COMP30810

Intro to Text Analytics

Dr. Binh Thanh Le

thanhbinh.le@ucd.ie

Insight Centre for Data Analytics

School of Computer Science

University College Dublin

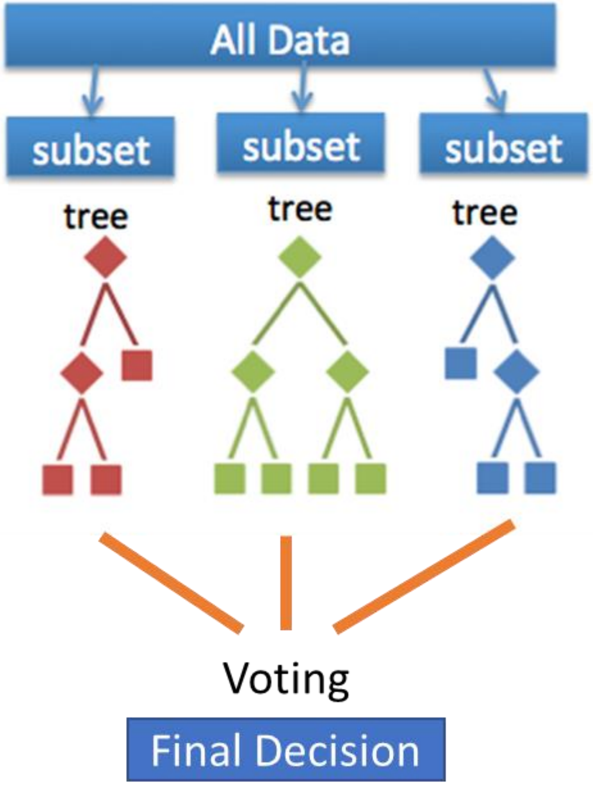
Today goals

- Understand Random Forest
- Understand how to apply RF in Text Analytics

What is Random Forest?

Random forest = learning ensemble consisting of a bagging of un-pruned decision tree learners with a randomized selection of features at each split.

- The term came from random decision forests that was first proposed by Tin Kam Ho of Bell Labs in 1995.
- The method combines Breiman's "bagging" idea and the random selection of features.



What is Decision Tree?

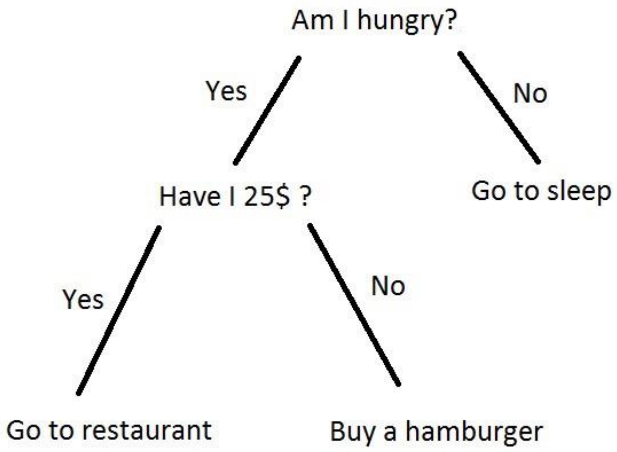
- Decision trees … one of most popular learning methods commonly used for data exploration
- A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).

