

Ask me Iris

Every day I hear about Machine Learning, but

What is Machine Learning?

How can I learn about it?

How can I use it?

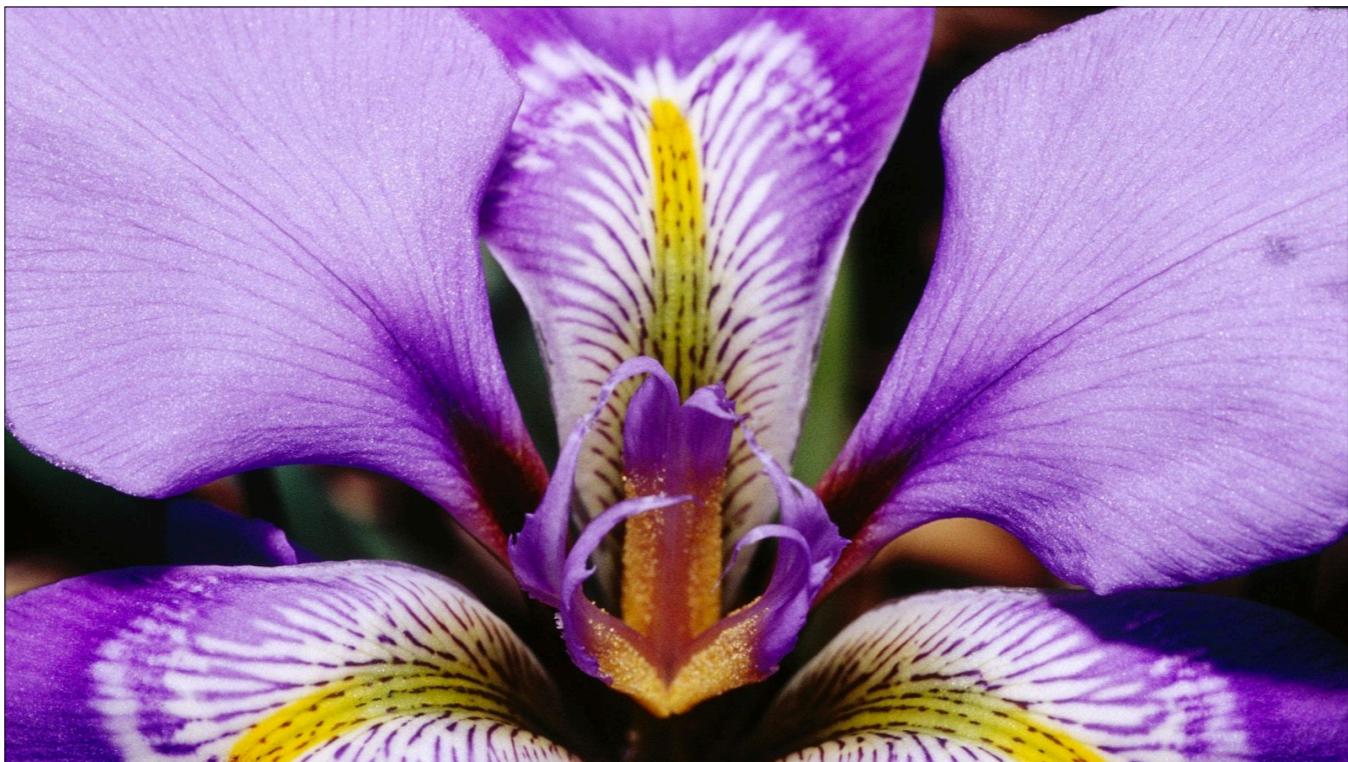
Machine Learning is a **process**, not an algorithm

It is a subset of tools & techniques in the field of Computer Science known as Artificial Intelligence



