

Introduction	2
Dataset descriptions	3
Quantum geographic information system.....	4
QGIS	4
PostGreSQL	6
PostGIS	7
Projects which use similar technologies	10
Blender.....	10
QGIS	10
VU.CITY.....	10
Projects which use VU city data.....	11
Farrells Architects	11
Iceni Projects.....	11
Geocoding address.....	12
Programming Stack	13
Python	13
Geopy.....	13
Pyproj.....	13
Numpy.....	13
Re	14
Mathplotlib	14
Pandas.....	14
SciPy	14
Pathlib	14
Time	15
Pickle	15
Plotly.....	15
SKlearn	15
Other languages.....	15
Blender.....	17
Dash	19
Machine Learning.....	20
Convolutional Neural Networks.....	20

Clustering	22
K-Means	23
Hierarchical Clustering	24
Bibliography	26

Introduction

This project is being completed for Lanu. The goal of the project is to automatically generate an accurate 3d model of a house in the UK given just its address as an input. To do this the project will create a Postgre database which will later be accessed by python using PostGIS functions and processed by machine learning algorithms until the shape of the given house is clear. Then a blender model like the one pictured in Figure 1 will be created. This example model is one Lanu already created using a different process.

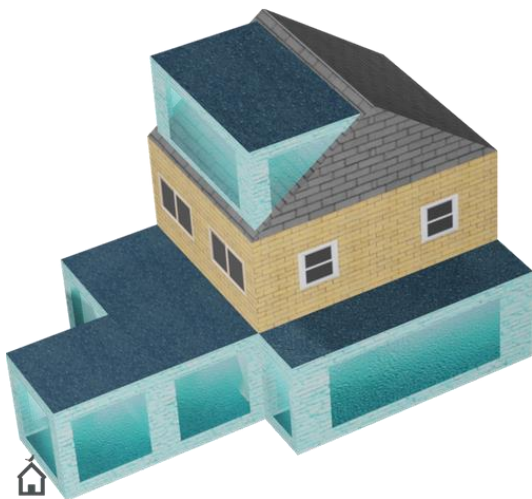


Figure 1, Example output

Dataset descriptions

This project will contain two separate datasets.

The National Polygon Service (NPS) provides chargeable data about registered land and property. NPS is a sub-set of the land register which contains all registered titles relating to lands and property. This dataset shows the indicative shape and position of each boundary or legal intersection of a registered tile for land and property in England and Wales. Every freehold or leasehold title has at least one index polygon. This dataset has more than 25 million titles and 28 million polygons. The sheer scale of this dataset can be seen in Figure 2. The data included in this dataset contains geometry specific to sites across England. This includes the X and Y values corresponding to the shape of the site. The dataset also contains columns corresponding to ID, date of creation, and when updates took place, and a version of the polygon identification number. The NPS dataset has seen uses in planning and property technology all over the UK by different factions of the government and business.

