

ML Club Meeting #3

Hands-on with Scikit-Learn

sidrrsh@gmail.com

ronithganijunta@gmail.com

Happy Halloween!

Candy will be given out today based on participation

Quick Review

To refresh your ML knowledge

Review of ML Terms

-
- Features - input variable (x)
 - Labels - the thing we're predicting (y)
 - Example - particular instance of data (unlabeled vs labeled ex.)
 - Model - relationship between features and labels
 - Training v. Inference

Basics of ML Learning

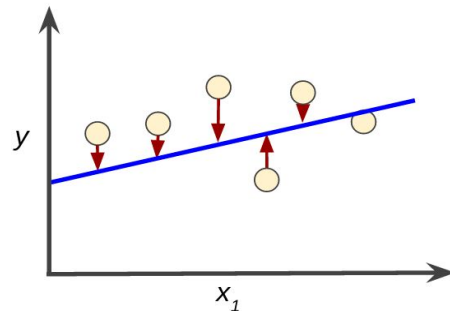
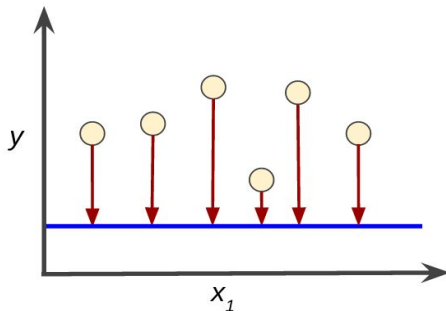
-
- A learning problem considers a set of n samples of data
 - predict properties of unknown data.
 - If each sample is more than a single number or is a multi-dimensional entry (multivariable)
 - It is said to have several attributes or features.
 - Supervised and Unsupervised Learning

Regression v. Classification

-
- Regression
 - Predicts continuous values
 - Quantitative
 - Classification
 - predict discrete values
 - Qualitative
 - Today we will be primarily dealing with
Classification

Loss functions

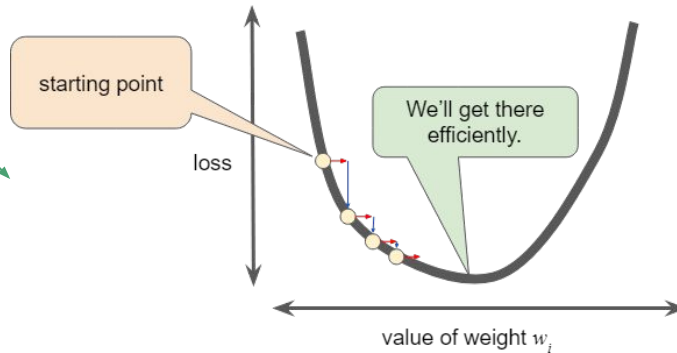
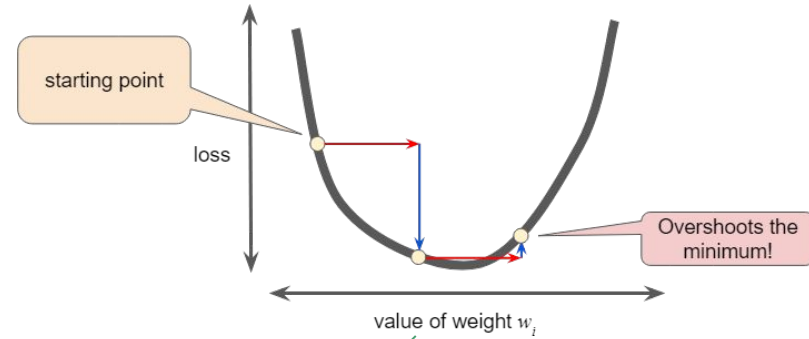
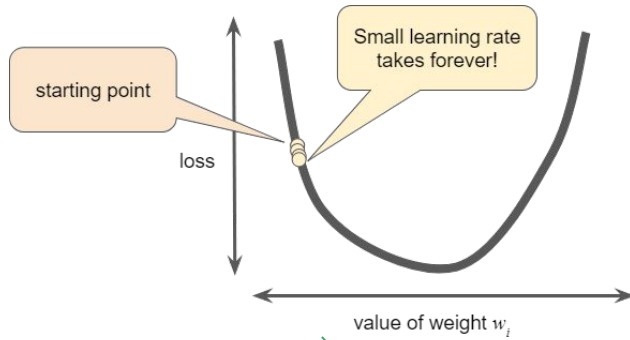
- How “bad” the model’s prediction was on a single example
- Which model below has higher loss?



