**DATA SCIENCE PROGRAMMING**

**FINAL PROJECT: GEOSPATIAL ANALYSIS**

**Fall 2019**

Submitted by: Professor:

**Karan Gupta Padmanabhan Balaji**

**Pankaj Patel**

**Soumya Roy**

**Vivek Singh**

**Introduction**

In our project, we address the issues regarding classification of spatial data and concentrate on building decision trees for the classification of such data. We used Geometric Information Systems (GIS) to build a solution to label “Rural” and “Urban” counties in 5 different states in the United States. These labels have been created based on the area of the county, the length of the roads and the length of rails passing through the county. We have calculated all of these fields using libraries like Geopandas, Shapely and Fiona

Further, we have built a solution to calculate the distance and travel time of a facility location from a given incident point, such as, the nearest hospital location and drive time from an accident spot.

To do this, we have used geospatial data files from the GIS system (geographical information system) and ArcGIS APIs.

Geospatial data provides the strong technical support for multi-level geo-spatial data acquisition, such as photogrammetry to remote sensing, traditional paper surveying and mapping to digital surveying and mapping, from mobile surveying and mapping to sensor networks.

**Related Work**

We have found similar projects of labelling American counties as Rural/Urban done in the past, however, the criteria that was used for those projects were very different. Mostly, they used population density, number of hospitals, libraries, recreation centers etc.

Moreover, we came across projects that were based on satellite imagery for classification of the counties. However, there could be problems such as image distortion due to bad weather or clouds. and is tailored for the astronomical application only. Thus, it is not suitable for these kind of classification problems.

**Problem statement**

The Precise Problem Statements identifies:

•Use of geospatial data for calculating the total road coverage and the total rail coverage for each county in 5 states in USA (Florida, Georgia, New York, California, Texas).

•Using these calculated fields to label if a county is “Urban” or “Rural” based on certain set of rules.

•Using machine learning model to train the classifier based on these labels.

•Using this concept, calculated the distance and travel time from an incident point to a facility point. For example, the nearest hospital from an accident point or the nearest shelter point from a disaster point.

**Data & libraries**

The datasets are obtained from below hyperlinks:

USA County Boundaries:

<https://www.sciencebase.gov/catalog/item/4f4e4a2ee4b07f02db615738>

US Rails dataset:

<https://hifldgeoplatform.opendata.arcgis.com/datasets/2a9677db741d4a78bd221586fe9a61f50>

US Roads:

<https://www.naturalearthdata.com/downloads/10m-cultural-vectors/roads/>

Libraries used:

* **Scikit-learn**
* **Shapely**
* **Matplotlib**
* **Python Shapefile Library (pyshp)**
* **ArcGIS**
* **pyproj**
* **Rasterio**
* **GeoPandas**
* **Pickle**

**ArcGIS:** The ArcGIS API for Python is a powerful, modern and easy to use Pythonic library to perform GIS visualization and analysis, spatial data management and GIS system administration tasks that can run both in an interactive fashion, as well as using scripts.

**Geopandas**: It is a Python library for working with vector data. It is based on the pandas library that is part of the SciPy stack. SciPy is a popular library for data inspection and analysis, but unfortunately, it cannot read spatial data. Geopandas was created to fill this gap, taking pandas data objects as a starting point. The library also adds functionality from geographical Python packages. Geopandas offers two data objects—a GeoSeries object that is based on a pandas Series object and a GeoDataFrame, based on a pandas DataFrame object, but adding a geometry column for each row. Both GeoSeries and GeoDataFrame objects can be used for spatial data processing, similar to spatial databases.

**Rasterio:** It is the go-to library for raster data handling. It lets you read/write raster files to/from [numpy](https://docs.scipy.org/doc/numpy-dev/user/quickstart.html#the-basics) arrays (the de-facto standard for Python array operations), offers many convenient ways to manipulate these array (e.g. masking, vectorizing etc.) and can handle transformations of coordinate reference systems.

**Scikit-image:** Library for image manipulation, e.g. histogram adjustments, filter, segmentation/edge detection operations, texture feature extraction etc.