- *A decision tree is drawn upside down with its root at the top*

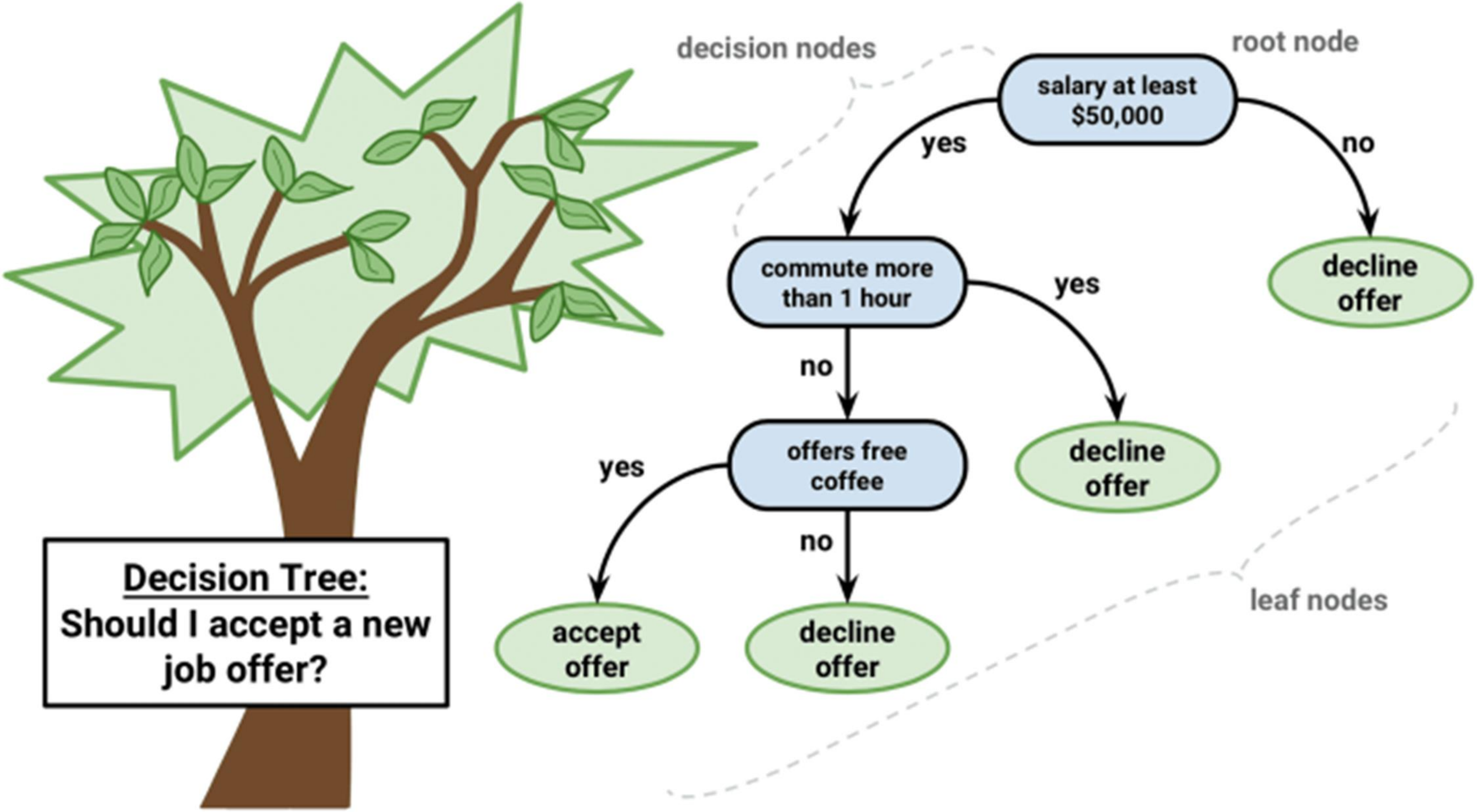
- ***Why Decision trees?***

Decision tress often mimic the human level thinking so its so simple to understand the data and make some good interpretations.

➔ Interpretability



Example of a tree

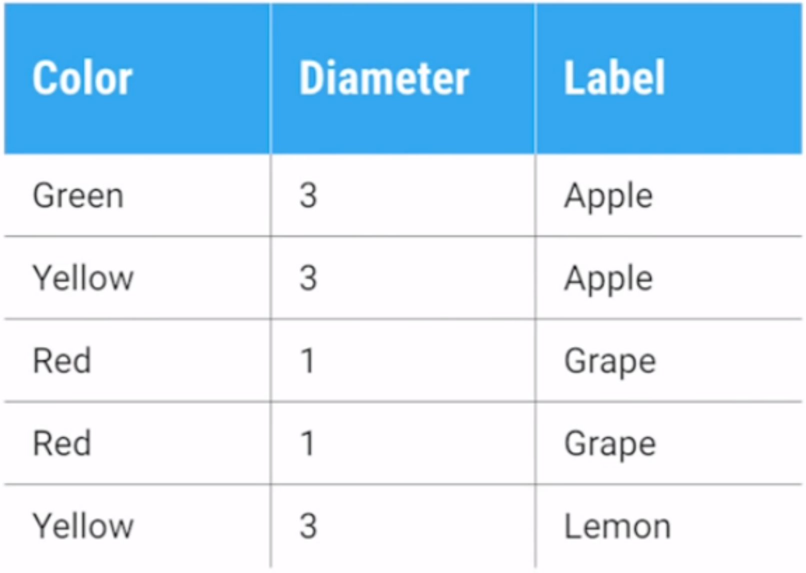


How to build the tree?

- There are couple of algorithms there to build a decision tree
  - ID3
  - C4.5
  - C5.0
  - CART

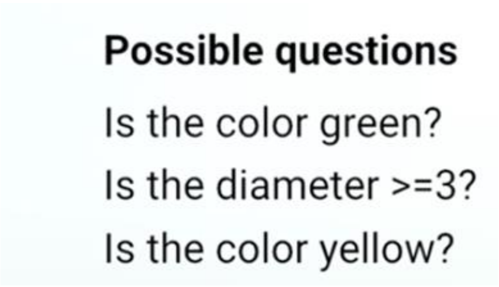
Classification And Regression Tree

Let’s take an example:



Color	Diameter	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

First question: What is the ROOT?



Possible questions

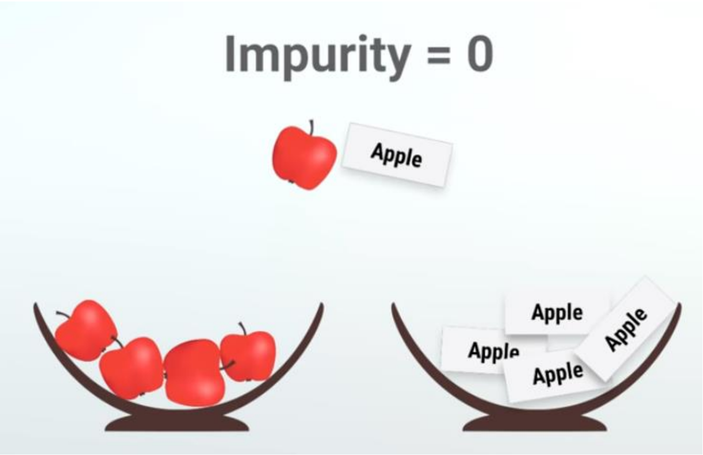
Is the color green?

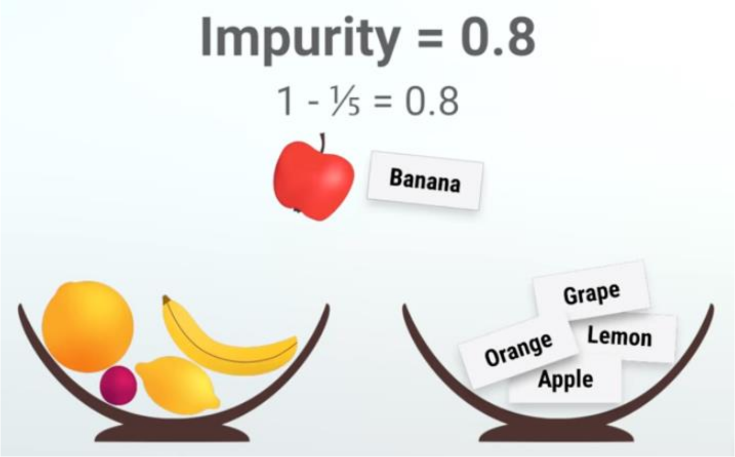
Is the diameter >=3?