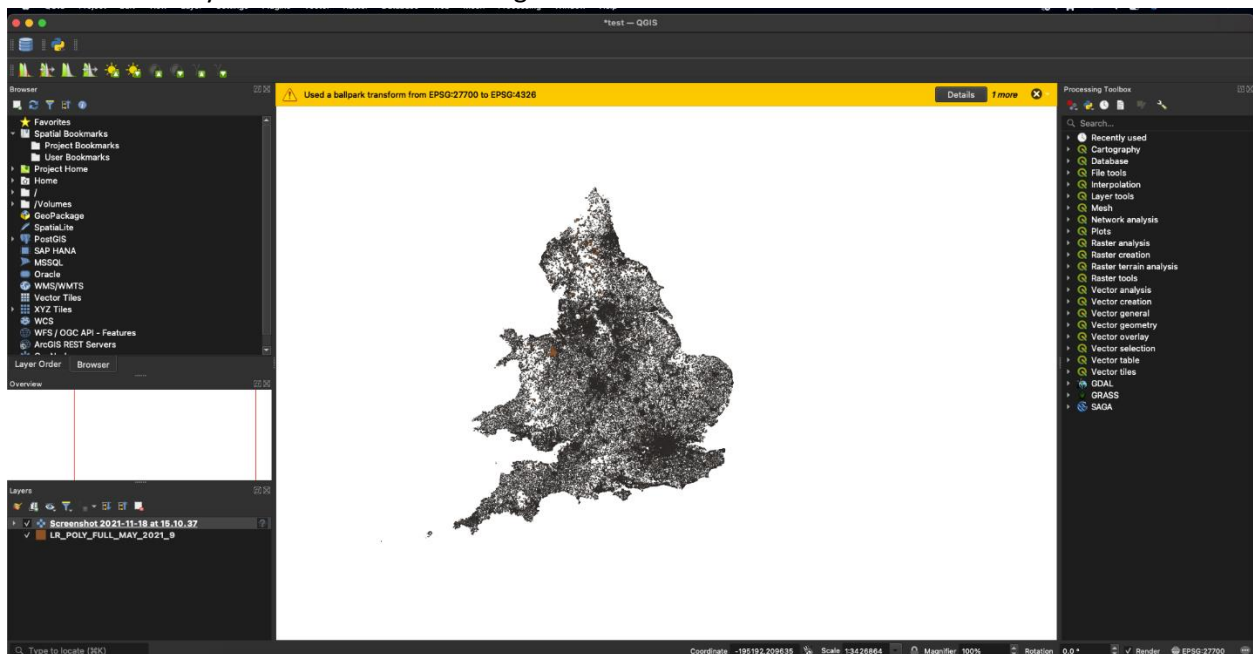


Figure 2, QGIS, NPS Dataset

The NPS dataset is a proven generally accurate and expensive dataset. It is sourced directly from the UK government. Who in relation to this specific dataset have tried to “make sure that our public data is accurate, but we cannot guarantee that is free from errors or fit for your purpose or use.”? Although its credibility comes from the British government. The dataset does contain some slips in quality. “Overlaps, gaps, slivers and underlaps all exist in the dataset” (GOV-UK, 2021). These may cause some problems to arise as it will become increasingly difficult to solve these errors with a coding solution. For example, it may prove more difficult to find a site's neighbors if an error has caused an echelon of a gap

between the two sites. To solve this problem a clever PostGIS solution may be necessary. The NPS dataset is also updated monthly, adding new sites, or updating site changes to the dataset. This will also make the dataset more accurate as it will not be left outdated. The NPS dataset is at the heart of this project. It will be used to extract the geometries of a given site. The accuracy and reliability of this geometry are crucial to the success of the project. (Admin, 2021)

The other dataset which I am going to use is a dataset obtained from a company called "Vu. City". This company maps entire cities extremely accurately using satellite data. In making these models, they make large raster datasets (VU-City, 2021) The dataset which was obtained from VU.CITY contains a square kilometer of normal vectors, each vector giving the height in that specific spot within the square kilometer. This dataset was obtained in a JSON format and was quickly changed to an array of ten pickle files which store data of a single 100-meter squared area.

Since their models are accurate the dataset can be said to be at least somewhat accurate. Although the only method to discover inaccuracies is to use the dataset and test its accuracy. Upon further research and some testing. It was discovered that some of the normal vectors in the dataset have negative Z values. This should be impossible and may be due to an error on VU. CITYs side. To solve this these vectors were simply multiplied by negative 1 as the values of the vectors seem to be correct.

This dataset will be used in the project in connection with the NPS dataset to determine the heights of all the points in each site queried. This is vital to the project's success. The dataset's existence is what sparked the idea for this project into existence.

