

MACHINE LEARNING MODEL DEMOCRATIZATION WITH OSM DATA

Shay Strong, PhD

Director of Data Science & ML (Eagleview)

Astrophysics, UT Austin

@shaybstrong

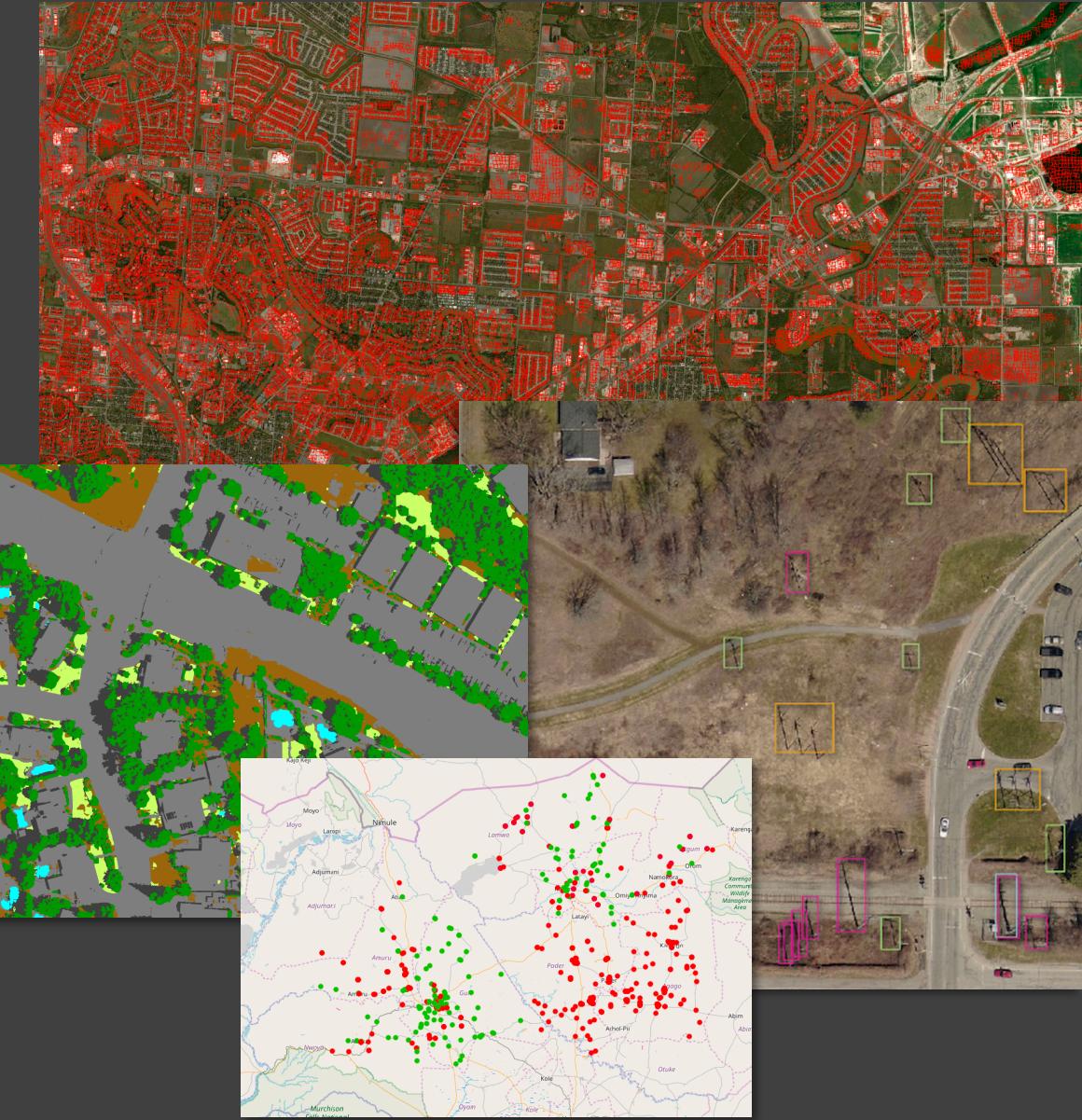
shay.strong@eagleview.com



'every car in Cincinnati'