**Scikit-learn:** The best and at the same time easy-to-use Python machine learning library. Regression, classification, dimensionality reductions etc.

**Descartes:** Enables plotting of shapely geometries as matplotlib paths/ patches. Also, a dependency for the geometry plotting functions of geopandas.

[**Pyproj**](http://jswhit.github.io/pyproj/)**:** For transformation of projections. It is mostly unnecessary when using the more convenient geopandas Coordinate Reference System (CRS) functions.

**Shapely:** manipulation and analysis of geometric objects

**Pickle:** Itis used for serializing and de-serializing Python object structures, also called marshalling or flattening. Serialization refers to the process of converting an object in memory to a byte stream that can be stored on disk or sent over a network

**Fiona:** read and writes geographic data

**Methodology**

The methodology used for the Project includes two parts:

**Part1:**

-Reading the shape files for counties, roads, rails in USA and convert to Geopanda DataFrames.

-Converting the default co-ordinate reference system (CRS epsg 4326) for each DataFrame to epsg 2778.

-Calculating the area for each county using polygon geometries present in the respective shape files.

-Find the intersection of roads within each county for 5 states. Then calculate the total lengths of roads in each county by summing up these intersections.

-Performed the same process for Rails.

-Converted the road lengths, rails lengths and the area of counties to miles.

-Create the Labels (Urban/Rural) based on the rules derived from the above created columns.

-Train a machine learning model to evaluate how well the rules are being learned by the model. Try to increase the accuracy of the model by performing hyperparameter tuning.

-Visualizing the rules learnt by the decision tree model.

**Part2:**

-Used the network module from ArcGIS to find the hospital that is closest to an incident.

-The closest facility solver gives usefulness to discovering the nearest areas to a specific information point

-Establishing connection to ArcGIS Online organization

-Established a ClosestFacilityLayer object to determine the nearest facilities and best routes to reach those facility locations.

-Created hospitals layer.

-Created incidents layer

-Performed the shortest route calculation from the incident point to the hospital.

**Techniques**

The following key techniques are being used in the project:

i) The geopandas function **Intersection** being used for the “Intersection length of Roads”

ii) Decision tree classifier from Scikitlearn

iii) Geopandas function **geometry.area** which returns a series containing the area of each geometry

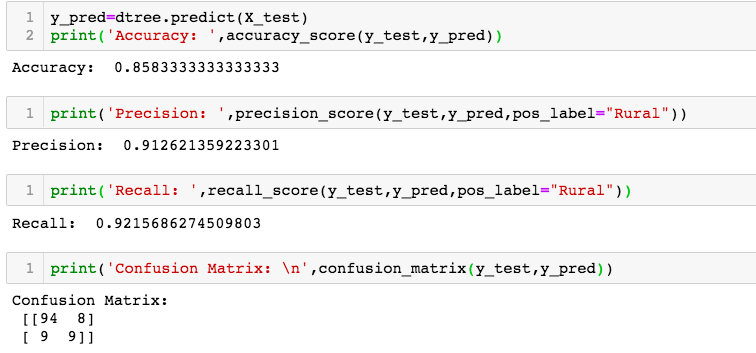
iv) **ArcGIS API for Python:** This network module can be used to solve different types of network analysis operations. In this sample, we see how to find the hospital that is closest to an incident.