Quantum geographic information system

QGIS

Quantum geographic information system (QGIS) is a free and open-source geographic information system (GIS). It allows users to create, visualize and analyze geospatial data and can be used on Mac, Windows, Linux, and mobile devices. It was originally developed by Gary Sherman in 2002, initially being released in 2009. QGIS was written in C++ and made extensive use of the QT library. Since 2012 QGIS is maintained by enthusiastic and engaged volunteer developers who regularly release updates. As a free software application that is licensed under GNU, QGIS can be freely modified to perform different or more specialized tasks. QGIS offers a wealth of GIS functions, provided by core features and plugins. (Admin, 2021)

Users can view combinations of vector and raster data (in 2D or 3D) in different formats and projections without conversion to an internal or common format. Supported formats for which to view this data include but are not limited to spatially enabled tables and views using vector formats such as PostGIS and spatiality, raster, and image formats which are installed by the Geospatial Data Abstraction Library (GDAL), mesh data, vector tiles, and spreadsheets. This makes QGIS a great choice for visualizing large datasets. QGIS also grants you a friendly GUI system that allows users to interactively explore spatial data. This GUI provides helpful tools such as a DB manager to help you manage your databases, an on-the-fly reprojection system, a spatial bookmarking system as well as other tools. also allows users to create, edit, manage and export vector and raster layers in such formats as GPX, GPS, and DXF. The

ability to create and edit multiple files formats and GRASS vectors layers as well as tools for handling and managing both vector and attribute tables, proves QGIS to be a valuable tool in dealing with said large datasets. An example of the QGIS interface can be seen in Figure 2. QGIS also gives the ability for users to perform spatial data analysis on spatial databases and other OGR-supported formats. QGIS currently offers vector analysis, sampling, geoprocessing, geometry, and database management tools. Analysis functions are run in the background, allowing users to continue working on datasets before the process has been completed. Other features of QGIS include the ability to publish maps on the internet, extend the QGIS functionality through plugins allowing QGIS to be adapted to your special needs, and use an inbuilt python console to write scripts for interaction with the QGIS environment. QGIS is not the only software application that grants users such features. (Dempsey, 2012)

QGIS also competes with other free-to-use software such as GRASS GIS, Mapserver, what3words, and SAGA GIS. Whilst these are free for users to use there also exists proprietary paid geographic information system software such as ArcGIS, SuperGIS, and GIS Cloud. The most popular competitor of QGIS is ArcGIS, the use of either software over the other is a decisive issue among software engineers and geographers alike. For this project needs research has shown QGIS to be the more suitable choice.

A major factor in this is the fact that QGIS is free and open source. This allows users to become familiar with QGIS and its features, layouts, and abilities without risking the loss of capital. The use of open-source GIS is also growing, expanding the active volunteer force which continues to work on QGIS on a regular basis and expand its functionality whilst fixing bugs. QGIS also provides great documentation for new users allowing for easier familiarization with the product. If a user is struggling with a certain issue, they can always submit a post to the QGIS stack exchange. The QGIS site also has a help section with introductory tutorials, educating users on how to work with raster and vector data (Docs, 2021). For this project documentation and easily accessible help are needed as it is my first experience in using a GIS system. Research has also shown that QGIS benefits over ArcGIS as it has superior data visualization. Such features as the interactive styling dock which has been described as “addictive”, something a master in GIS software “can’t live without” (Menke, 2017) as well as complex features like the array of QGIS renderers and sub-renderers and Blending models. Another edge QGIS has over its most fierce competitor, ArcGIS, is its ability to consume versatile and large data. “No questions asked. QGIS has the edge for consuming data “QGIS supports over 70 vector formats, a much larger amount than ArcGIS. This may be important to the project as large datasets are being dealt with. Since working with PostGIS functions may be a major part of the project. The choice to use QGIS is glaringly obvious as QGIS has the benefit of being “born to work with PostGIS”. (GISGeography, 2021)

The project will use QGIS and its PostGIS features to initialize, crop, and query The National Polygon dataset (NPS) which has been previously discussed (Kurt, 2012). Firstly the entire dataset will be initialized to a table inside PGAdmin4 which is software allowing users to create a database and link it to QGIS. Once linked the Geoprocessing tools which QGIS provides, specifically the ‘Clip’ tool, will be used to crop the dataset into a more manageable size. Once in this manageable size the dataset will be scanned visually using QGIS’s great visualization browser which will display the data accurately on screen. In doing this unfor seen errors in the dataset may be discovered. If so the vast editing features

of QGIS may aid in solving these issues. The NPS dataset should then be able to be queried for specific results and further processed on the front end of the project.

