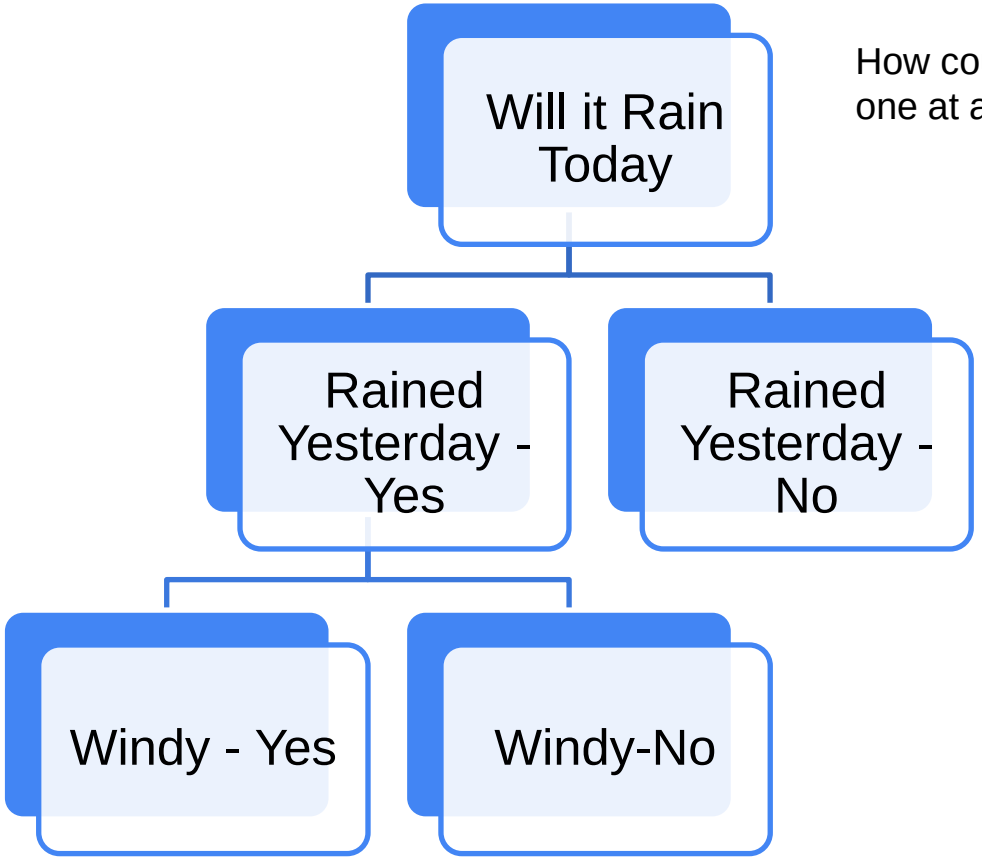
Workshop Trees

The idea behind a decision tree

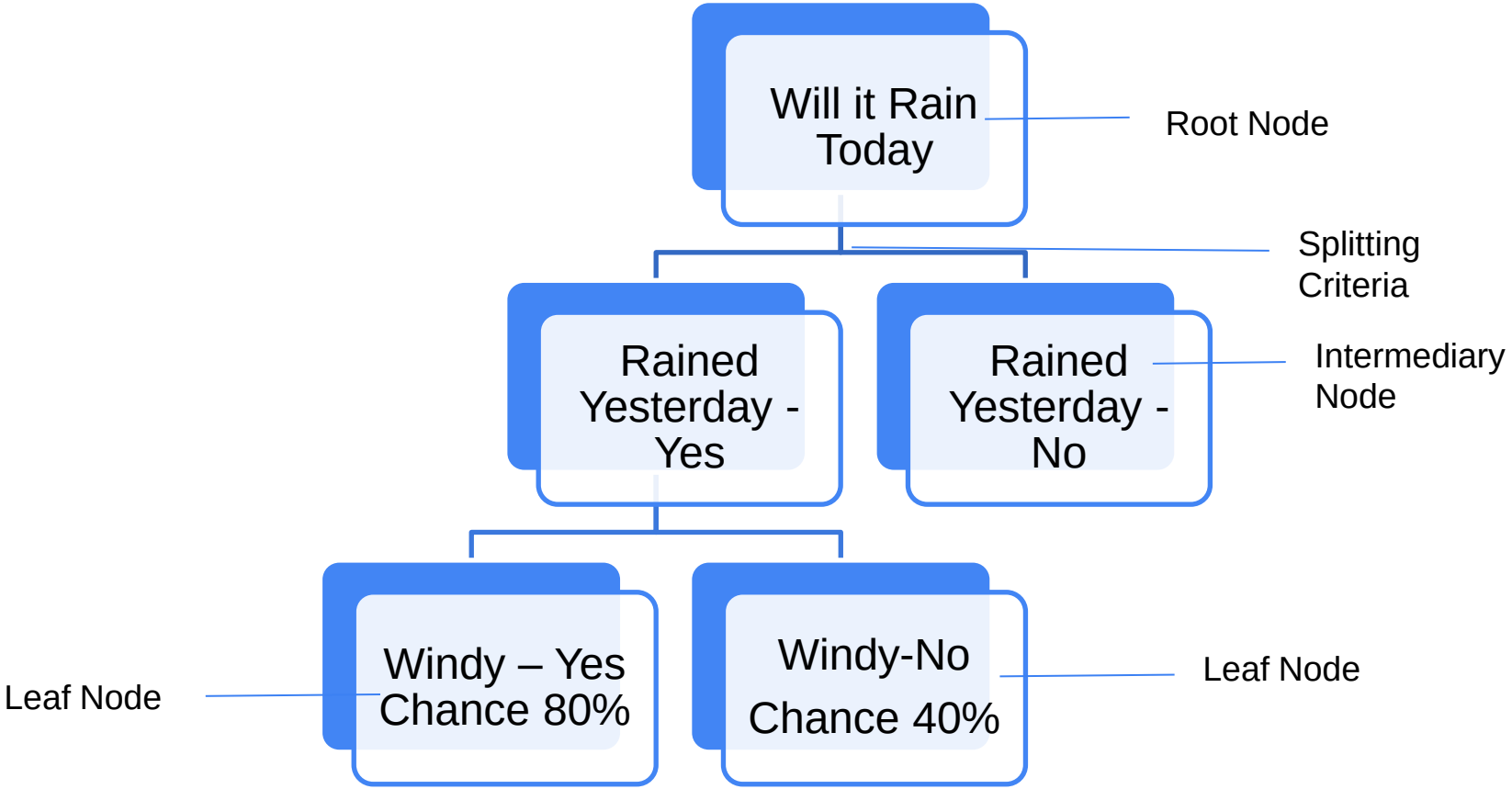


How could different variables influence outcome one at a time

Applications:

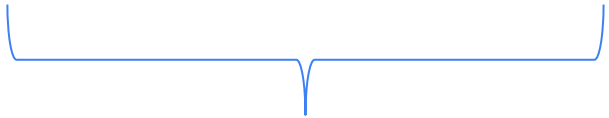
1. Given variables to make decisions
2. To also understand how certain groups differ from one another

Components


Will it Rain Today
Root Node
Splitting Criteria
Rained Yesterday - Yes
Rained Yesterday - No
Intermediary Node
Windy – Yes Chance 80%
Windy-No Chance 40%
Leaf Node
Leaf Node

An excel example

gender	Location	bought_last_30_days	buy
M	Metro	Y	buy
M	Non-Metro	N	no_buy
F	Non-Metro	N	no_buy
F	Metro	N	buy
M	Metro	Y	no_buy
M	Metro	N	no_buy