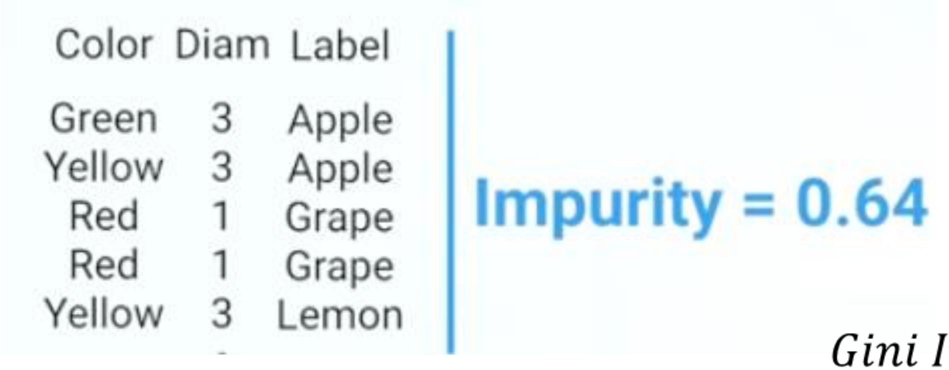
Is the color yellow?

…

Gini Impurity - *Gini Index*







Color	Diam	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Impurity = 0.64

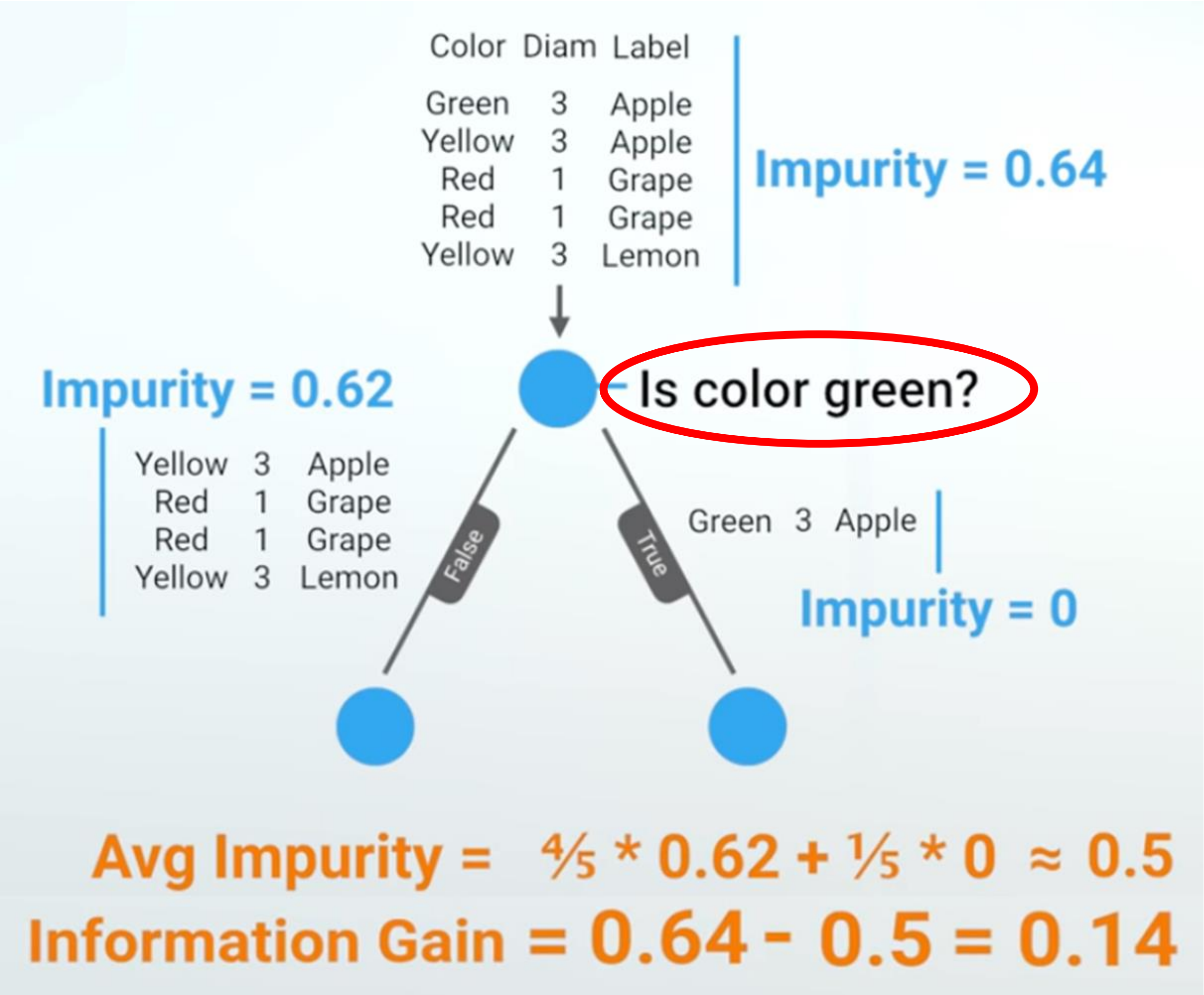
$$p(Apple) = 2/5$$
$$p(Grape) = 2/5$$
$$p(Lemon) = 1/5$$

$$GiniIndex = 1 - \sum_j p_j^2$$

$$Gini\ Impurity = 1 - \left[\left(\frac{2}{5}\right)^2 + \left(\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right]$$

= 0.64

Information Gain



Color	Diam	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Impurity = 0.64

Is color green?

