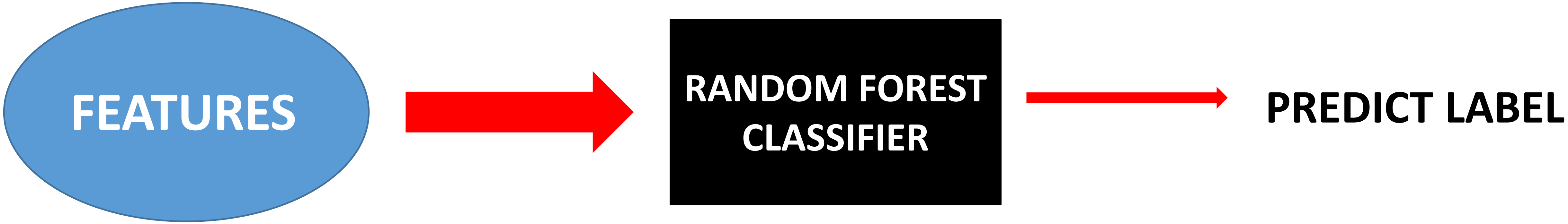
Random Forest Classifier

Damilola O.Said

Advanced Bioinformatics

20th of April, 2017

GOAL



FEATURES

RANDOM FOREST CLASSIFIER

PREDICT LABEL

DATA STRUCTURE

objects

	2° Structure	Free energy/nt	GC Content	Type
Sequence 1				Not miRNA
Sequence 2				Not miRNA
Sequence 3				miRNA
Sequence n				Not miRNA

DATA STRUCTURE

Features

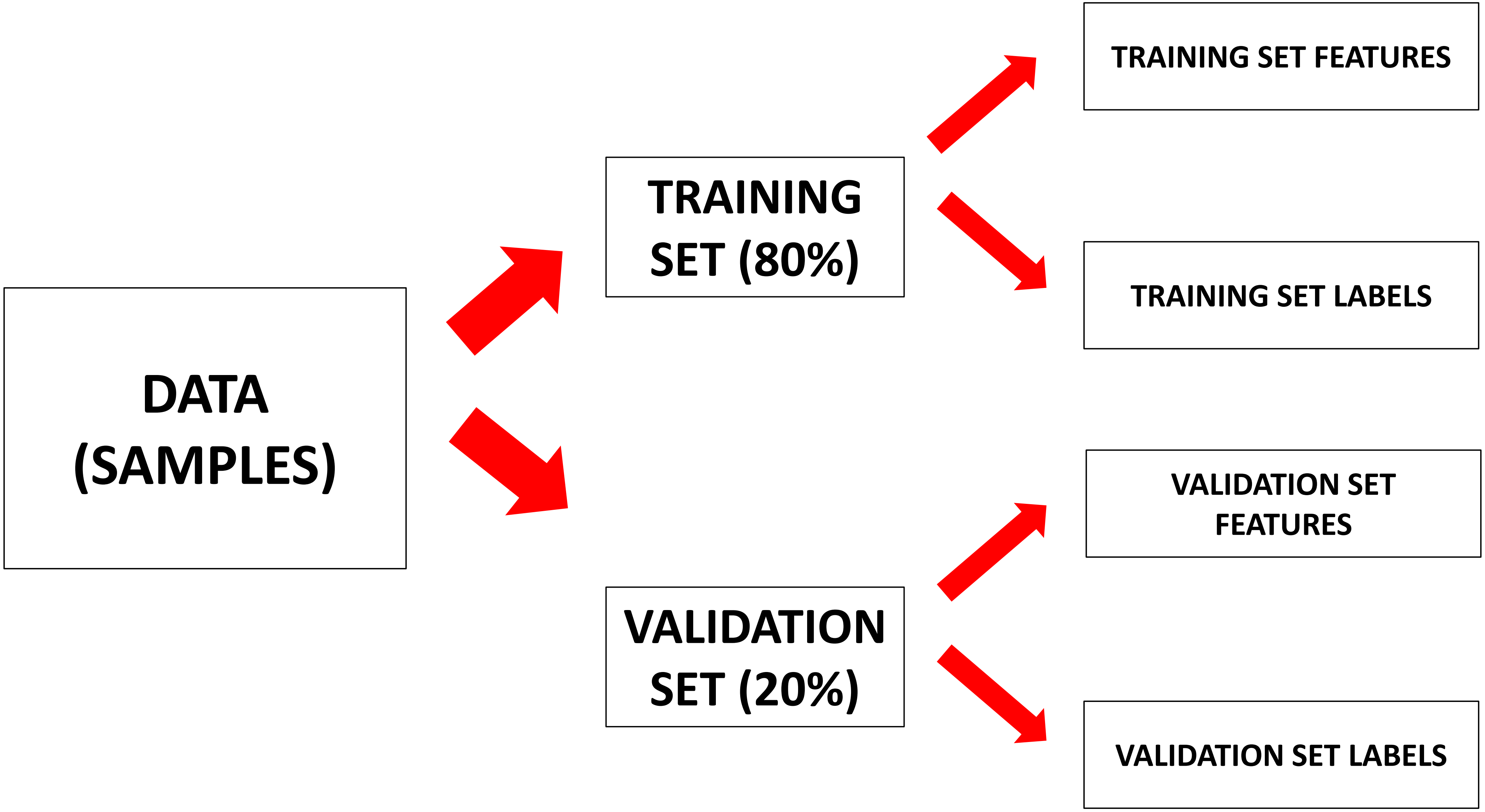
	2^{o} Structure	Free energy/nt	GC Content	Type
Sequence 1				Not miRNA
Sequence 2				Not miRNA
Sequence 3				miRNA
Sequence n				Not miRNA

