

---

# **AgriTech---USGS-LIDAR**

***Release 0.1***

**Rukundo Germain**

**Aug 22, 2021**



## CONTENTS



This project creates a package to access Entwine Point Tile(EPT) LiDAR data from AWS and return the data in the format of a geopandas dataframe for ease of analysis.

**AgriTech—USGS-LIDAR** is a project for analysing how water flows through a maize farm field. Water is very important for crop growth and health. We can better predict maize harvest if we better understand how water flows through a field, and which parts are likely to be flooded or too dry. One important ingredient to understanding water flow in a field is by measuring the elevation of the field at many points. The USGS recently released high resolution elevation data as a lidar point cloud called USGS 3DEP in a public dataset on Amazon. This dataset is essential to build models of water flow and predict plant health and maize harvest.

**TO BE DONE:** produce an easy to use, reliable and well designed python module that domain experts and data scientists can use to fetch, visualise, and transform publicly available satellite and LIDAR data. In particular, your code should interface with USGS 3DEP and fetch data using their API.

**DATA:** LIDAR high definition elevation data - **USGS 3DEP** - The **USGS recently released high resolution elevation data as a lidar point cloud in a public dataset on Amazon**. This dataset is complicated to understand and use, and it would be useful to have an easy way to access and use it in order to build models of water flow and predict plant health and maize harvest.

### INSTALLATION:

**These libraries must be installed before :** Laspy

Geopandas

rasterio

PDAL

### HOW TO INSTALL:

pip install <library name>

pip install -r requirements.txt # requirements.txt contains all needed for the project to be installed

### 3 MAIN PARTS OF THE PROJECT:

- *Data Fetching and Loading:* the task is to write a modular python code/package to connect to the API, query the data model to select with a specified input and get a desired output. For example, submit a boundary (GPS coordinates polygon) and receive back a raster of the height of the terrain within the boundary. [Tutorial](#)

The first step was to access the USGS data through the public [data url](#)

A .txt is created for the regions available and used to create a dropdown where a user can choose the regional data they would like to analyze. The data used for analysis in this data is from the state of IOWA, with the label 'IA\_FullState'.

The boundaries will be user inputs that depend on the area a user wants to analyze. The boundaries are reprojected before input into the pdal pipeline.

Using the pdal library, we create a pipeline that can be used to access the data inside the ept.jsons. The pipeline uses read.ept to read the file, the second layer filters the dataset depending on classes, we retain classes 2-7 and 9. The classes represent the type of data collected e.g. ground. The third layer writes the data in 3 different formats, the first format is a .laz file, the .laz file is transformed to a .tif. The data is also written to a .geojson in the pipeline.

The .geojson is used to create a geodataframe and the geometry of the dataset is changed to two dimensions, removing the elevation. The elevation is represented as a column in the final dataframe.

The .tif file is plotted to produce a pixel image of the area. The .tif is also used to generate the shapefile of the area.

scripts used can be found on [github](#)

- ***Terrain Visualization:*** Include an option to graphically display the returned elevation files as either a 3D render plot or as [codes and visualisations](#)
  - ***Data Transformation:***
    1. Topographic wetness index (TWI) - as an additional column returned with geopandas dataframe
    2. Standardized grid - A python code that takes elevation points output from the USGS LIDAR tool and interpolates them to a grid.
- 

---

**Note:** This project is under active development.

---