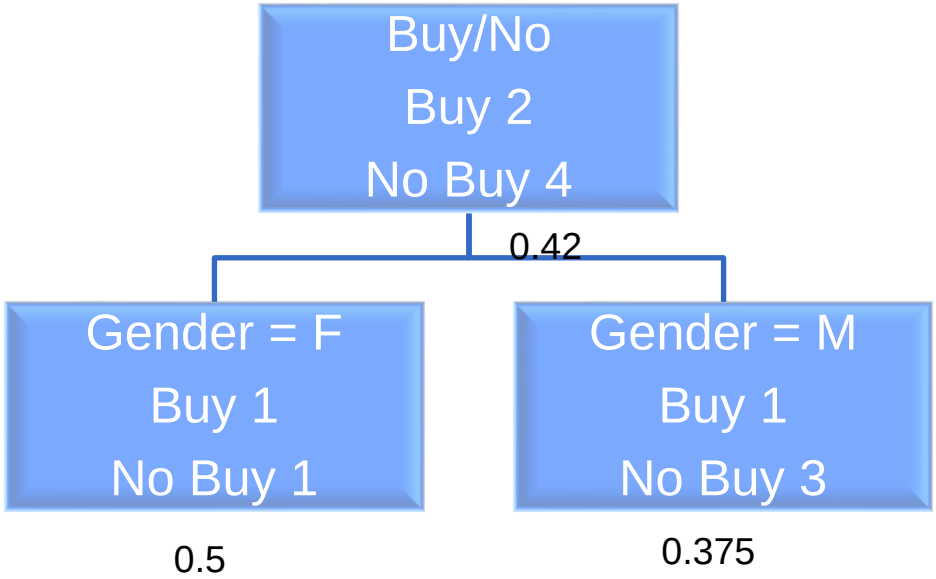
Dep. Variable

Independent Variables

Quick concept

1. We need to split the dataset one variable at a time
2. Keep doing this recursively
3. Identify a stopping condition

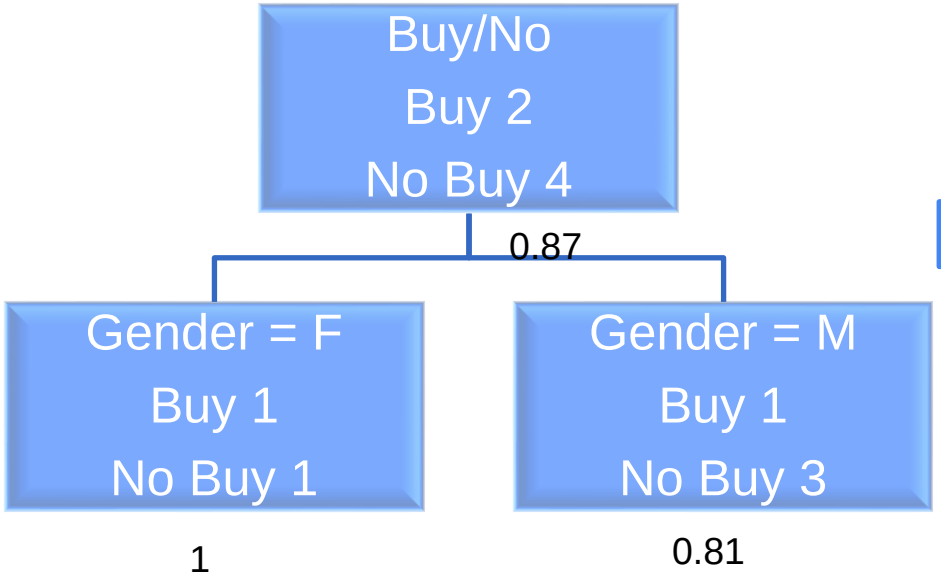
Splitting Criteria – Gini Index



Gini Impurity = $1 - \sum p(x_i)^2$

[Google Sheet for workings]

Splitting Criteria – Information Gain



Entropy = ∑-plog(p)

Information Gain = 1- Entropy

Google Sheet for workings

From Excel – Example (Entropy)

COUNTA of buy	*buy*							
Location	buy	no_buy	Grand Total	prob1	prob2	Entropy	Weights	Weighted Entropy
Metro	2	2	4	0.5	0.5	1	0.6666666667	0.6666666667
Non-Metro		2	2	0	1	0	0.3333333333	0
Grand Total	**2**	**4**	**6**			1		**0.6666666667**

Variable	**Entropy**	**Info gain**
gender	0.874	0.126
location	0.667	0.333
bought_last_30_days	0.874	0.126

Regression Trees

1. Take a feature and split
2. Get Variance of left node
3. Get Variance of right node
4. Weighted average of variance across all nodes

Variance = (X-mu)^2/N

Pruning:

Cost Complexity alpha :