As the steps involved are novel some may cause challenges. This may include the difficulty of correctly linking PGAdmin4 to QGIS or cropping the dataset with the clip tool. To aid in this I will use gis.stackexchange.com to help in finding solutions to problems which may arise. An example of a PGAdmin4 structure can be seen in Figure 3 below.

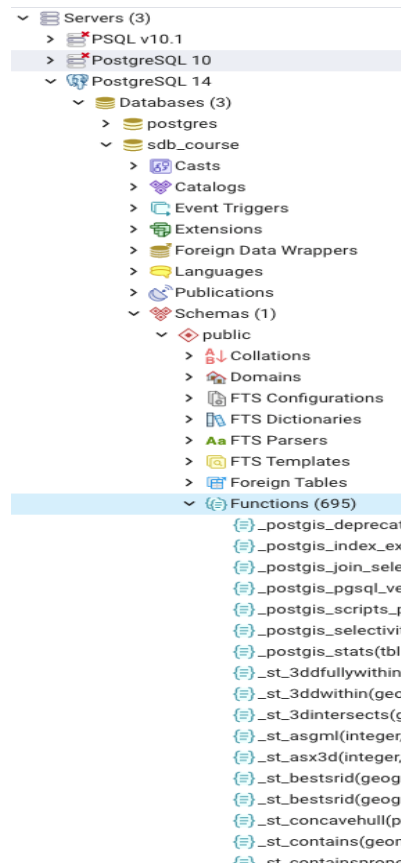


Figure 3, PgAdmin4 interface and structure

To conclude, research has proven that QGIS is an excellent choice of operator for the NPS dataset. It is a well-documented, easily accessible free to use tool which contains features that are more than sufficient for this project use-case scenario.

PostgreSQL

PostgreSQL is the leading open-source relational database. It has been developed for 30 years. It has proven reliability and performance, robust features, and data integrity. “PostgreSQL can form the core database of any kind of service, application or system” (Admin, 2021). It has hundreds of open-source plug-ins and extensions which can be used to extend its core capabilities. PostgreSQL’s stored procedures allow users to carry out operations that might normally take several queries in a single function.

PostgreSQL is used across all types of tech and related companies to store large amounts of data. “5296 companies reportedly use PostgreSQL in their tech stacks, including uber, Netflix” and Instagram. This shows that PostgreSQL is a reliable and tested relational database way to store data. This project will utilize PostgreSQL to store the NPS dataset. Shp2pgsql is a command-line tool to import ERSI shapefiles into PostgreSQL. This project will utilize shp2pgsql to import the NPS dataset into a PostgreSQL. The PostGIS extension will then be implemented to interact with the database. (Anon., 2019)

PostGIS

The use of PostGIS along with its vast functions and features which acts as the back-end of the project is vital to the project's success.

PostGIS is a spatial database extender for a PostgreSQL database. It adds support geographic objects allowing location queries to be run in SQL. PostGIS includes support for Gist basrf R-Tree spatial indexes, and functions for analysis and processing of GIS objects. PostGIS was developed in April 2001 by Refrations Research Inc. Although PostGIS is open source under the GNU license most of its development is now updated by its Project Steering Committee which coordinates the general direction, release cycles, and documentation. (Admin, 2019)

PostGIS adds functions, operators, and index enhancement to a PostgreSQL database. These useful functions and operators make PostgreSQL a robust spatial database management system. These functions allow you to both queries given certain inputs and also transform the data into different outputs. Each function is overloaded meaning upon different input types, different output types can be expected. This is extremely useful. (Admin, 2019)

ST_AsText is a PostGIS method. This is function is a useful formatting tool that takes geometry as an input and returns the Well-Known Text representation of the geometry. This is used when extracting geometry from the database, this geometry is usually represented in a PostGIS ST_Geometry object. ST_GeomFromText is the inverse of ST_AsText, it allows users to construct a PostGIS ST_Geometry object from the OGC Well-known text representation. These functions will be used throughout the database interaction within the project when both inputting geometry and when using other functions to extract geometry from the NPS dataset.

