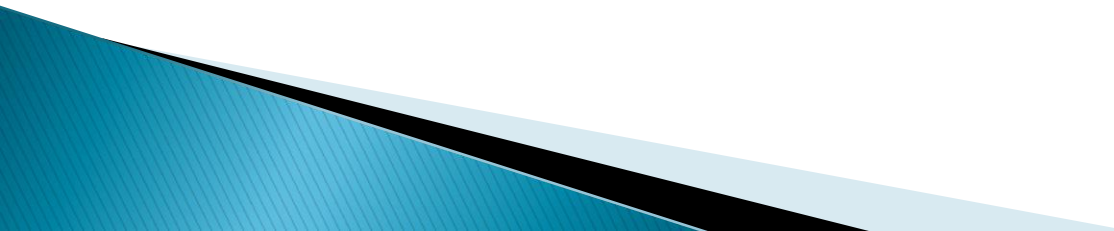
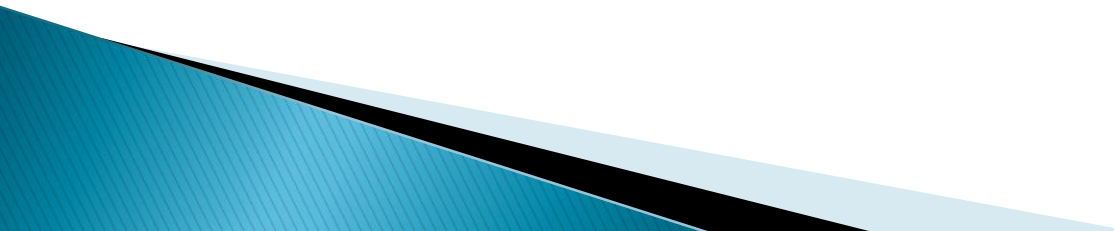


PLANT GENE CLASSIFICATION AND FUNCTIONALITY PREDICTION

OBJECTIVE:

- ▶ Gene classification or identification refers to recognize the area through certain computational process where genomic DNA can found.
 - ▶ DNA sequence is available with us then the objective is to predict the use of function of gene from basic data.
- 

CONSTRAINTS:

- ▶ Prediction of short genome annotation in given data.(short gdna may occur at many places in given patterns)
 - ▶ Super Imposition or lateral Imposition of two genes.
 - ▶ Breakage of gene syndrome may cause damage to plant.
- 

TEST DATA SET / FRAMES:

SEQUENCE	CELL CYCLE	COLOUR	PLANT GENE
TCGCTGC	17	RED	YES
TCGTCGT	19	YELLOW	NO
GCTGCCC	14	GREEN	YES
TGGCTTG	28	GREEN	S
ACTCTGC	20	RED	YES

FACTS:

- ▶ Heridity
- ▶ Must give the genetic material to younger genes.

RULES:


- Must give good yield.
 - Proper nutrients must be maintained
- 

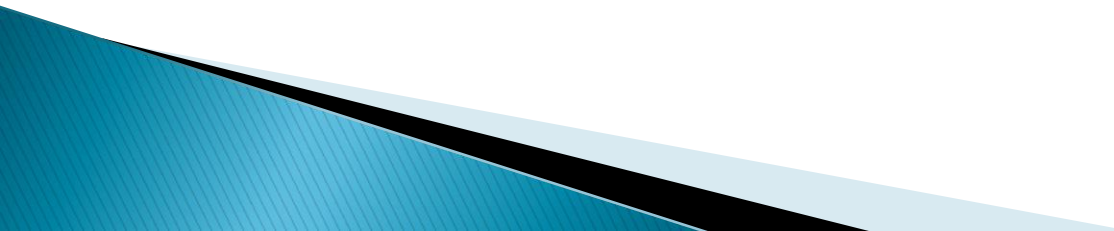
DECISION TREE:

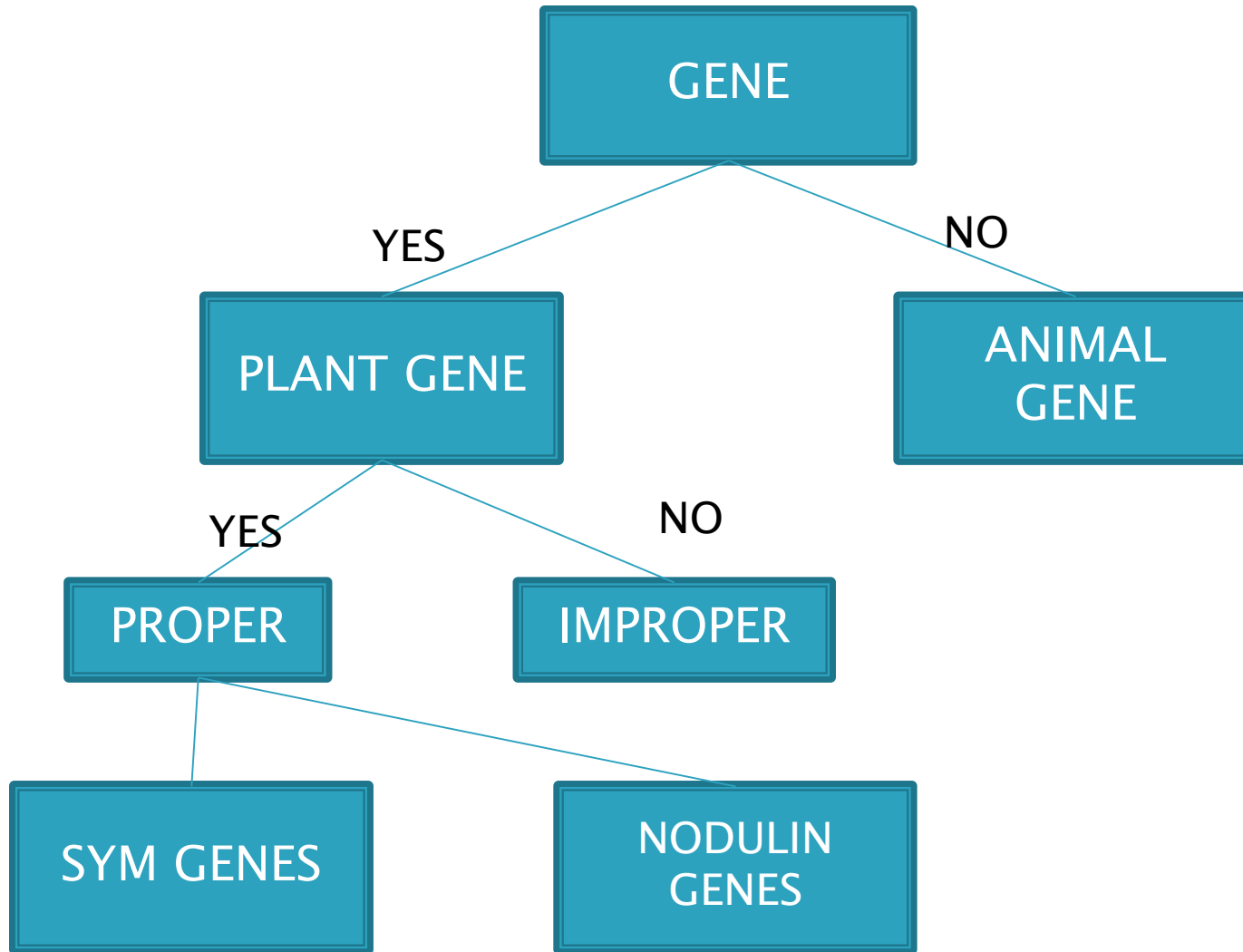
Get list of rows(dataset) which are taken into consideration for making decision tree(recursively at each nodes).

Calculate uncertainty of our dataset or Gini impurity or how much our data is mixed up etc.,

Generate the list of all questions which need to be asked at the node. Partition of rows into true rows and false rows based on each question asked.



- ▶ Calculate the information gain based on Gini impurity and partition of Data from previous step.
 - ▶ Update highest information gain based on each question asked and update best question based on the information gain.
 - ▶ Divide the node and best question. Repeat again from step 1 again until we get pure node.
- 

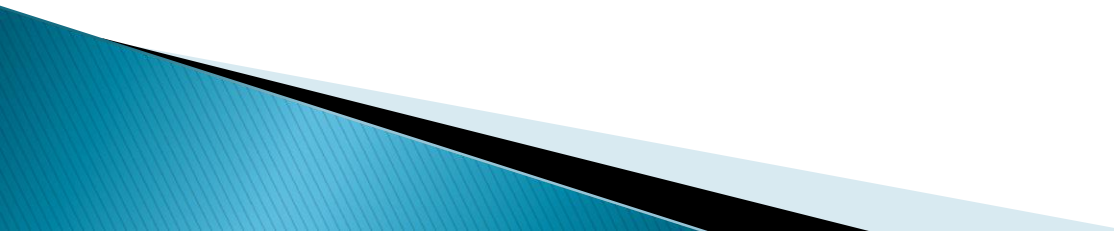


APPROACH TO SOLVE THE PROBLEM:

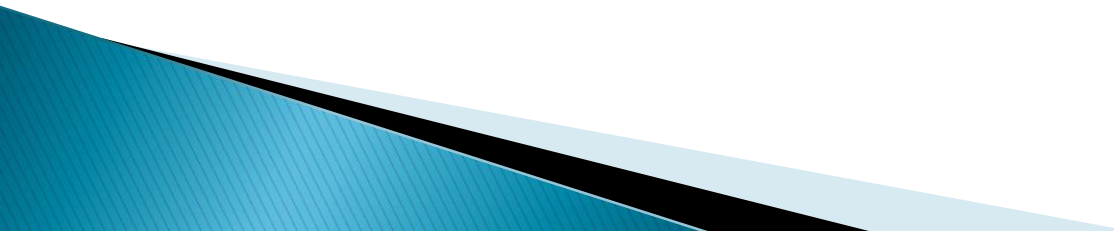
Our approach to solve the problem is

Lower the entropy higher the information gain

So, we are taking features having more information and gain than any other features. When the desired plant gene has more pattern annotation in any dataset, it gives green indication.



PYTHON PACKAGES :

- ▶ numpy
 - ▶ statistics
 - ▶ pandas
 - ▶ pymc3
 - ▶ arviz
- 

THANK YOU