Impurity = 0.62

Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

False

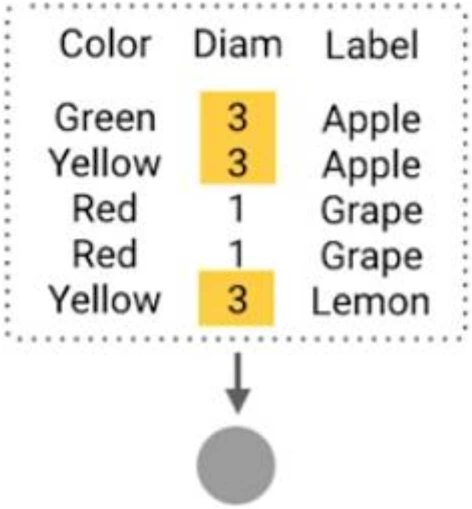
True

Green	3	Apple

Impurity = 0

Avg Impurity = ⅘ * 0.62 + ⅕ * 0 ≈ 0.5

Information Gain = 0.64 - 0.5 = 0.14

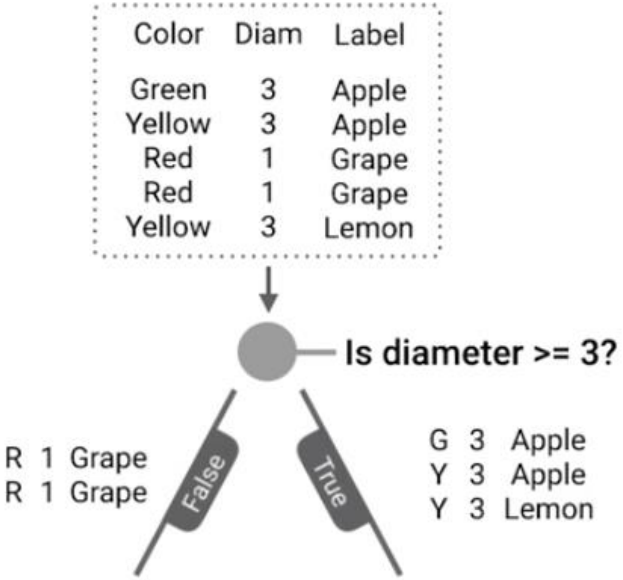


Color	Diam	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Information Gain

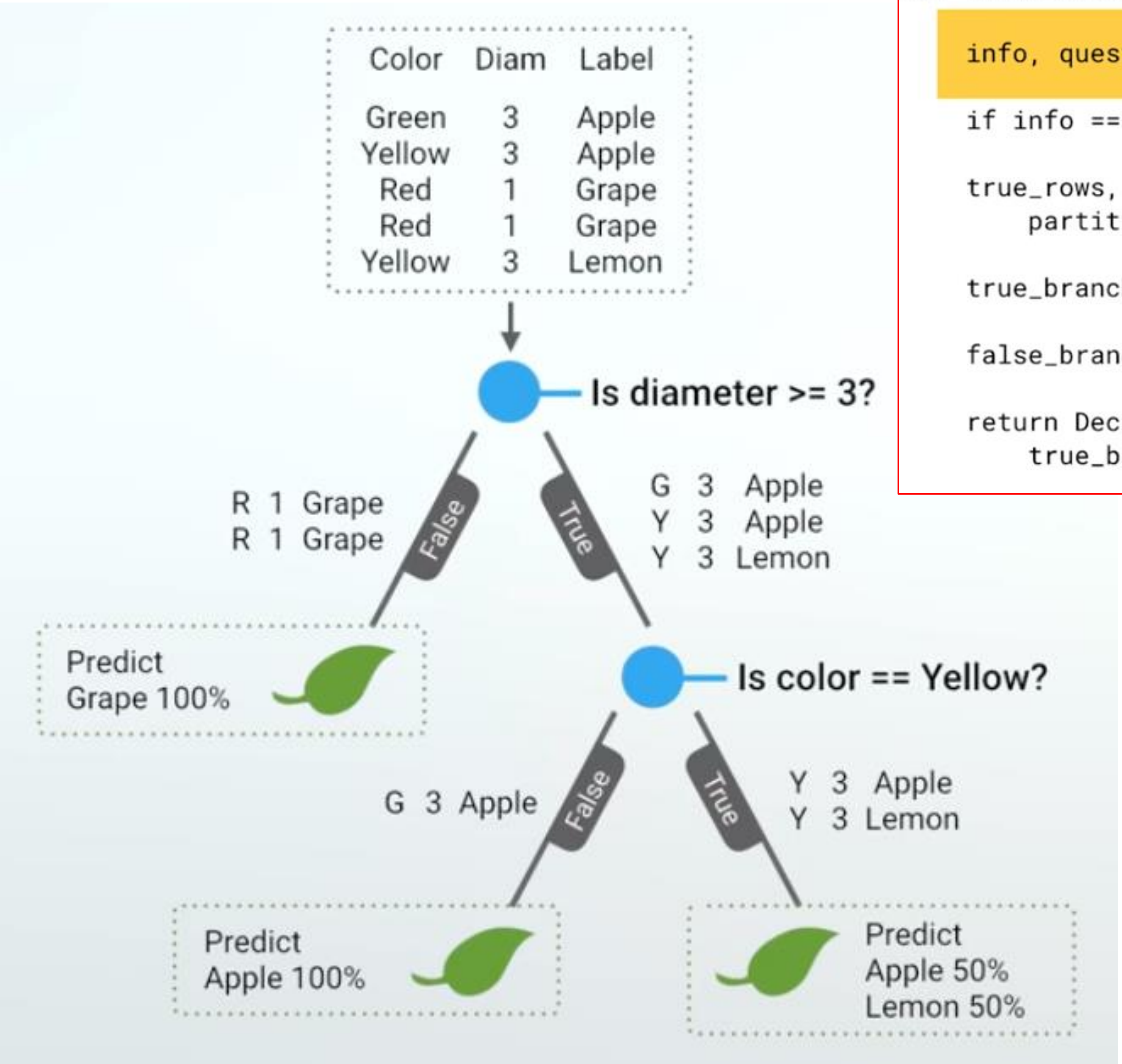
Question	**Gain**
Color == Green?	0.14
Diameter >= 3?	0.37
Color == Yellow?	0.17
Color == Red?	0.37
Diameter >=1?	0