Spatial reference system functions such as ST_SetSRID and ST_transform are included in the function list. These functions allow users to change the SSRID of the geometry taken from the database. This project will need to use ST_transform as both datasets are using different spatial reference systems. ST_Transform is a simple function that takes geometry and SSRID integers as inputs and transforms the geometry into this SSRID. This or ST_SetSRID will be used in the project to convert the NPS dataset which is in ESPG:4326, the most common coordinate reference system as it covers the whole globe into EPSG:27700, the system used in the VU city dataset, which just covers the UK. ST_SetSRID is like ST_Transform although it does not transform the data, it simply sets the ID column of the Geometry to the SSRID inputted

Some of the functions can help users discover the topological relationships within their database. This also allows users to perform a wide range of checks and functions upon their database. The useful topological functions for the project include ST_contains, ST_Overlaps, ST_Dwithin, ST_Covers,

ST_Equals, and maybe many more. ST_contains allows users to input two geometries, or geometry and a single point. It returns true if not a single point of geometry one lies in the exterior of geometry two and at least one point of the interior of geometry two lies in the interior of geometry one. This function could be used to query the NPS database by inputting a single point which is presumably a geo-coded address and receiving the geometry of the given site within the NPS dataset back. Examples of where ST_contains would return true can be seen in numbers 1,2, and 4 in Figure 4 below. ST_overlaps is another topological function that returns true if “two geometries intersect and have the same dimension but are not completely contained by each other” (Postgis, Unkown). This function takes two geometries as input and returns a boolean. This function could be used to find the neighboring sites of a given site. Since upon research some of the sites in the NPS dataset may have an echelon-sized gap between them, a better, more effective way of finding neighboring sites might be to use ST_Dwithin. ST_Dwithin returns all geometries within a given distance of an inputted geometry. It takes geometry and a double as input and returns a list of geometries. ST_Equals is a function which when given two geometries inputs will return a boolean value depending on the equality of the two geometries. ST_Equals still returns True if the points are ordered differently within the two geometries. This can be used in the project to check if a site has been input twice. ST_Covers returns true if no point in one inputted geometry is out another inputted geometry. It also allows you to input two geographies datatypes instead. This could be used in the project to determine the area or postcode a site or collection of sites are. Examples of situations where ST_Covers would return true are shown in numbers 1,2,3, and 4 in Figure 4.0.

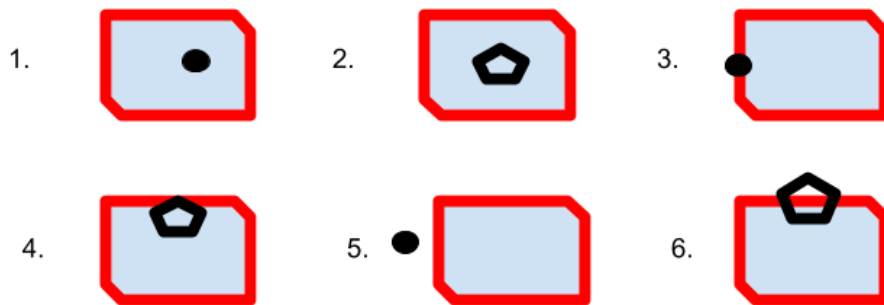


Figure 4, St_covers example

Measurement functions will also be implemented into the project. These compute measurements of distances, areas, and angles quickly. This saves time as these are standard functions needed when using geometry. It may also be less costly as these functions are written in C++ rather than python which the project is written. The measurement functions which are planned to be used in the project are ST_Area, ST_ClosestPoint, ST_shortestLine, ST_LongestLine, and ST_Distance. ST_Area is a function that takes a single geometry or geography as an input and returns the area of polygonal geometry. Importantly the values returned in the “units specified by the SRID” (Admin, Unkown). This means that if the geometry is in ESPG:4326 the area will be returned in spheroid degrees squared. To get a useable value in square meters or square feet the geometry must first be transformed into ESPG:27700 or another co-ordinate