Gradient Descent and Learning Rate

-
- Goal of an ML regression model is to minimize loss
 - Gradient descent is how we get to the loss minimization on the loss v. weight graph
 - Learning rate - adjustable hyperparameter (knob) to set gradient descent
 - Remember ... a learning rate shouldn't be too large or too small

“Correcting the Learning Rate”



Scikit-Learn Tutorial

Introduction to ML datasets and libraries

What is Scikit-Learn

- The first ML library that we will officially be interacting with
 - Libraries are collections of easy-to-use frameworks for ML
- A tool for data analysis and mining
- It is built upon two open-source python libraries - NumPy and SciPy

NumPy and SciPy

- NumPy - fundamental package for large, multi-dimensional arrays and matrices
 - Includes a large collection of high-level mathematical functions
- SciPy - Python library used for scientific computing and technical computing
 - Has modules for optimization, linear algebra, integration, interpolation, special functions
- We will heavily use both of these libraries across the year

Data sets

- We commonly test one data set against another data set to understand its properties
- Training set v. Testing set
 - Training set - we learn some properties
 - Testing set - we test the learned properties
- The Scikit package comes with a few standard data sets for both classification and regression

Installing Scikit-Learn

Installing ML packages and conducting a simple programming exercise

How to install scikit-learn pt. 1 (Mac)

- 1. Please search on google for Anaconda Download for Mac and download it from this link: <https://www.anaconda.com/download/>
- 2. Follow the instructions of the package installer and do not install Microsoft VS code
- 3. Open terminal and input the following command to update your pip install
- `pip install --upgrade pip`
- After your pip updates or is installed, now input the following line of input:
- `python -m pip install --user numpy scipy matplotlib ipython jupyter pandas sympy nose`

How to install scikit-learn pt. 2 (Mac)

- Now input this line of input into your terminal
 - `“pip install -U scikit-learn”` - you do not need the quotes:)
 - After the system runs this command, you should get a final output like this



```
Successfully installed scikit-learn-0.20.0
```

- If for some reason, that doesn't work, you can try `“conda install scikit-learn”`
 - Please see me if you are having issues

How to install scikit-learn (Windows)

- Please download WinPython from this link: <https://winpython.github.io/>
- Install Sublime Text as well: <https://www.sublimetext.com/>
- As it takes a lot of time to download WinPython and other softwares for Windows, please join someone who has a MacOS device
- Let's get into our first programming problem with ML!

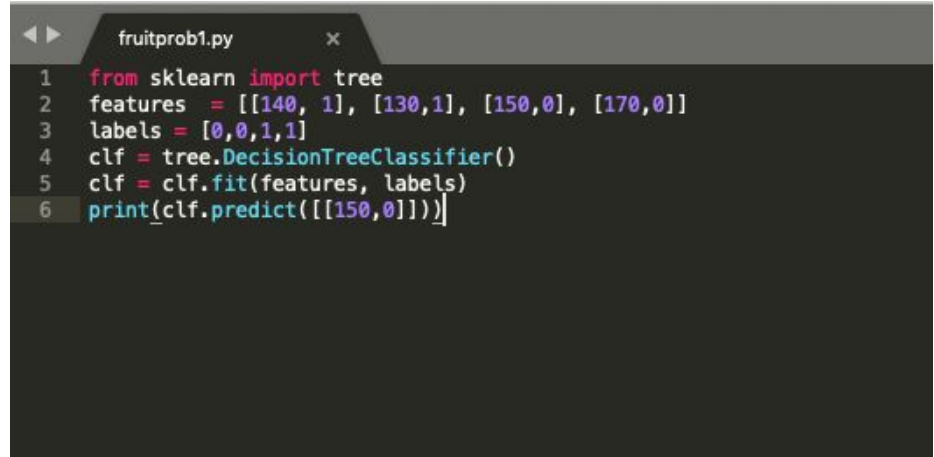
Our first ML programming problem



Fruit Program exercise

- Let's run and use the fruit example given in the exercise
- 1. Install Sublime Text as your text editor: <https://www.sublimetext.com/3>
- 2. Paste the following code (word-for-word) into a new Python script in Sublime Text and name the file as “fruitprob1.py”:

- You **MUST** save it to desktop



```
1 from sklearn import tree
2 features = [[140, 1], [130,1], [150,0], [170,0]]
3 labels = [0,0,1,1]
4 clf = tree.DecisionTreeClassifier()
5 clf = clf.fit(features, labels)
6 print(clf.predict([[150,0]]))
```

How to run the file

- For Mac users, return to your terminal and input “cd desktop” in the terminal
- Then input “python fruitprob1.py”
 - If it successfully works and doesn't give you a syntax error, you should an output like this:

```
Siddharths-MacBook-Pro:desktop siddharthsharma$ python fruitprob1.py  
[1]  
Siddharths-MacBook-Pro:desktop siddharthsharma$
```
 - This shows that our classifier classified the fruit successfully as an orange [1]

Implementing “Ghost v. Not Ghost”

- Similar to the example in the video of using a classifier to write a program that would predict if a fruit was an apple or orange
- We can write a program in python to predict if an object is a ghost or not based on a training set of data and then a testing set
- The parameters are:
 - if the object is white, it has a value of “1” and is probably a ghost
 - If the object is transparent, its weight is “0” grams, and is probably a ghost
- Set up your training data and testing data with values for color and weight using inspiration from our fruit problem

Sources

- Google crash course for ML
- Scikit-learn installation instructions
- Stack overflow
- Wikipedia