This is the ROOT



Color	Diam	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

Repeat Previous Steps

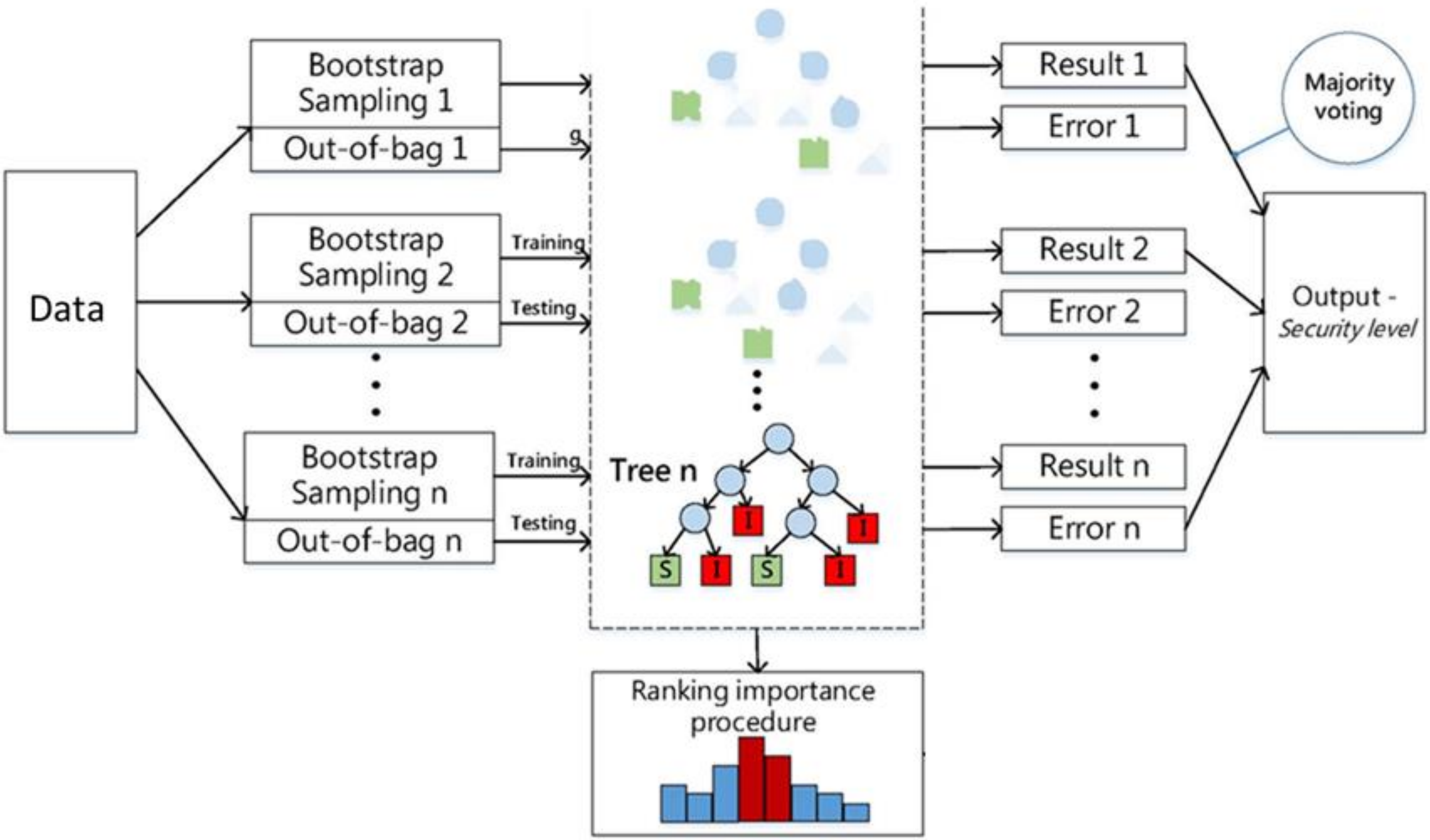


Color	Diam	Label
Green	3	Apple
Yellow	3	Apple
Red	1	Grape
Red	1	Grape
Yellow	3	Lemon

```
def build_tree(rows):

    info, question = find_best_split(rows)

    if info == 0: return Leaf(rows)

    true_rows, false_rows =
        partition(rows, question)

    true_branch = build_tree(true_rows)

    false_branch = build_tree(false_rows)

    return Decision_Node(question,
        true_branch, false_branch)
```

If the child node is “pure” (has instances from only one class) tag it as a leaf and return.