THE OSM + ML POTENTIAL

- ML (CNN) scales extraction of visual patterns w/ supervision
- A wealth of curated, crowd-sourced expertise that could be used for supervision
 - Still requires additional curation & extraction
- Avenue for adding contributions & education
 - Co-proposed creating a focus at UW (Geohackweek) to modify standard GIS content with end-to-end OSM to model generation
 - *Could we create a user 'market' of open-geoML models?* 🤔



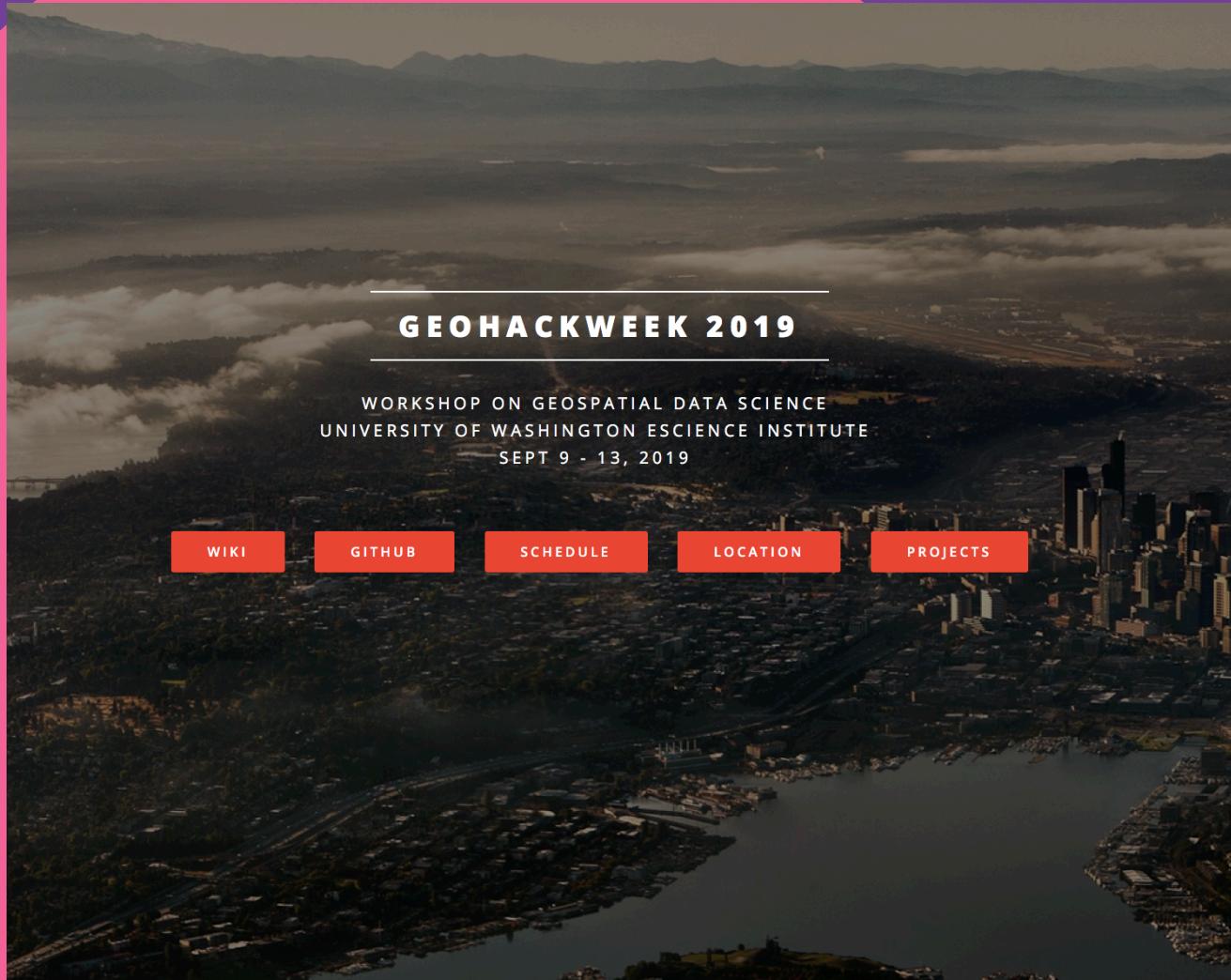
AN OVER-ABUNDANCE OF ML PLATFORMS

- Democratize dev of 'geoML', lower the barrier to entry for students & practitioners, & obliterate the 'practice' of platform commercialization
- geoML is not hard, but daunting
 - Difficulty as a DS with limited cloud-orchestration resources
 - Opensource/open-framework desires (common tools, community driven)
 - The POC is easy. The scaling is hard.
 - Big spenders 'own' the university programs (drive content)
 - Model marketplaces & commercialization that force you into platform use