Tree score = (Sum of Error)^2 + alpha * number of terminal nodes

Different sub trees– **Find the best alpha based on cross validation results**

Google Sheet for workings

Ref : https://www.youtube.com/watch?v=D0efHEJsfHo

Case Study 1

Credit card company wants to pre-approve its customers. It has many relevant details about the customers, we need to decide whether they should approve or not

Rows: Customers

Columns: CIBIL score, user location, age, income, gender, income, device etc

Labels: ‘Good’, ‘Bad’

Problem - Dataset

	Unnamed: 0	income	age	experience	bureau_score	married	house_ownership	car_ownership	risk_flag	profession	
0	19607	2514921	31.00000	4.00000	651.00000	single	rented	no	0	Psychologist	
1	75516	7047674	28.00000	4.00000	526.00000	single	rented	yes	0	Economist	R
2	63804	2749317	30.00000	2.00000	526.00000	single	rented	no	0	Secretary	R
3	63676	7378274	24.00000	0.00000	764.00000	single	rented	no	0	Flight attendant	
4	50914	9574585	27.00000	5.00000	739.00000	single	rented	yes	0	Technician	

Independent Variables:

Index(['income', 'age', 'experience', 'bureau_score', 'married', 'house_ownership', 'car_ownership', 'profession', 'city', 'state', 'current_job_years', 'current_house_years', 'device'], dtype='object')

Dependent Variable

```
0    236567
1     43433
Name: risk_flag, dtype: int64
```

Lets look at the data

income	age	experience	bureau_score	married	house_ownership	car_ownership	risk_flag	profession
2514921	31.0	4.0	651.0	single	rented	no	0	Psychologist
7047674	28.0	4.0	526.0	single	rented	yes	0	Economist
2749317	30.0	2.0	526.0	single	rented	no	0	Secretary
7378274	24.0	0.0	764.0	single	rented	no	0	Flight attendant
9574585	27.0	5.0	739.0	single	rented	yes	0	Technician

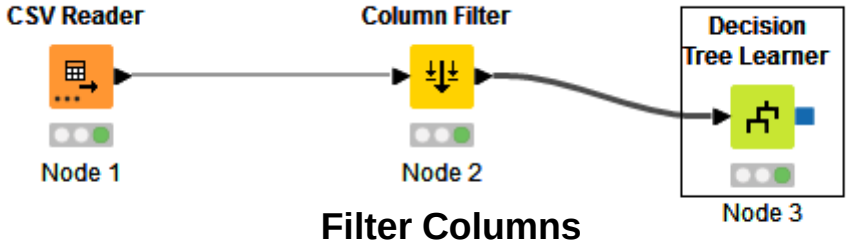
Label

	city	state	current_job_years	current_house_years	device
	Chandrapur	Maharashtra	4.0	14.0	Oppo
	Ramagundam[27]	Telangana	3.0	13.0	Xiaomi
	Ramagundam[27]	Telangana	2.0	14.0	samsung
	Adoni	Andhra Pradesh	0.0	11.0	samsung
	Imphal	Manipur	5.0	10.0	Vivo

```
0    236567
1     43433
Name: risk_flag, dtype: int64

0    0.844882
1    0.155118
Name: risk_flag, dtype: float64
```

Basic Decision Tree (With KNIME)



