Intro to Python - Day 4

joseph.rejive

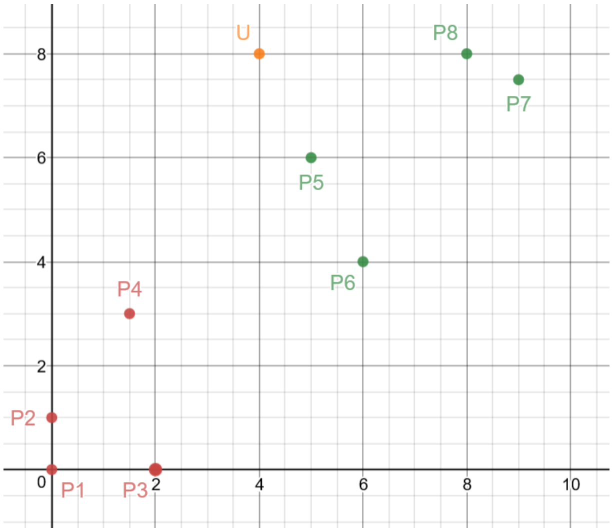
November 2018

1 Machine Learning

Machine Learning is a subset of Artificial Intelligence where models learn patterns in data. There are countless applications to this, which include detecting cancer from images and human speech recognition (like Siri or Alexa). In this lecture, we'll create a model which can predict whether a tumor is malignant or benign based off a breast cancer dataset.

2 K Nearest Neighbors

The K Nearest Neighbors algorithm (KNN) is a simple but powerful classification algorithm. When the algorithm tries to classify a given point U, it finds the k nearest points (where k is an integer) and classifies U with the most common group. Lets take a look at an example: [1]



Let's say our algorithm is classifying the orange point (lets call it point U) at (4,8). This point can either be part of the green group or the red group. Now we must select a value for k. For this example, we'll say k equals 3. The algorithm will then compute the distances from all the points *P1, P2, P3, ... P8* to point U. Below is a table of the distances:

P1	P2	P3	P4	P5	P6	P7	P8
8.94	8.06	8.25	5.59	2.24	4.47	5.02	4.0

[1] Graph generated using Desmos Graphing Calculator

The algorithm will then select the nearest 3 points, which are *P5, P8* and *P6.* Since all 3 points belong to group green, point U will be classified as green.

Now say, for example, we selected $k = 5$. The nearest 5 points would be *P5, P8, P6, P7* and *P4.* Even though *P4* belongs to group red, there are more points in group green. Thus, point U would still be classified as green.

2.1 Implementing KNN in Python

For the first example, we'll be creating creating a model on some toy data: a group of points in 2D space. We'll be using 2 libraries: numpy and sklearn. Numpy is the Python scientific computing package. It supports many matrix related operations. Sklearn is a library that has many built in programs meant for machine learning. It supports many algorithms and we'll be using it for its KNN class.

First let's install the required libraries. On Windows, open command prompt and type the following commands:

```
py -m pip install sklearn
py -m pip install numpy
```

On Mac, the commands are:

```
pip3 install sklearn
pip3 install numpy
```

Now, we'll import our libraries and generate our data.

```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier

features = [[0,0], [0,1], [2,0], [1.5,3], [6,4], [5,6], [8,8], [9,7.5]]
labels = [0,0,0,0,1,1,1,1]
```

In machine learning, features are inputs while labels are outputs. In this example, we're given the (x,y) coordinates of each point, which are our features. We want to predict whether the point belongs to group 0 or group 1 (the group that the point belongs to is the label).

```
knn = KNeighborsClassifier(n_neighbors = 3)

features = np.array(features)
labels = np.array(labels)

knn.fit(features, labels)
```

In this code snippet, we use sklearn's built in KNeighborsClassifier and we set the number of neighbors (k) to three. In the next two lines, we convert the features and labels lists to a numpy array. We do this because sklearn relies on numpy arrays to perform matrix operations. In the final line, we train the model using our features and our labels.

```
my_points = [[1,1], [6,6], [4,2]]
print(knn.predict(my_points))
```

Now that we trained our model, we can test various points using the "predict" function. The "predict" function accepts a list of points and returns an array of the labels (in other words, what group each point belongs to). Now let's use a more interesting dataset.

2.2 Classifying Breast Cancer

The files “features.txt” and “labels.txt” contain the data for this project. Each row of “features.txt” contains info on a tumor. Each element in the row corresponds to the following information: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses. Each number in “labels.txt” tells what type of tumor each row in “features.txt” is (2 for benign and 4 for malignant). Now, let’s load in the data.

```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier

features = np.genfromtxt("features.txt", delimiter = " ")
labels = np.genfromtxt("labels.txt", delimiter = " ")
```

Numpy’s “genfromtxt” function takes data from a text file and puts it into a numpy array. Since each number in the text file is separated by a space, we set the delimiter equal to a space. Now lets partition the data into training and testing data.

```
train_features = features[:550]
train_labels = labels[:550]
test_features = features[550:]
test_labels = labels[550:]
```

We’ll be using the training data to train our model on 550 data points. The rest of the data will be used as testing data to determine how good our model is. Now, we’ll create the KNN model using sklearn.

```
knn = KNeighborsClassifier(n_neighbors = 10)
knn.fit(train_features, train_labels)

accuracy = knn.score(test_features, test_labels)
print("Our model achieved an accuracy of", accuracy*100, "%")
```

Once we fit our model on “train_features” and “train_labels”, we can evaluate how good our model is by using the “score” function. This function uses our test data (data our model has never seen before) and compares the predicted value to the actual value in “test_labels”.

3 Conclusion

The KNN algorithm is a very simple classification algorithm. There are many more powerful tools, such as neural networks, that can really solve some amazing problems. For example, there are neural networks that can paint pictures and play games such as Dota 2 and Mario Kart. Below, I’ve listed some resources that I’ve used to learn more about Machine Learning and AI:

1. A youtube channel called “Sentdex” that clearly explains many concepts in Machine Learning, including KNN and neural networks.

2. A YouTube channel called “Siraj Raval” that explains amazing applciations of Machine Learning.

3. TJHSST’s Machine Learning Club Website: www.tjmachinelearning.com