UW GEOHACKWEEK

(<https://geohackweek.github.io>)



GEOHACKWEEK 2019

WORKSHOP ON GEOSPATIAL DATA SCIENCE
UNIVERSITY OF WASHINGTON ESCIENCE INSTITUTE
SEPT 9 - 13, 2019

WIKI

GITHUB

SCHEDULE

LOCATION

PROJECTS

- **2018 Software Carpentry Approach**
 - Last minute volunteer to teach ML in 1 hr 1 month after moving to Seattle
 - 1 week to cover many aspects of geospatial + github + opensource
 - Team projects
- **2019 ML Revolution**
 - We can do so much more!
 - Leverage HOT-OSM data sets as vector training data
 - **Raster -> Vector -> Machine Learning**
 - <https://github.com/scottyhq/geohackweek2019-raster>
 - https://github.com/geohackweek/tutorial_contents/tree/master/vector/notebooks
 - <https://github.com/shaystrong/sagely/>

EDUCATION

- Leveraging OSM + ML + cloud compute
 - University of Washington's GeoHackweek Multi-disciplinary, multi-background
 - HOT-OSM tasking (vector) → ML outcome building prediction in 1 week!
 - 50 new OSM mappers!!
 - AWS sponsorship (negotiated free compute!)
 - **Personal goal (unaffiliated):** *Advance stagnant geospatial activities and create a community invested in producing optimal solutions that become foundational to advanced, open endeavors.*



Search or jump to... Pull requests Issues Marketplace Explore

University of Washington Geohackweek
eScience Institute <https://geohackweek.github.io/>

Repositories 42 Packages People 95 Teams 7 Projects

Find a repository... Type: All Language: All New

preliminary
Preliminary Tutorial
Jupyter Notebook 3 0 1 0 Updated 5 days ago

geohackweek.github.io
CSS 27 14 0 0 Updated 5 days ago

sample_project_repository
Sample repository for Hackweek projects
Jupyter Notebook 0 0 0 0 Updated 17 days ago

datasharing
JavaScript 2 0 0 0 Updated 17 days ago

administrative
Hackweek project management.
Jupyter Notebook 0 0 22 0 Updated 18 days ago

