Proof of Concept: Scenic Route Application

# Proposition

Is it possible to make a sports application designed primarily for running or perhaps cycling which can create a sort of scenic route based on a user’s interests with the help of Machine Learning algorithms.

# Why?

Traditionally when we talk about distance we think about GPS or GIS to calculate the most optimal or fastest route for us. But what if we actually don’t want the fastest route but actually want to see some countryside, some monuments or other POI’s. Well this is where machine learning comes in.

# Machine Learning Summary

Since I had no prior knowledge of AI (learning) algorithms before I took on this challenge, I had to spend a lot of time researching about machine learning and how it could be applied properly to create such a scenic route.

When I started my research, I was surprised as to how deeply integrated machine learning was in our society. You can find it in our emailing applications to filter out spam, there’s self-driving cars and also all the algorithms Google and co use to determine a user’s interest to display ads and whatnot based on clicks, search results, etc.

Generally machine learning revolves around making predictions with the help of data, be it big chunks or small chunks, obviously the more data the more accurate but the process will take longer. Data basically goes through algorithm which analyses the data and produces and further trains a set model based on all the data entered. It’s different than the traditional way of programming where we give strict and static programming instructions to our systems with all the if and while loops.

In machine learning there’s two types of techniques: supervised and unsupervised learning. Supervised learning is the technique where we enter data with a value (and certain features) we know we want our model to be able to predict as opposed to unsupervised learning where we want to find clusters of data which are alike. Machine learning always uses the same workflow: first it is important to ask yourself the correct question, second thing you need is data, third you’re going to have to select the algorithm suited for your use case, fourth you will have a training model that came out of the data analysis with the help of the algorithm and lastly you will have to test your model with additional data and enhance the model where needed.

# How?

I’m sure there’s different ways and solutions to go about this, since I’m only a beginner I’ll give my take on it based on a week’s research.

Say we needed to make such a scenic route between two locations on Google Maps. As always with machine learning we need data, the first question to ask is of course what kind of data we need and how accurate we want our model to be. In machine learning you have certain algorithms that can recognize patterns in images which is what we will use here. Since we will make a scenic route we need a lot of images but prior to that we probably want a trained model beforehand collecting images so we already have some kind of prediction. Reason why we’d need a model beforehand is because we need a way to know what kind of image we’re looking at so we can link or connect it to POI’s (images in this sense) between the two locations. When we process and classify these images we could for example label them with numbers: 1 for coastal images, 2 for forest images, 3 for cultural buildings and so on. We can link the user interest or input, dependant on how we will determine what the user wants to do or likes, to the images that were classified from our POI’s on the map between the two locations.

So our first step is to find out how to get images from in between two locations, there’s a couple ways to do this: one of them would be to start scraping Google Streetview which is possible because each image has a panorama id attached to it which we can use for labeling, second is to get info from POI’s on Google Maps in a certain radius, again dependant on how accurate we want our app to be (maybe based on user input) which again we can use for labeling and lastly we can scrape data from external websites like Instagram, Tripadvisor and Flickr.

# Sources

<https://www.pluralsight.com/>

<https://www.udemy.com/>

<http://www.digitaleng.news/de/supervised-machine-learning/>

<https://blog.everlaw.com/2015/08/06/cabs-and-machine-learning/>

<https://www.forbes.com/sites/bernardmarr/2016/12/08/what-is-the-difference-between-deep-learning-machine-learning-and-ai/#5954b75326cf>

<http://machinelearningmastery.com/what-is-deep-learning/>

<http://www.randalolson.com/2015/03/08/computing-the-optimal-road-trip-across-the-u-s/>

<https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/>

<https://www.coursera.org/learn/machine-learning/lecture/Ujm7v/what-is-machine-learning>

<http://www.popsci.com/article/science/algorithm-maps-most-beautiful-route-where-youre-going>

<http://infolab.stanford.edu/~ullman/mmds/book.pdf>

<https://www.promptcloud.com/data-scraping-vs-data-crawling/>

<https://andrewpwheeler.wordpress.com/2015/12/28/using-python-to-grab-google-street-view-imagery/>

# Logs

**3-8/08**

* Brush up on Python (since a lot of ML is traditionally written in Python)
* Researching and compiling data about machine learning
  + How does Google calculate optimal paths?
  + Course on Pluralsight and Udemy
  + What is Machine Learning used for these days?
  + Supervised vs unsupervised learning
  + What is Deep Learning
* Data-Scraping vs Data-Crawling
  + What’s the difference between the two
  + When to use what
* Data-Mining

**9-11/08**

* Documenting the proposition, why did I choose it
* What is Machine Learning?
* How will we go about this proposition (theoretically)?
* Basic Exercises in Python about ML
* How to calculate the most optimal path between a number of set points with machine learning
  + Writing a small test in JS for showcase purposes
* How to process and classify images with a machine learning recognition algorithm
  + Small test in python to showcase further implementation for proposition