Filter Columns

Configure Dependent Variable

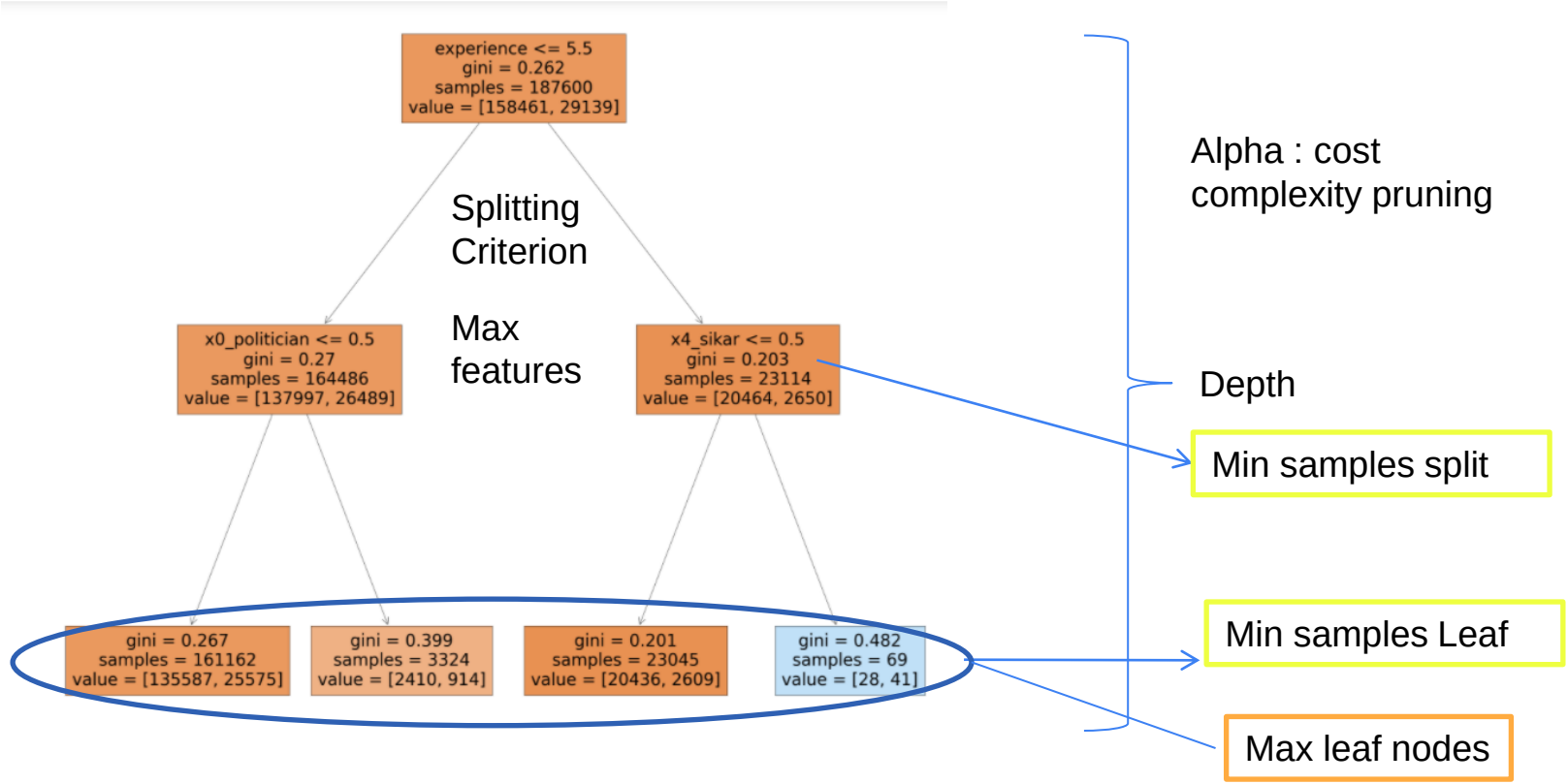
Over to Python

Decision Tree with Sklearn

decistion_tree.png

Algorithm	Test	Training
Decision Tree - Default	0.527	0.527

Hyper Parameters for decision trees