wiki
5 0 0 0 Updated 18 days ago

hackweeks
0 0 0 0 Updated 20 days ago

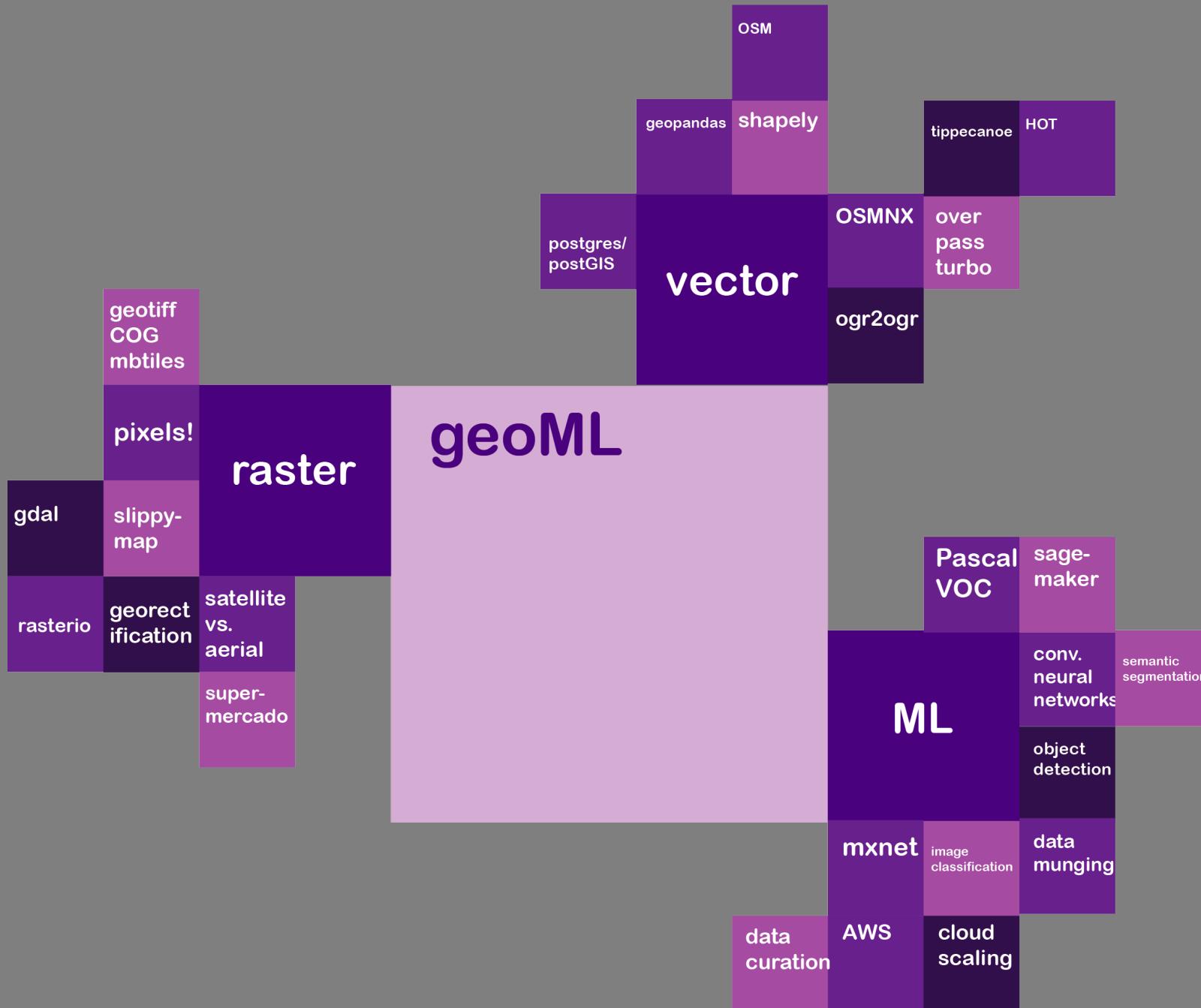
reproducible-research
JavaScript 2 0 0 0 Updated 28 days ago

ghw2018 Archived
CSS 2 0 0 0 Updated on Jun 26

Top languages
Jupyter Notebook JavaScript HTML CSS Dockerfile

People 95 >





OSM vector data as ML training labels

Cyclone Kenneth 2019-04-25

Part I

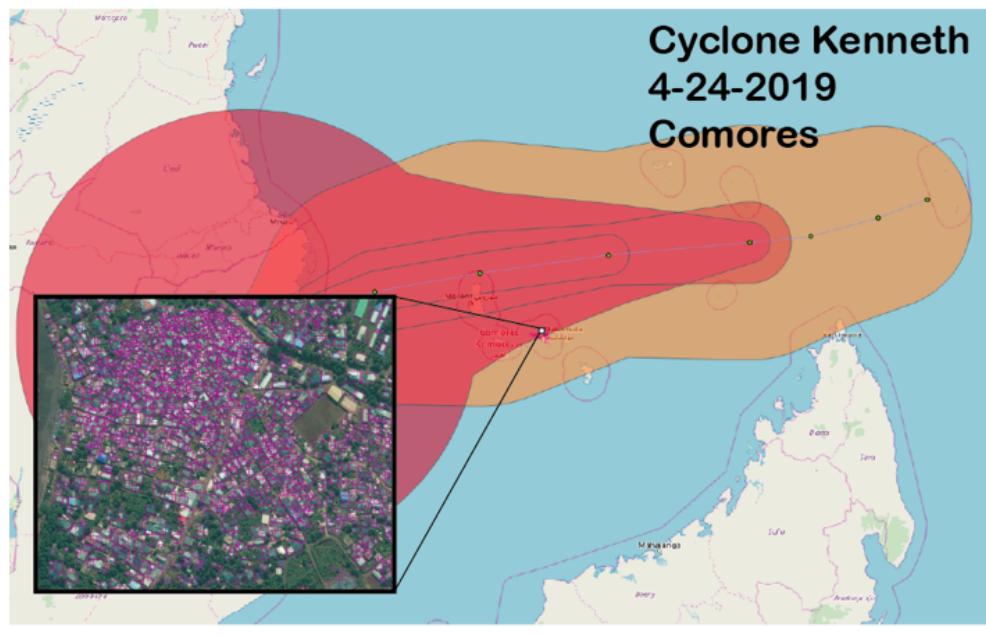
So I wanted to create a seamless tutorial for taking OpenStreetMap (OSM) vector data and converting it for use with machine learning (ML) models. In particular, I am really interested in creating a tight, clean pipeline for disaster relief applications, where we can use something like crowd sourced building polygons from OSM to train a supervised object detector to discover buildings in an unmapped location.

