceid=chrome&ie=UTF-8>
4. <https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4>
5. <https://www.quora.com/How-would-you-validate-a-model-you-created-to-generate-a-predictive-model-of-a-quantitative-outcome-variable-using-multiple-regression>
6. <https://www.quora.com/Can-you-pick-an-algorithm-you-like-and-walk-me-through-the-math-and-then-do-the-implementation-of-it-in-pseudo-code>