Over to Python

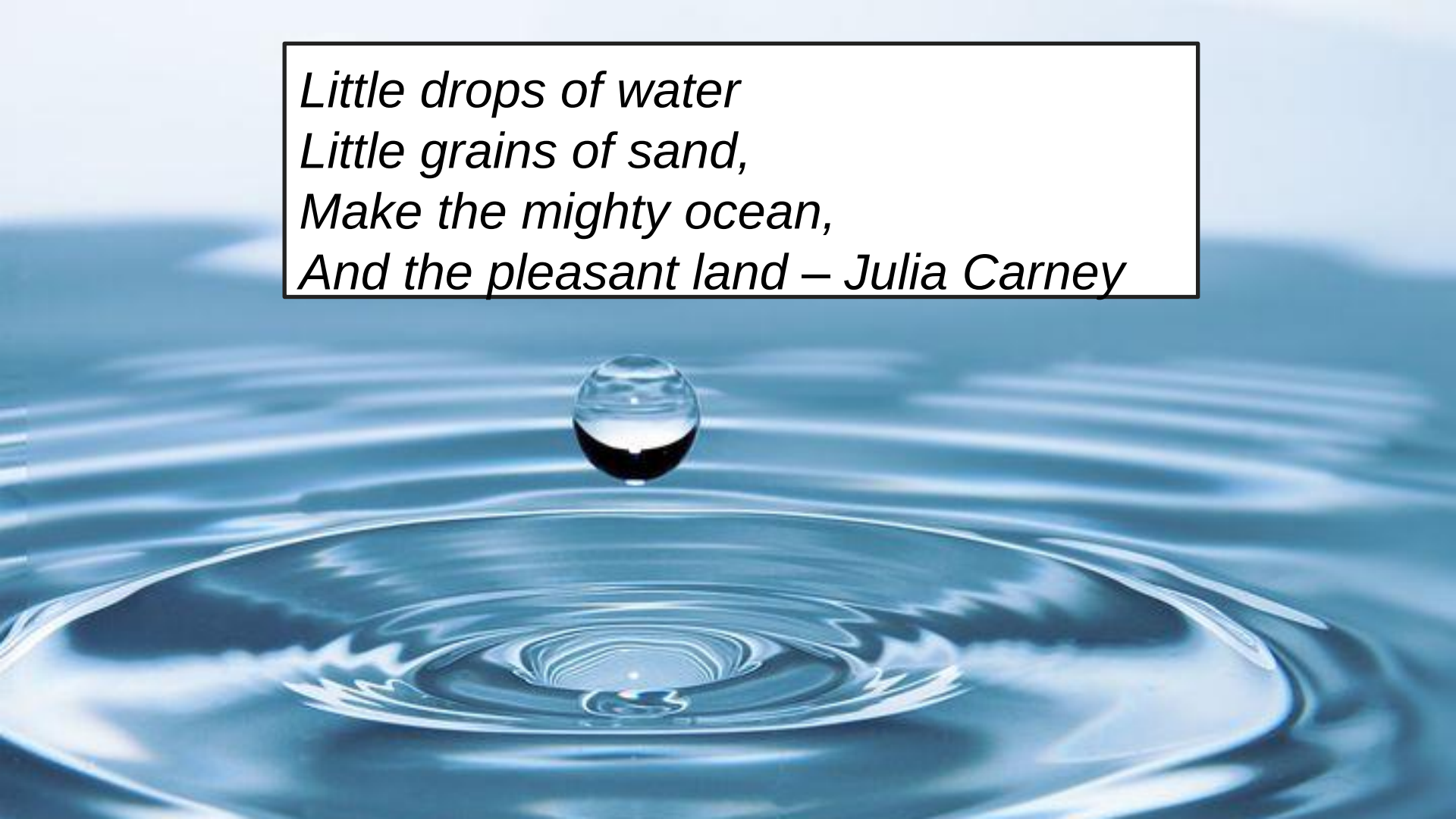
Need for Ensembles

Trees have low bias and high variance

Depth	Test	Training
5	0.527	0.527
50	0.818	0.877
100	0.882	0.995

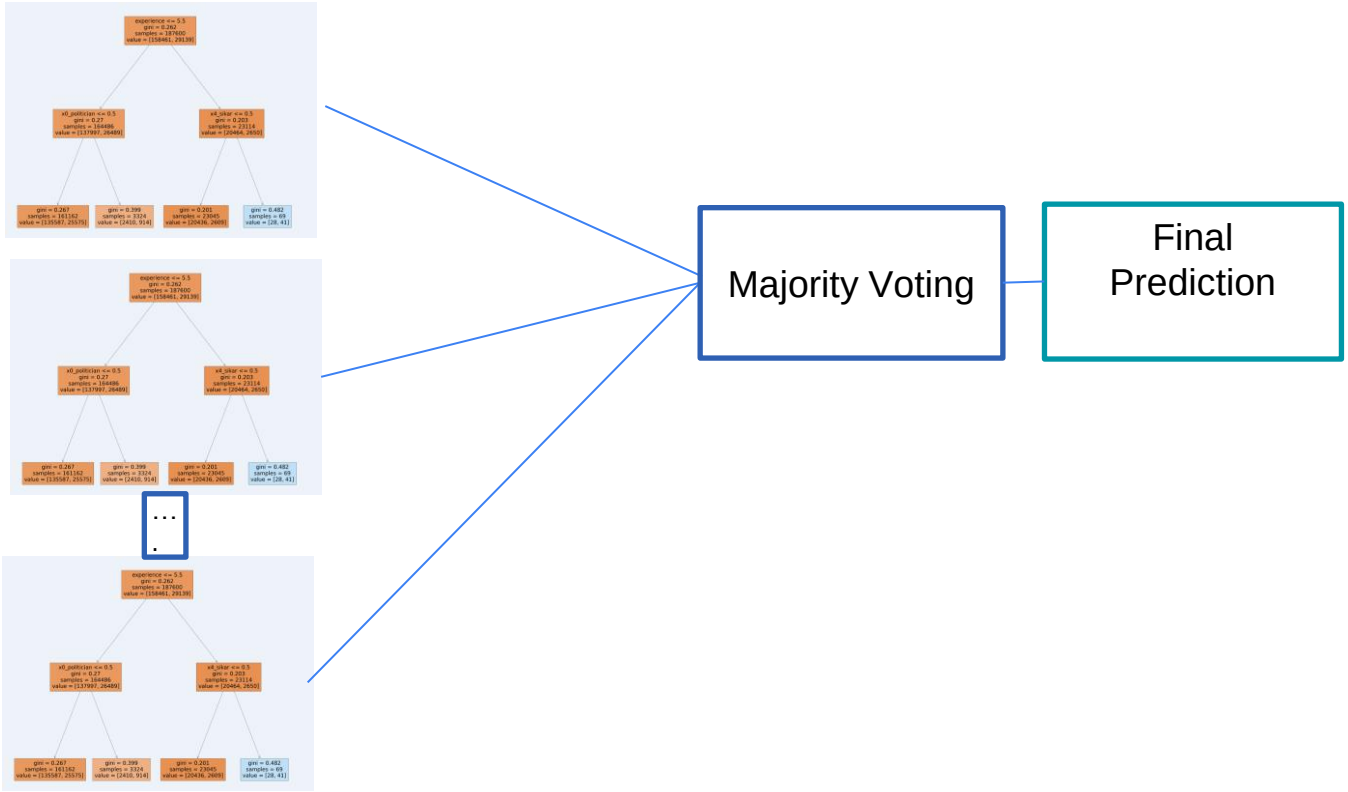
Over Fitting