The recipe for building a basic deep learning object detector is to have two components: (1) training data (image (raster) + label (vector) pairs) and (2) model framework. The deep learning model itself will be a Single Shot Detector (SSD) object detector. We will use OSM polygons as the basis of our label data and Digital Globe imagery for the raster data. We won't go into the details of an SSD here, as there are plenty sources available. We will run the object detector in [AWS Sagemaker](#).

In this part I you will:

1. Get vector data from OSM
2. Convert them to labels for a CNN object detection (using Apache MXNET)
3. Store them in VOC style
4. Create optimized .rec files for porting them into the AWS Sagemaker world
5. AWS S3 & EC2

I anticipate using this tutorial in conjunction with [HOT-OSM](#) related tasks -- where we may have drawn vector data as part of a specific project and know it exists. For the purpose of establishing a demo, we will use a recent HOT OSM task area that was impacted by Cyclone Kenneth in 2019, Nzwani, Comores.



<https://github.com/shaystrong/sagely>

AWS Sagemaker Training and Deploying

Cyclone Kenneth 2019-04-25

Part II

In this part II notebook, we will upload the data to AWS S3 that we generated for training in the previous notebook. We will kick off an AWS Sagemaker object detection job and monitor the results. At the end of this notebook, you will have trained your own OSM-based CNN object detector!



A couple of things worth noting:

- 💡 ML models are not super useful unless they are scaled across a large amount of data
- 💡 To effectively scale across data, you need to be efficient
- 💡 Because we will be passing sensitive data to this notebook in order to scale our cloud compute through Sagemaker, we will use papermill to run this notebook from within python. It creates a simple wrapper around the notebook so that we can specify variables.