system that only covers a small sector of the globe. ST_ClosestPoint simply takes two geometries and returns the point in geometry one which is closest to geometry two. ST_ShortestLine like ST_ClosestPoint takes two geometries except instead of returning a single point it returns the smallest 2-dimensional line imagined between the two geometries. These functions will be used in the project to determine which sites are neighboring each other and the distance between them. ST_LongestLine acts in inverse to ST_ShortestLine, returning a line which is the length of the longest line between the two geometries this might also help in determining neighboring sites. ST_Distance simply returns the “minimum 2D Cartesian distance between two geometries”. Importantly like ST_Area the value returned is “determined by the SRID” so ST_Transform will have to be used beforehand. (Admin, Unknown)

Geometry processing functions are also contained in PostGIS. These functions compute geometric constructions or alter the geometry. The project will make use of these to discover the altered geometry of given sites. The function which is planned to be implemented in the project is ST_ConvexHull, ST_Concavehull as well as ST_OffsetCurve.

ST_Concavehull discovers a “possibly concave geometry that encloses all input geometry vertices. The result is a single polygon, line or point”. The functions allow a geometry, a float, and a boolean as inputs. The float determines the target percent of the area of the convex hull the PostGIS solution will “try to approach before giving up or exiting” (Admin, Unknown). This means that a target percentage closer to 1 will give you a similar answer as the ST_ConvexHull function. The ST_Convexhull function takes geometry as an input and computes the convex hull of said geometry. The convex hull is the “smallest convex geometry that encloses all geometries in the input.”[15]. A convex hull of a set of geometries is typically used to determine the area of effect based on a set of point observations. Figure 5 shows an example of this function working. This may be used in the project to turn a collection of site geometry into a street geometry to be experimented with later.

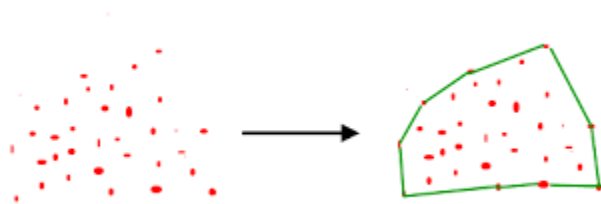


Figure 5, ST_ConvexHull example

Affine transformation functions which are used to change the position and shape of geometries might be used in the project to perform standard rotations and translations on the geometry. The project might implement ST_Translate, ST_RotateX, ST_RotateY, and ST_RotateZ. ST_Translate allows PostGIS users to input geometry and a translation and obtain a new translated geometry in return. Like other PostGIS functions, the returned data is in the same SRID as the original geometry. The three rotation functions simply compute a rotation upon an inputted geometry when given either a radian value or degree value. This might be used to normalize the orientation of the entire street.

The last set of functions that are planned to be implemented in the project are bounding box functions. These functions produce or operate on bounding boxes. They also provide geometry values by using automatic casts. The functions used from this set are ST_XMax, ST_XMin, ST_YMax, ST_YMin. They all take geometry as an input and return the max or min x or y value within the set.

Projects which use similar technologies

Blender

Blender is used across the spectrum of the tech and film industry. Blender is used in many mega-media projects which use its software to create computer-generated animation. The first professional film project to utilize blender was SpiderMan-2. Microsoft, Facebook, Unity, and NVIDIA also utilize blender in different ways. Unity is a real-time development platform. It allows users to import Blender assets into its system. This includes .blend files which is an entire blender scene in a single file. These examples prove that blender is more than capable of producing production-level models and animations.

QGIS

Many companies and other software producers and products make use of PostGIS as a database backend. CitySurf Globe is a GIS software which models raster data such as satellite images and aerial photographs.