Classifications are a probabilistic guess about patterns in your data

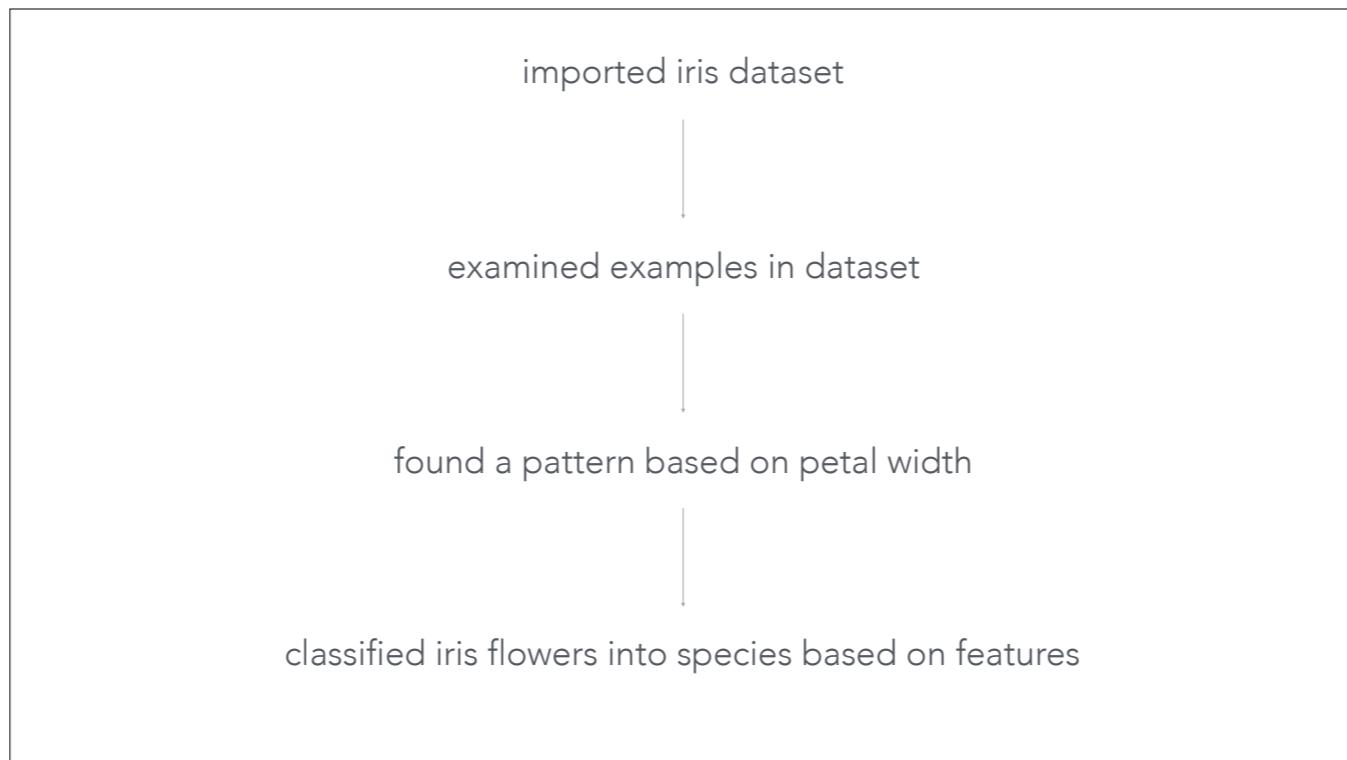
Can we create a user experience to demonstrate how algorithms learn to make decisions in the ML process?





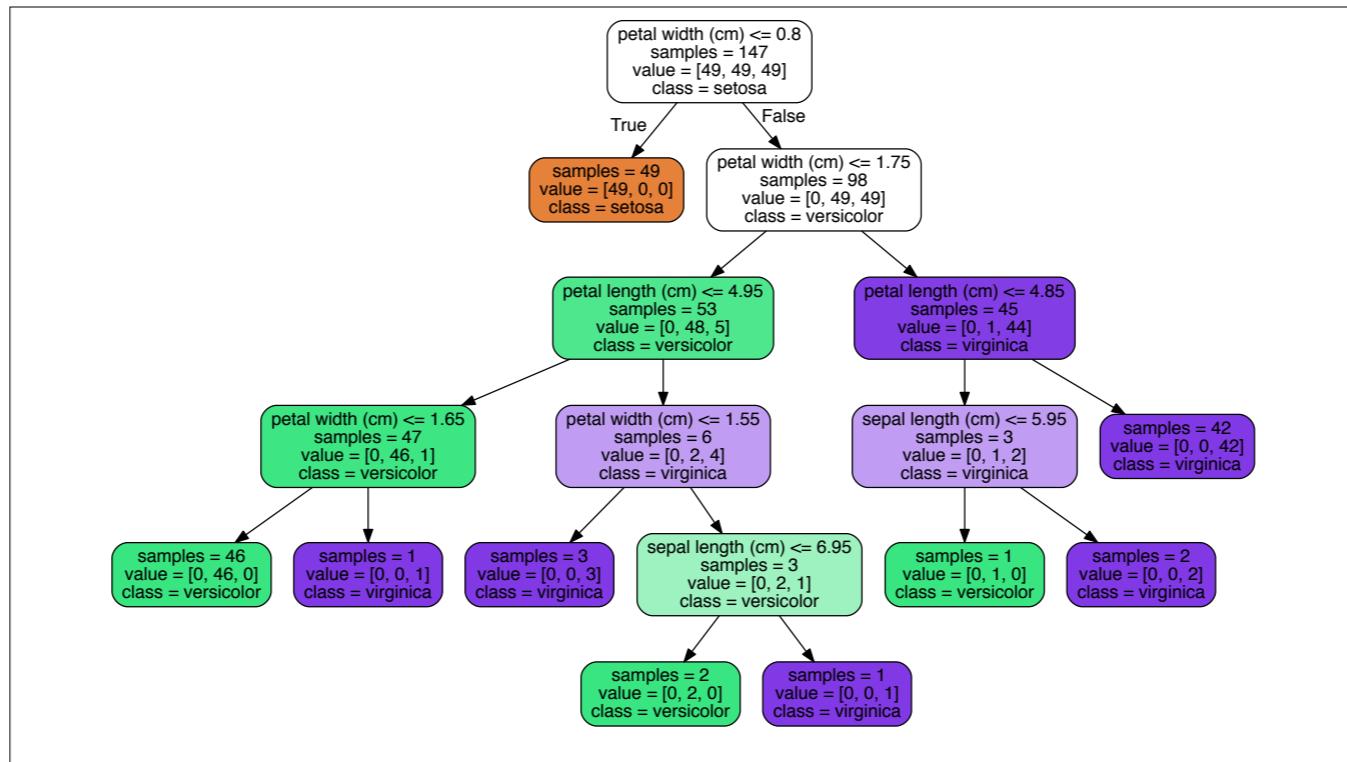
- 1) Import dataset
- 2) Train a classifier
- 3) Predict label for new flower
- 4) Visualize the Decision Tree