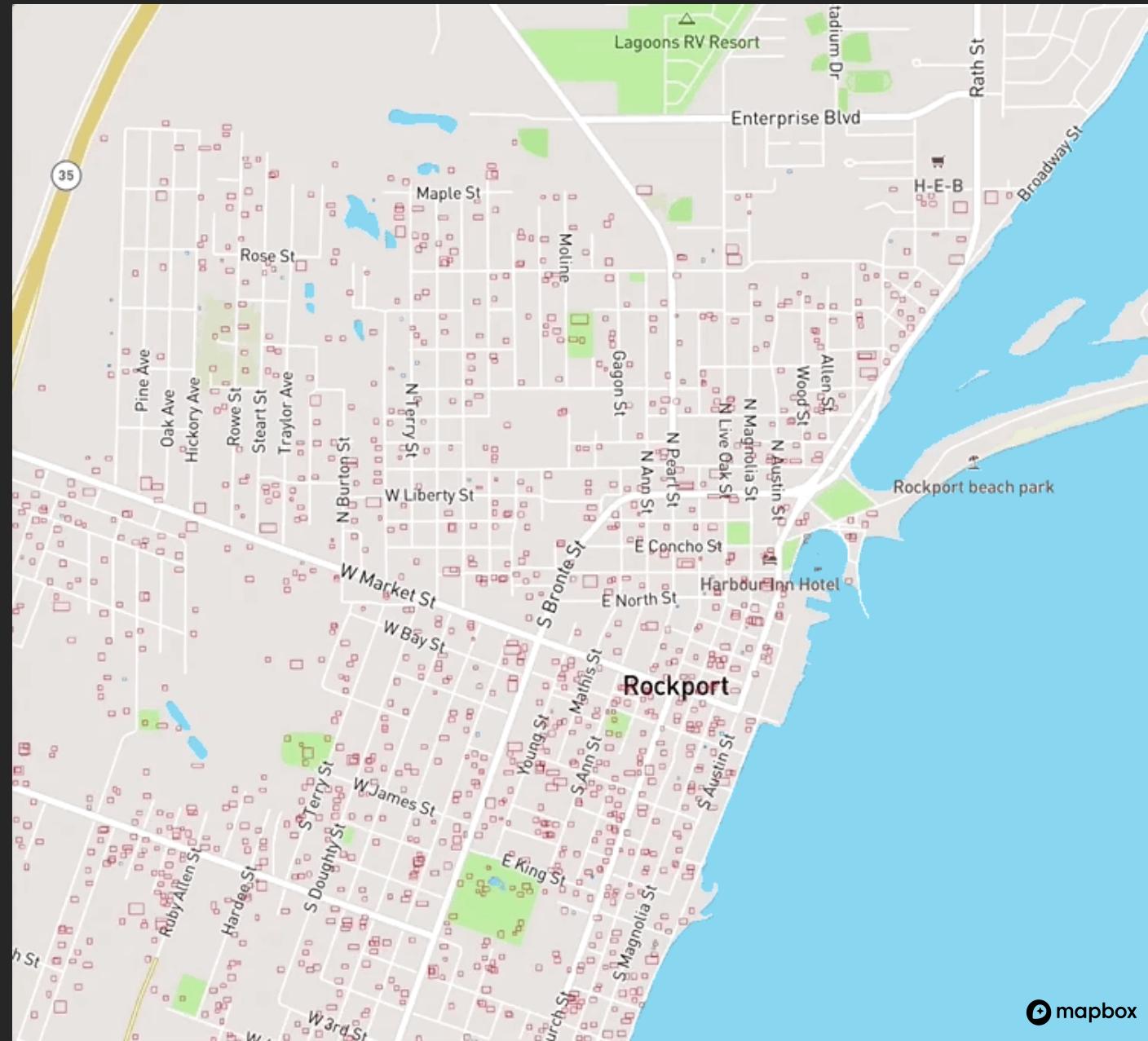
e.g.

```
import papermill as pm
pm.execute_notebook('osm_ml_training_pt2.ipynb','osm_ml_training_pt2_out.ipynb', parameters
= dict(sage_bucket='',my_bucket='', role=''))
```

```
In [9]: import sagemaker
from sagemaker import get_execution_role
from sagemaker.amazon.amazon_estimator import get_image_uri
```

We will use 'papermill' (<https://github.com/nteract/papermill>) to pass sensitive variables to this jupyter notebook. Things like passwords, cloud locations, etc, should be parameterized as a best practice -- Never stored in a repo (especially public facing).

OSM TO STORM RESPONSE



- Flood commercialized marketplaces with models leverage-able by a broader community
 - Extricate 'free' models from their marriage to black-box platforms
- What could incentivization or a social currency approach look like in the long run?
- Challenges: Imagery, cloud compute, annotation cleansing

The screenshot shows the Algorithmia AI Layer interface. At the top, there are navigation links: Product, Pricing, Learn, Sign In, Try It For Free, and Get Your Demo. Below this is a search bar with a magnifying glass icon and a 'Search' button. The main section is titled 'Algorithms' and features four cards: 'Text Analysis' (Make sense of unstructured text), 'Machine Learning' (Teach your app to teach itself), 'Computer Vision' (Identify objects in images), and 'Deep Learning' (Learn from your data). Below these cards are links for 'Utilities', 'Microservices', 'Web Tools', 'Time Series', and 'Sentiment Analysis'. A 'Top Rated' section lists 'Colorful Image Colorization' and 'Summarizer'. At the bottom, there are filters for 'Top Rated', 'Most Called', and 'Recently Added'.

The screenshot shows a 'WIRIED' magazine article. The title is 'NEED SOME AI? YEAH, THERE'S A MARKETPLACE FOR THAT' by DAVEY ALBA, BUSINESS 09.15.16 01:27 PM. The article features a large image of a person's face and a 'Share' section with links for Facebook and Twitter.

The screenshot shows the AWS Marketplace interface. The search bar is set to 'Machine Learning (382 results)'. The results list includes 'Deep Learning Notebook (Python 3, Tensorflow)' by '1.13, Pytorch 1.1' (sold by 'SignoData'), 'HERE Location Services' by 'HERE' (version 1.0), and 'H2O.ai Driverless AI Latest (BYOAI)' by 'H2O.ai' (version 1.7.1). Each listing includes a brief description, price, and user reviews.