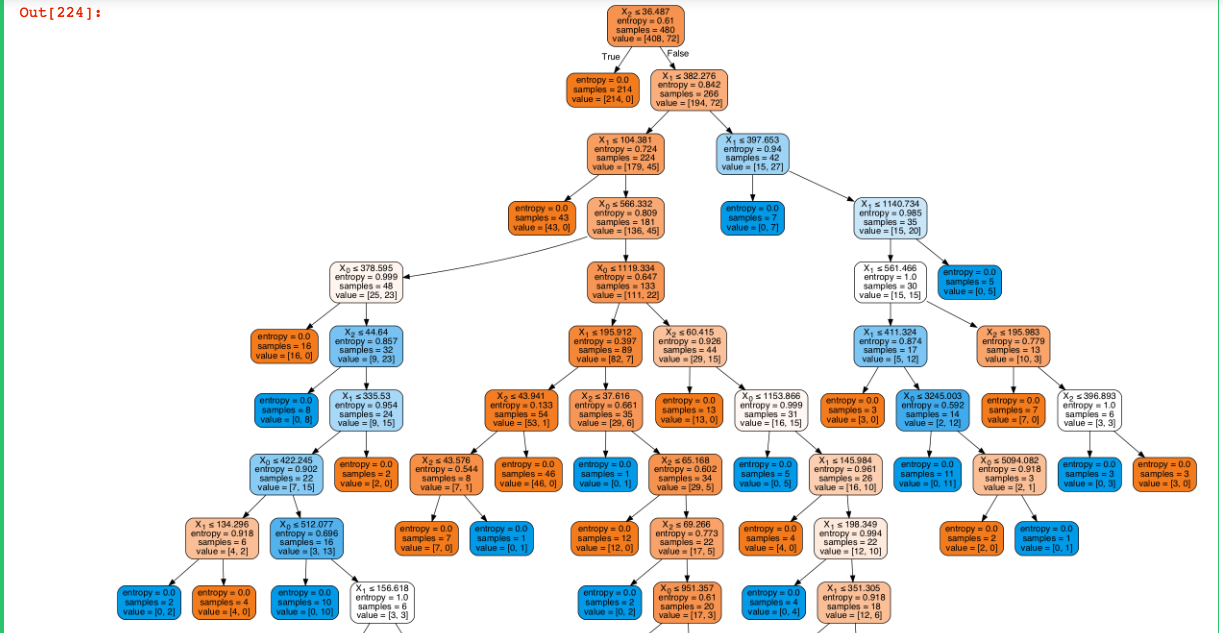
**Evaluation**

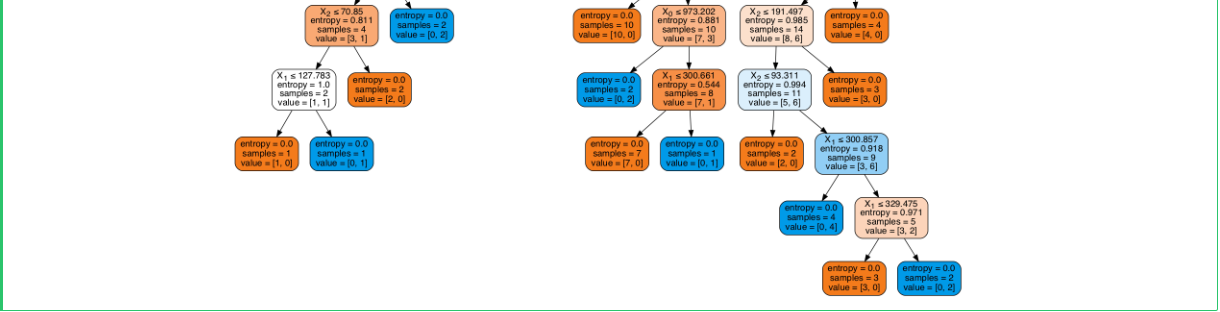
We have used a decision tree classifier for predicting the labels. The metrics that we have used for evaluation are Accuracy, Precision, Recall and Confusion Matrix. (Screenshot below)

However, in this case, accuracy is the key evaluation metric since we wanted to evaluate how good is the model in learning the rules. We have also plotted the decision tree to visualize the rules that the model has learnt. This was possible only with a decision tree classifier.

Later we tried to change the parameters of the model to check if the model can learn the rules better and make improved predictions.