Demo



- 1) Import dataset
- 2) Train a classifier
- 3) Predict label for new flower
- 4) Visualize the Decision Tree

How did Iris know how to
classify our flower?



Challenges

```
print np.random.rand(iris.data)

Traceback (most recent call last):
  File "/Users/matthewkrey/Development/cs_59a/week6/ask me iris.py", line 86, in <module>
    print np.random.rand(iris.data)
  File "mtrand.pyx", line 1623, in mtrand.RandomState.rand (numpy/random/mtrand/mtrand.c:18928)
  File "mtrand.pyx", line 1143, in mtrand.RandomState.random_sample (numpy/random/mtrand/mtrand.c:14895)
  File "mtrand.pyx", line 163, in mtrand.cont0_array (numpy/random/mtrand/mtrand.c:2383)
TypeError: only length-1 arrays can be converted to Python scalars

Process finished with exit code 1
```

```
print np.random.choice(iris.data)

Traceback (most recent call last):
  File "/Users/matthewkrey/Development/cs_59a/week6/ask me iris.py", line 86, in <module>
    print np.random.choice(iris.data)
  File "mtrand.pyx", line 1393, in mtrand.RandomState.choice (numpy/random/mtrand/mtrand.c:16477)
ValueError: a must be 1-dimensional

Process finished with exit code 1
```

```
### CHALLENGE ###
shape = np.shape(iris.data)
shuffle = np.random.shuffle(iris.data)
X_train = iris.data[:200, 1:5]
selected = iris.data[np.random.randint(iris.data.shape[0], size=1)]
print selected
```

Future Improvements

(1) Messy Data - introduce Data Cleaning

(2) Functions

(3) Refactor, Refactor, Refactor

Appendix

(1) Human-AI Interaction

(2) Accessibility of ML Tools & Techniques

Free of charge

High quality education & documentation

Minimal code

Minimal infrastructure

- Free of charge (YouTube + Python + GitHub + PyCharm Student Edition + LPTHW)
- High quality education & documentation (Josh Gordon - Google; scikitlearn.org)
- Minimal code (8 line Classifier; no OOP, performance optimizations, even functions!)
 - The majority of the lines of code in this project, [insert % of total lines], was printing strings to guide UX
- Minimal infrastructure (Mid 2014 MacBook Pro; no data center, no data science team)

Acknowledgments

- [Josh Gordon - Machine Learning Recipes](#)
- [scikitlearn](#)
- Artificial Intelligence - What Everyone Needs to Know, Jerry Kaplan
- [Iris flower data set](#)
- [scikitlearn - Dataset loading utilities](#)
- [Learn Python the Hard Way - Zed Shaw](#)
- https://github.com/MatthewKrey/cs59a_final_project

Include list of library dependencies

UX Design

The screenshot shows a code editor window with two tabs: "ask_me_iris.py" and "project_notes.txt". The "ask_me_iris.py" tab contains the following Python code:

```
1 # Import Dependent Libraries & Iris Data Set
2
3 import sklearn
4 import numpy as np
5 import pandas as pd
6 import pydotplus
7 import time
8 from sklearn.datasets import load_iris
9 from sklearn import tree
10 from numpy import random, arange
11 from random import randrange
12 from sklearn.externals.six import StringIO
13
14 # Test data set
15
16 iris = load_iris()
17 # print iris.feature_names
18 # print iris.target_names
19
20 # Split Training & Test Data
21
22 test_idx = [0,50,100]
23
24 # training data
25 train_target = np.delete(iris.target, test_idx)
26 train_data = np.delete(iris.data, test_idx, axis=0)
27
28 # testing data
29 test_target = iris.target[test_idx]
30 test_data = iris.data[test_idx]
31
32 # Train Decision Tree Classifier on training data
33
34 clf = tree.DecisionTreeClassifier()
35 clf.fit(train_data, train_target)
```