DATA STRUCTURE

Targets/Labels

	2^{o} Structure	Free energy/nt	GC Content	Type
Sequence 1				Not miRNA
Sequence 2				Not miRNA
Sequence 3				miRNA
Sequence n				Not miRNA

SPLITTING THE DATA



DATA (SAMPLES)

TRAINING SET (80%)

VALIDATION SET (20%)

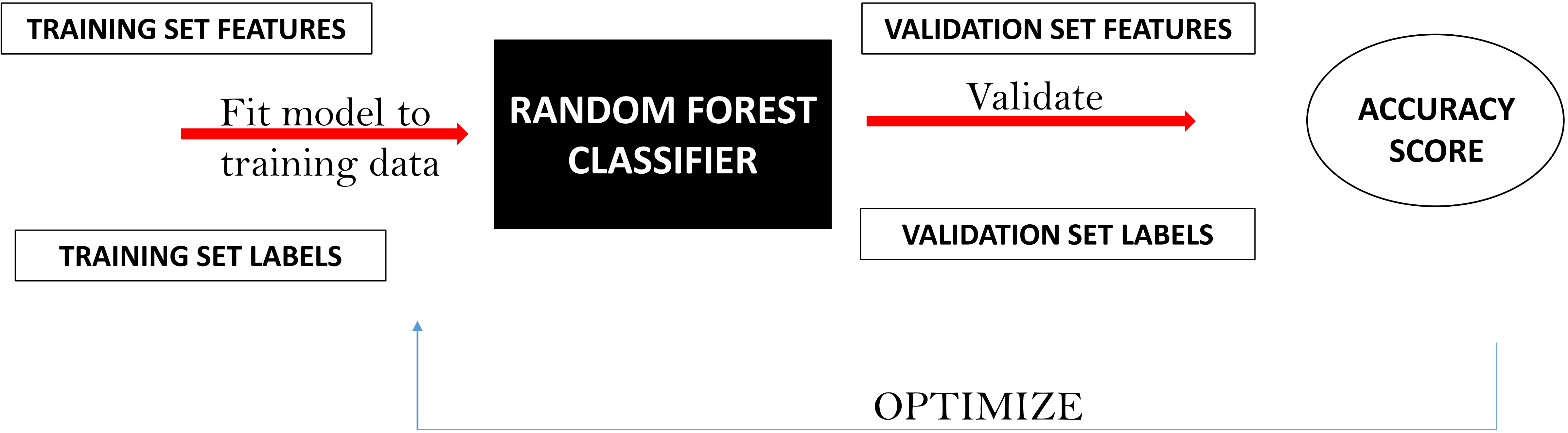
TRAINING SET FEATURES

TRAINING SET LABELS

VALIDATION SET FEATURES

VALIDATION SET LABELS

TRAINING THE CLASSIFIER



TRAINING SET FEATURES

TRAINING SET LABELS

Fit model to training data

RANDOM FOREST CLASSIFIER

VALIDATION SET FEATURES

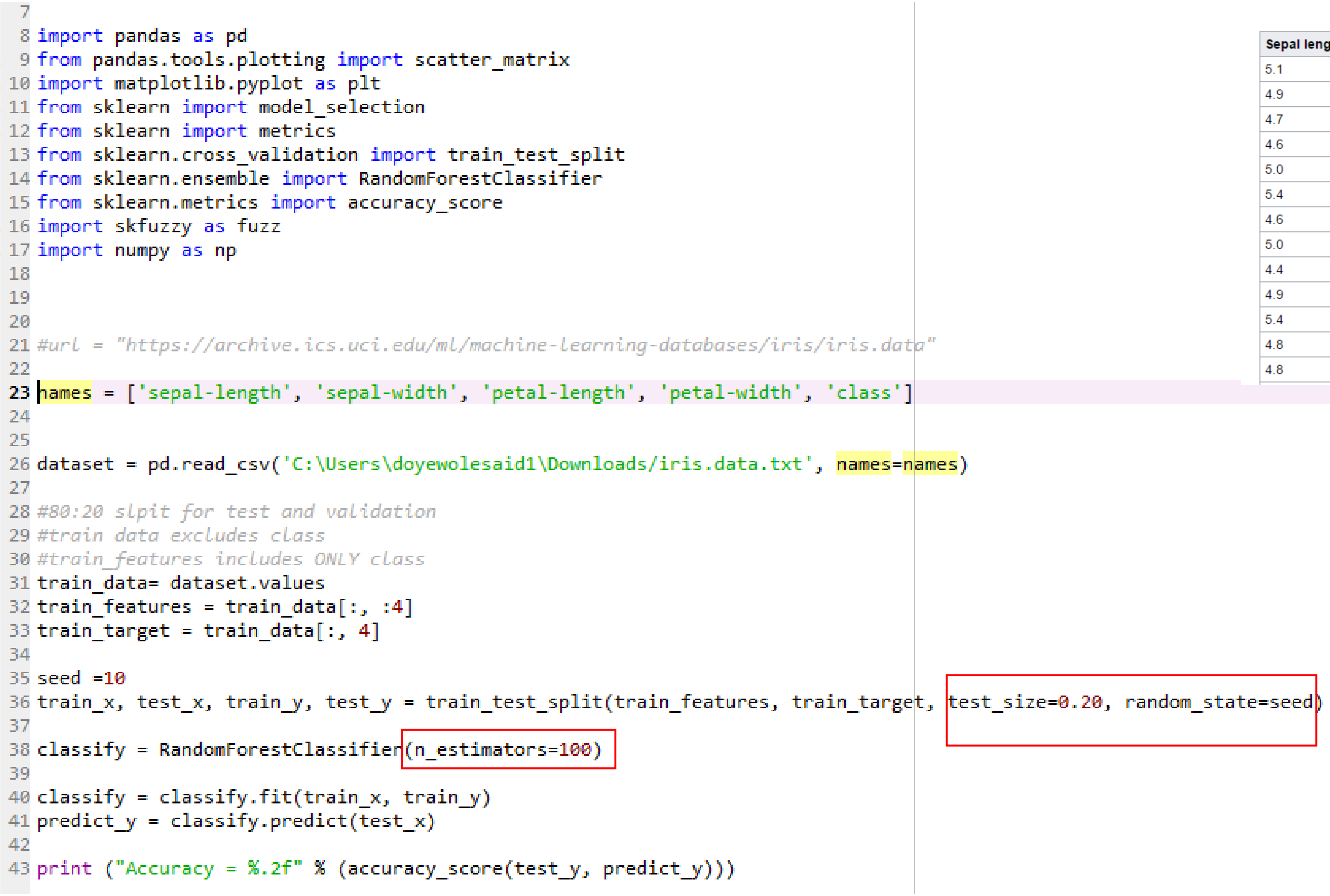
Validate

VALIDATION SET LABELS

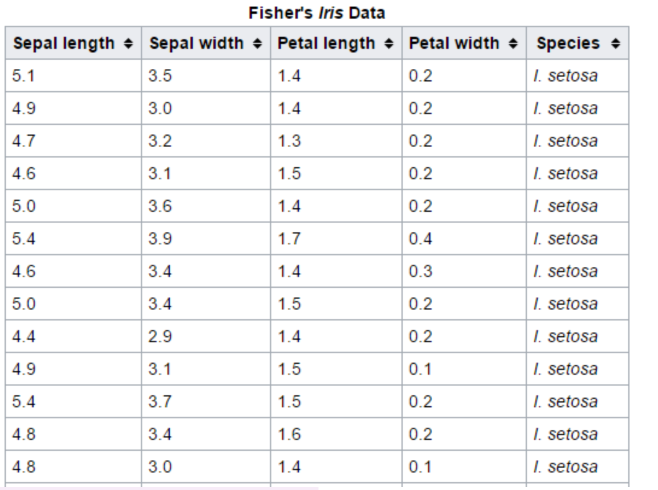
ACCURACY SCORE

OPTIMIZE

THE CODE: IRIS DATASET



```python

import pandas as pd
from pandas.tools.plotting import scatter_matrix
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn import metrics
from sklearn.cross_validation import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import skfuzzy as fuzz
import numpy as np



#url = "https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"

names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']