****

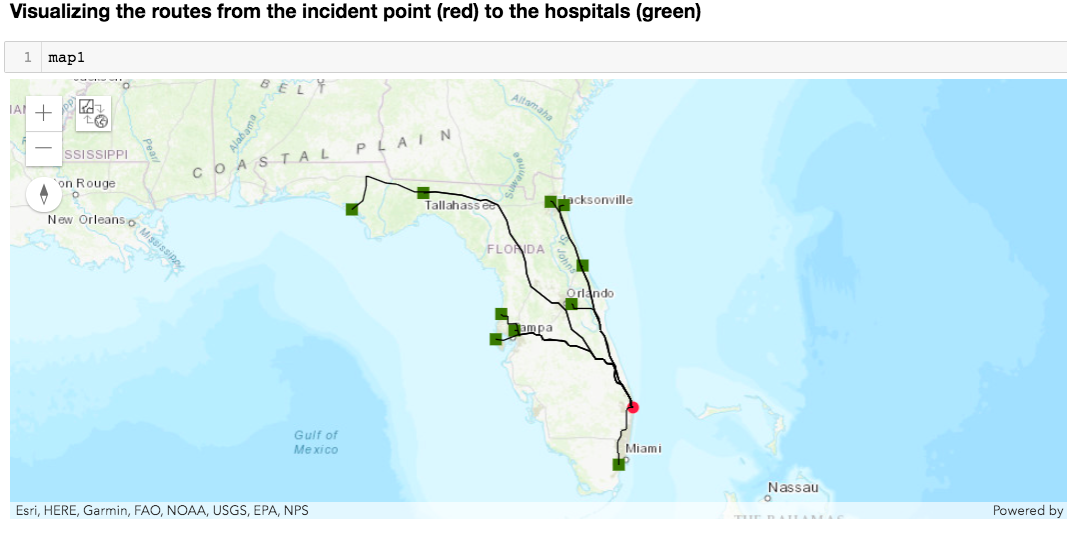
****

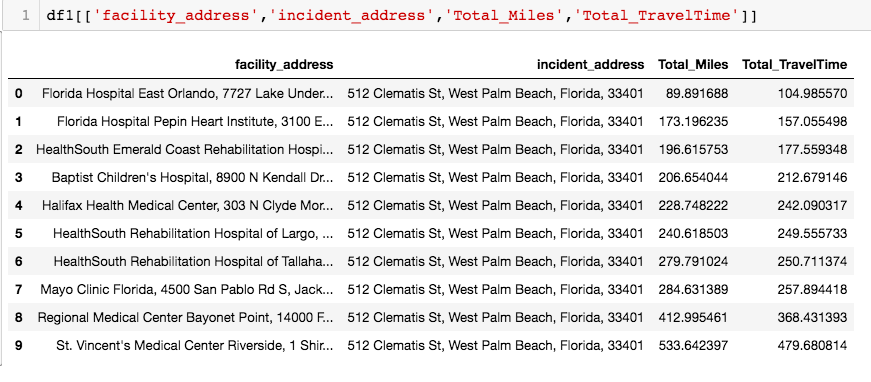
****

**Results**

Below are the Part 2 results for showing the incident spot (red spot) and the routes connecting to all the given list of hospitals.

The results are sorted by the distance from the incident spot (shown in Pic2)

****

****

**Future Work**

Currently, we have used just few parameters as the feature set for the prediction. We want to increase this parameter space that will in turn help in creation of better labels. For example, include the ratio of land vs water area, population density etc.

The rules that we have currently applied to create the labels are simplistic and straight forward in nature. We want to create complex algorithms for the labelling part.

Currently we have included only 10 hospitals to demonstrate the functionality for finding out the nearest hospitals to the incident point. In future, would want to widen this application to include a larger set of hospitals covering a bigger area. We would also want to apply this logic for other facilities such as fire stations, shelters, hotels etc.

**Implications (how can this be used?)**

Various classification labels using complex rules based on derived information from geographies can be implemented. For instance, if a county requires any development or not based on the present infrastructure or how good is a county in terms of connectivity if a new business/public facility needs to be setup

Using the geospatial coordinates, precise locations can be identified and based on this location any mishap like fire, robbery, accidents, etc. in that location can be reported to the nearest fire station, police station or hospitals depending on the case.

Crime Rate Analysis based on location and geography can be performed using geopandas and ArcGIS libraries.

Land cover classification for remote sensing using deep learning is another important application of geospatial analysis.

**References:**

<https://developers.arcgis.com/labs/what-is-arcgis/>

<http://geopandas.org/>