Random Forest

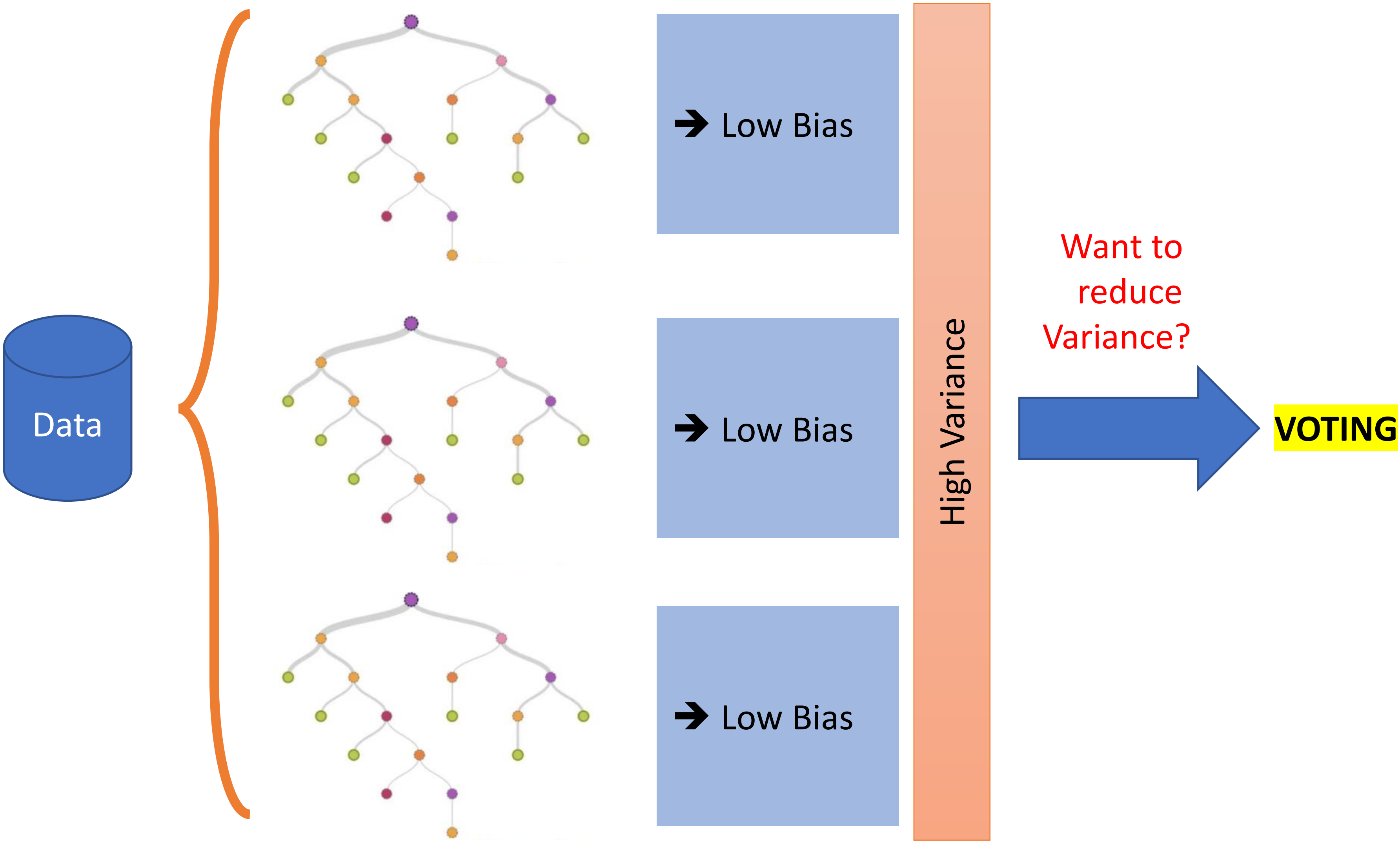


Random Forest vs Decision Tree

Random Forest	Decision Tree
- Classification + Regression	- Classification + Regression
- Require much of data for Bagging step	- Does not require much of data
	- Easy to interpret and make for straightforward visualizations
- Can provide the Feature Importance scores	
	- This is a greedy model, meaning it makes the most optimal decision at each step, but does not consider the global optimum.
- Can avoid the overfitting	- Decision trees are prone to overfitting, especially when a tree is particularly deep
- Many trees can make the algorithm to slow and ineffective for real-time predictions	

Why vote?

Decision Trees have usually ***low bias*** *because they maximally overfit to the training data.*



Data

→ Low Bias

→ Low Bias

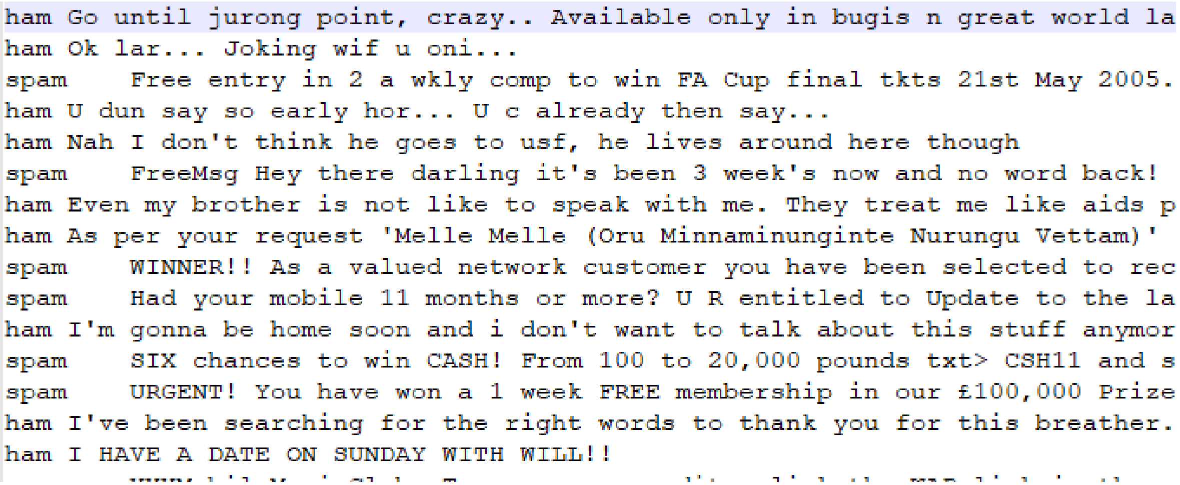
→ Low Bias

High Variance

Want to reduce Variance?