dataset = pd.read_csv('C:\Users\doyewolesaid1\Downloads/iris.data.txt', names=names)

#80:20 slpit for test and validation
#train data excludes class
#train_features includes ONLY class
train_data= dataset.values
train_features = train_data[:, :4]
train_target = train_data[:, 4]

seed =10
train_x, test_x, train_y, test_y = train_test_split(train_features, train_target, test_size=0.20, random_state=seed)

classify = RandomForestClassifier(n_estimators=100)

classify = classify.fit(train_x, train_y)
predict_y = classify.predict(test_x)

print ("Accuracy = %.2f" % (accuracy_score(test_y, predict_y)))
```



Fisher's *Iris* Data

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*
4.4	2.9	1.4	0.2	*I. setosa*
4.9	3.1	1.5	0.1	*I. setosa*
5.4	3.7	1.5	0.2	*I. setosa*
4.8	3.4	1.6	0.2	*I. setosa*
4.8	3.0	1.4	0.1	*I. setosa*

(150 samples, 4 features + label)

REFERENCES

1. Brownlee, J. http://machinelearningmastery.com/machine-learning-in-python-step-by-step/
2. Martin, D. http://nbviewer.jupyter.org/github/donnemartin/data-science-ipython-notebooks/blob/master/kaggle/titanic.ipynb