VU.CITY

VU city is the company that supplied the second dataset used in the project, which is a relatively small area cutout of their larger dataset. VU city themselves have used the larger dataset to make models “accurate to 15cm, the VU.CITY platform covers the whole of London and the centers of other major UK and international cities. VU.CITY integrates data and demonstrates the real-life impact of buildings on their surrounding environments.” (VU-City, 2021) Cities like New York, Liverpool, Belfast, and Glasgow as well as many others are examples of cities been fully modeled by VU city. An example of some buildings rendered into London city can be seen in Figure 6 below. Members of Lanu were lucky enough to befriend some members of VU city at the “Techstars London Accelerator” (TechStars.com, 2020). This ignited the project's flame as they offered to grant a section of their data to it. Some of the key features that VU city gives to their users include city models which are constantly updated “including future development proposals”. They also allow users to import their models into VU cities larger model which can also be modified. They give users the ability to perform micro-climate analysis such as “sunlight and overshadowing” (VU-City, 2021)



Figure 6, Example of VU.City output

Projects which use VU city data

There is a large subset of architects, designers, and city planners who utilize VU.CITYS data to great extent. This proves that the data is relatively clean and very useable in a professional setting.

Farrells Architects

Farrells Architects is an example of another company that produces quality projects using VU cities data in their process. “Farrells is a team of architects whose thoughtful and cutting edge designs create beautiful places across the world” (VU.CITY, 2021). They use Vu City data to foresee how a project of theirs might impact the landscape. This tells them how to tweak their designs to suit the surrounding area. They also use “VU.CITY to communicate designs securely with clients and planners to enrich their understanding of design proposals”. These features allow Farrells architects to propose more risky designs than before as they can test the output using VU.CITYs platform.

Iceni Projects

Another company that utilizes VU.CITYS data is Iceni Projects. “Iceni Projects are a planning consultancy that helps build better ‘places to work, rest and play whilst ensuring they are timeless for future generations” (VU.CITY, 2021). Iceni uses the data to make heritage impact assessments, preliminary viewpoint studies, and support their townscape. Iceni uses the great visualization aspect of VU.Citys to data to convince councils what they are building suits the area. “Councils can see, officers can see very quickly when they see a development in VU.CITY that it makes sense to them” (Handcock, 2021)

Geocoding address

Geocoding is the process of converting a street address into geographical coordinates. Which can be used to query a dataset or place markers on a map. To do this, this project will utilize Google's Geocoding API. The API provides a direct way to access Google's geocoding algorithm directly through an HTTP request. This project will access this API through python and use its results to query the NPS database. Upon research, a common error has been discovered. This is where some sites contain two or more locations for two or more houses while some houses' geocoded location can sometimes be missing or found in a nearby street. One can see an example of this error when viewing google earth and google maps. Some houses appear to be misnumbered or have no number at all. This would seriously hinder the project as it would cause the query of the VU city database to be incorrect and return incorrect irrelevant data.

To solve this problem, this project will attempt to fix googles error using the NPS dataset. This is seen to be possible as the NPS dataset is proprietary and it's possible the reason the bug persists is due to google not having access to such site and property data. A python script will be needed. The script will keep track of each site's geometry and its neighbor's geometry and attempt to use other PostGIS functions to fill in the missing geometry. In Figure 7 below the solution can be visualized. Here the red dots represent the geocoded address, the blue squares represent the sites that were correct, to begin with, and the red sites represent those that have been fixed and put into place by the algorithm.