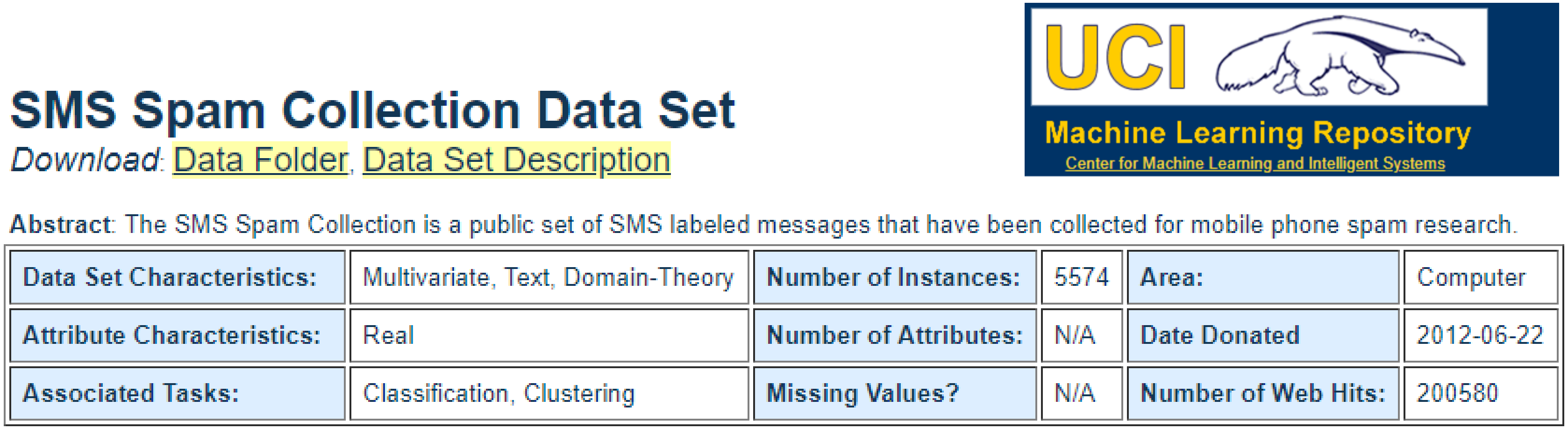
VOTING

Example for Text Analytics – Ham/Spam SMS



```
ham Go until jurong point, crazy.. Available only in bugis n great world la
ham Ok lar... Joking wif u oni...
spam    Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005.
ham U dun say so early hor... U c already then say...
ham Nah I don't think he goes to usf, he lives around here though
spam    FreeMsg Hey there darling it's been 3 week's now and no word back!
ham Even my brother is not like to speak with me. They treat me like aids p
ham As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)'
spam    WINNER!! As a valued network customer you have been selected to rec
spam    Had your mobile 11 months or more? U R entitled to Update to the la
ham I'm gonna be home soon and i don't want to talk about this stuff anymor
spam    SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and s
spam    URGENT! You have won a 1 week FREE membership in our £100,000 Prize
ham I've been searching for the right words to thank you for this breather.
ham I HAVE A DATE ON SUNDAY WITH WILL!!
```

Download at: https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection



UCI
Machine Learning Repository
Center for Machine Learning and Intelligent Systems

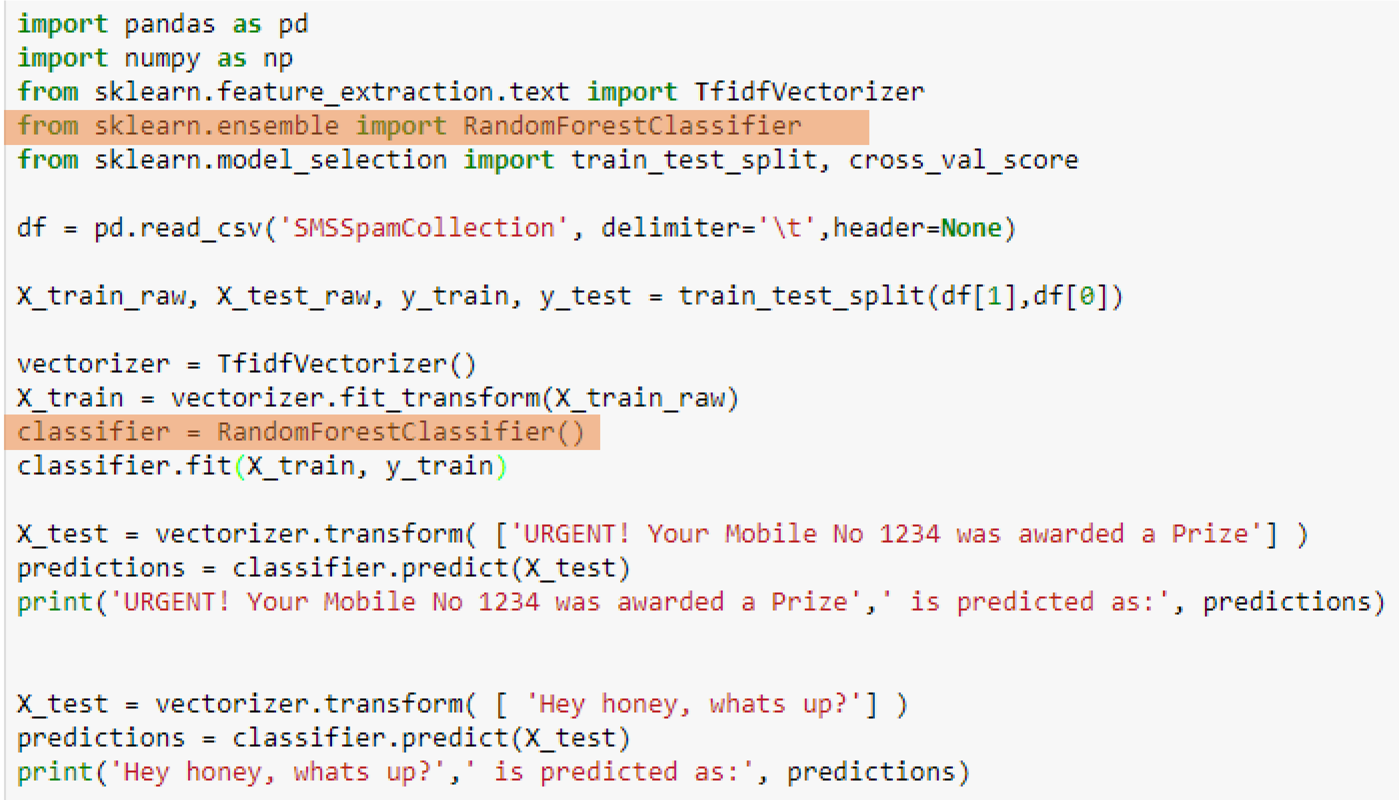
SMS Spam Collection Data Set

Download: Data Folder, Data Set Description

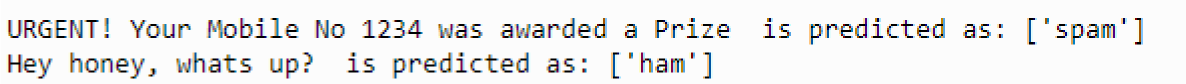
Abstract: The SMS Spam Collection is a public set of SMS labeled messages that have been collected for mobile phone spam research.