Solution: Build many ‘simple trees’ and take the final vote of the those predictions



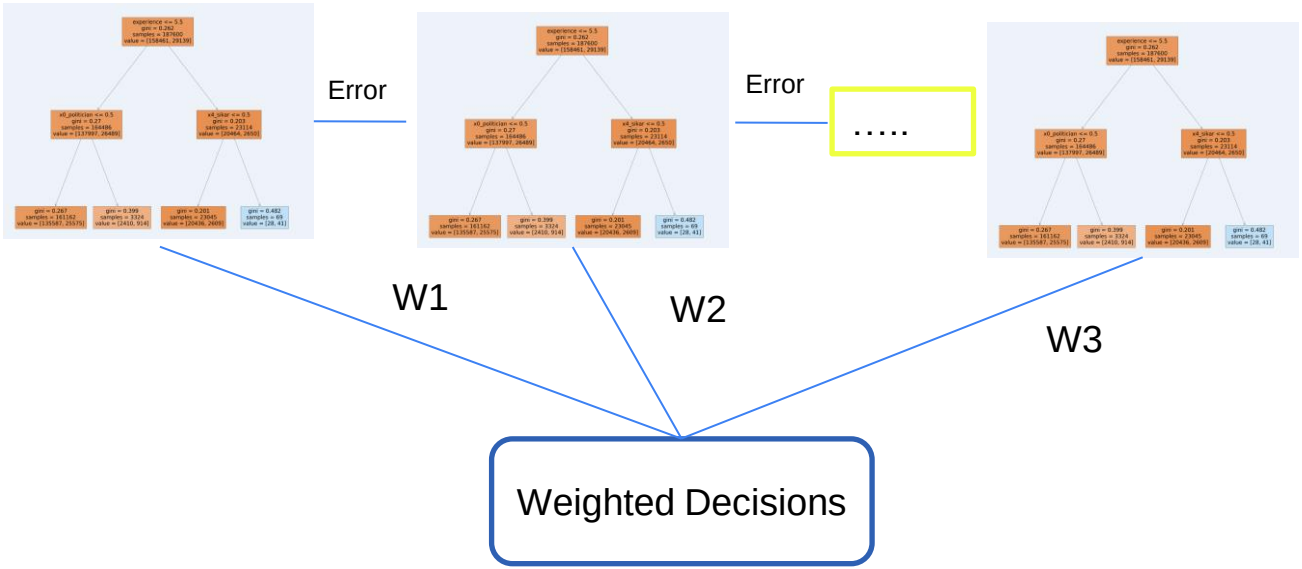
Little drops of water
Little grains of sand,
Make the mighty ocean,
And the pleasant land – Julia Carney

Bagging



https://www.kaggle.com/prashant111/bagging-vs-boosting

Boosting



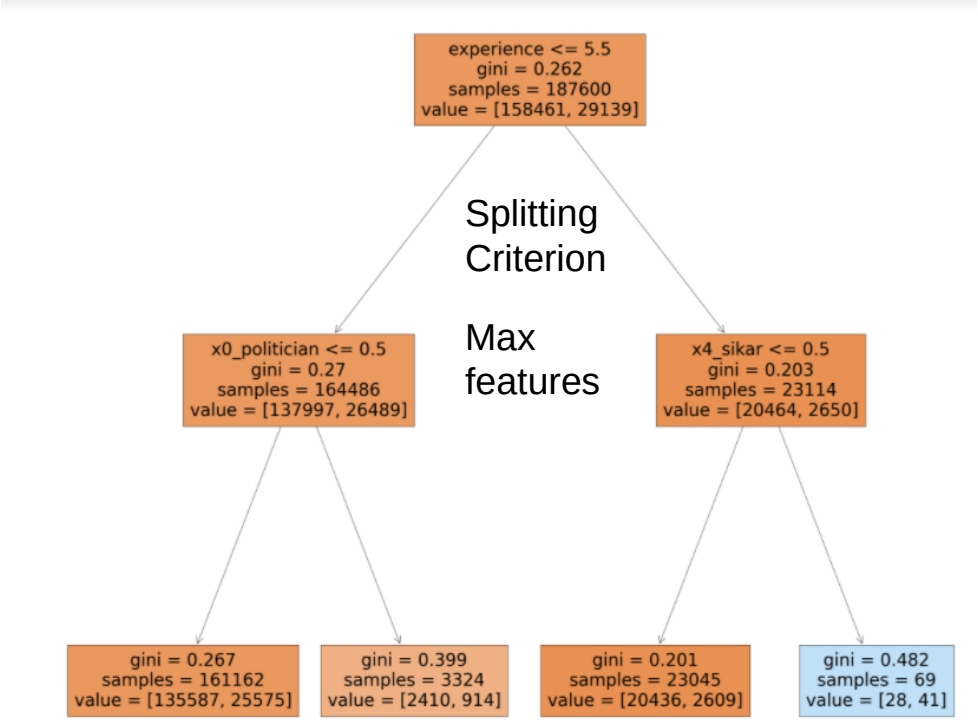
Reduce Variance = Bagging
Reduce Bias = Boosting

Bagging and Random Forests

- Random Forest is a special case of Bagging Algorithm
- Bagging is more generic, can take any estimator
- Bagging has subset of features in each training
- RF takes a subset of features in each split

Random Forest Sklearn (Key Parameters)

Handles higher dimensionality better



Splitting Criterion

Max features

BootStrap (Sample or No Sample)
max_samples
N_estimators

Min samples split

Min samples Leaf

Max leaf nodes

Over to Python

Tuning Random Forest

```
params = {
    'n_estimators': [40,50,100],
    'criterion': ["gini", "entropy"],
    'max_samples': [0.1,0.2,0.5,1],
    'max_features':[0.1,0.2,0.3]


}
```

Algorithm	Test	Training	CV
Bagging	0.942	0.982	0.938
RF	0.96	0.982	0.956

Over to Python

Summary

- Gini, Entropy, Variance
- Pruning
- Decision Trees in Python
- Ensembles
- Bagging & Boosting
- Bagging & Random Forest in scikit learn