Data Set Characteristics:	Multivariate, Text, Domain-Theory	**Number of Instances:**	5574	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	N/A	**Date Donated**	2012-06-22
Associated Tasks:	Classification, Clustering	**Missing Values?**	N/A	**Number of Web Hits:**	200580

Example in Text Analysis



```python
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score

df = pd.read_csv('SMSSpamCollection', delimiter='\t',header=None)

X_train_raw, X_test_raw, y_train, y_test = train_test_split(df[1],df[0])

vectorizer = TfidfVectorizer()
X_train = vectorizer.fit_transform(X_train_raw)
classifier = RandomForestClassifier()
classifier.fit(X_train, y_train)

X_test = vectorizer.transform( ['URGENT! Your Mobile No 1234 was awarded a Prize'] )
predictions = classifier.predict(X_test)
print('URGENT! Your Mobile No 1234 was awarded a Prize',' is predicted as:', predictions)


X_test = vectorizer.transform( [ 'Hey honey, whats up?'] )
predictions = classifier.predict(X_test)
print('Hey honey, whats up?',' is predicted as:', predictions)
```



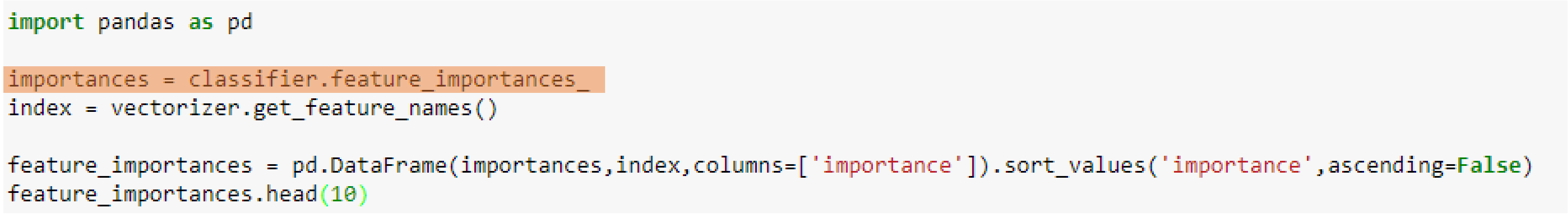
```
URGENT! Your Mobile No 1234 was awarded a Prize  is predicted as: ['spam']
Hey honey, whats up?  is predicted as: ['ham']
```

Feature Importance

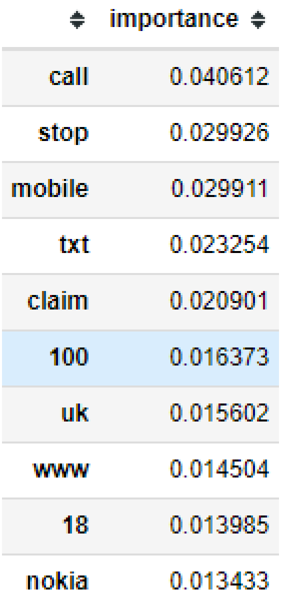
It is nice if we can see “How are important of token words?”

➔ Make an extra analysis on this

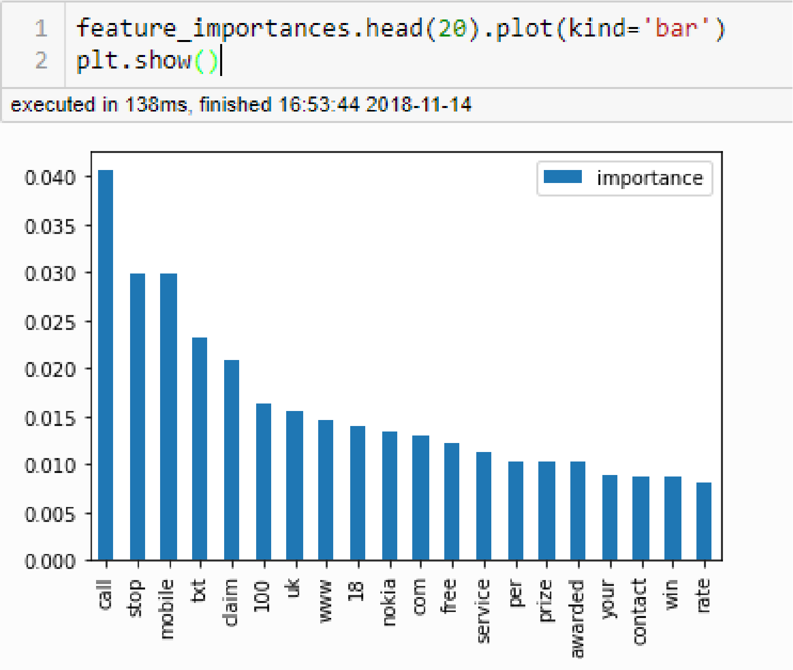
- Dictionary for corpus?
- Feature extraction/selection?



```
import pandas as pd

importances = classifier.feature_importances_
index = vectorizer.get_feature_names()

feature_importances = pd.DataFrame(importances,index,columns=['importance']).sort_values('importance',ascending=False)
feature_importances.head(10)
```



	importance
call	0.040612
stop	0.029926
mobile	0.029911
txt	0.023254
claim	0.020901
100	0.016373
uk	0.015602
www	0.014504
18	0.013985
nokia	0.013433



```
feature_importances.head(20).plot(kind='bar')
plt.show()
```

executed in 138ms, finished 16:53:44 2018-11-14