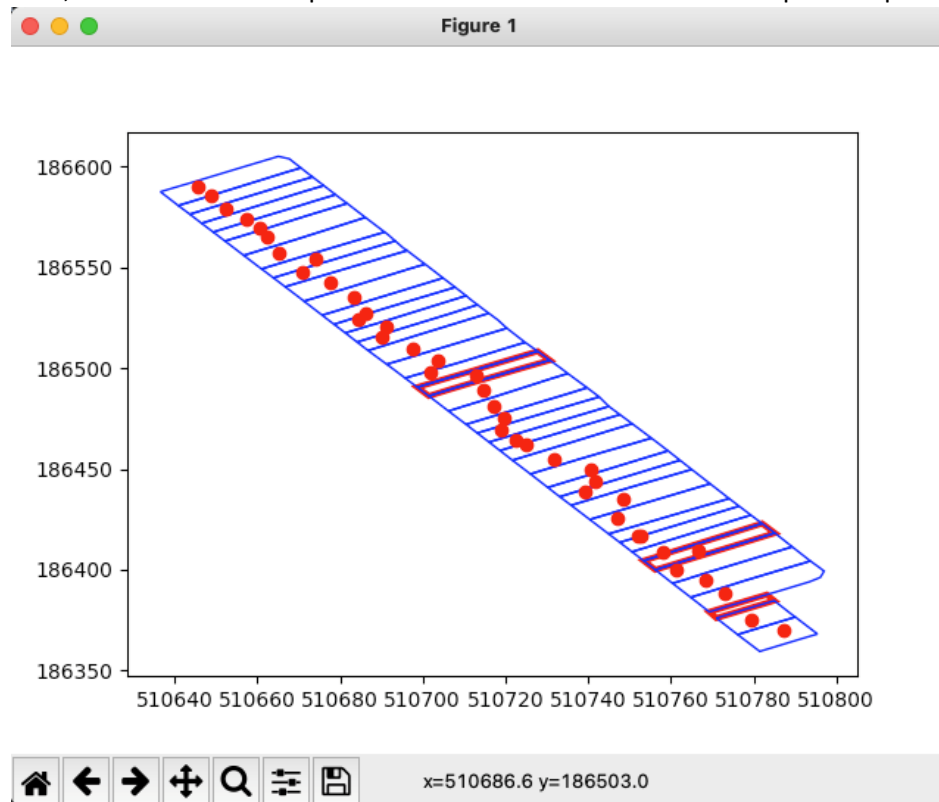


Figure 7, Solution to geocoding problem

Programming Stack

Python

Python is an interactive, interpreted, object-oriented programming language with dynamic semantics. Python often uses English keywords whereas comparable languages like Java and C++ use punctuation, this improves learnability and readability. Python supports lots of modules and packages which are easily accessible through the python interpreter to any user. Python is a high-level language; it is simple and “easy to learn syntax emphasized readability and therefore reduces the cost of program maintenance”. Python allows users to easily debug their code. When a bug is found the debugger does not cause a segmentation fault, it instead simply prints a trace stack. This allows for inspection of local and global variables and overall less time spent debugging.

A major benefit of python is its vast selection of both in-built and user-created libraries which are easy to install and free to use. This project will use at least 15 of these libraries and quite possibly more. These are crucial to the success of the project and allow the project to be completed in a timely manner.

Geopy

Geopy is one of the modules necessary for the completion of the project. Geopy is a Python client for geocoding web services. Geopy allows users to use Python to locate the coordinates of addresses, cities, countries, and landmarks. This project will make use of Geopys GoogleV3 library and API call. This API can return a location point given an address. It also returns a timezone and address given a location point. For this project, a list or array of the address will need to be turned into a location point in the same geographic coordinate system as the datasets being used. Geopy is suitable to make quick work of this problem.

Pyproj

Pyproj is another one of Python's useful packages. It is a cartographic projection and coordinate transformations library. PyProj is a Cython wrapper, meaning the library itself only provides a Python interface to interact with the PROJ.4 functionality which was originally written in C. This project needs PROJ.4 to convert data into different geographical coordinate systems. For example, the NPS dataset which comes naturally in ESPG:4326 needs to be converted to ESPG:27700 to be used to query the VU city dataset. For this reason, PyProj is extremely useful.

Numpy

Numerical Python (Numpy) is another open-source Python package. It consists of multidimensional array objects and a collection of routines and functions for processing an array in Python. NumPy allows users to perform mathematical and logical operations, Fourier transformations, shape manipulation, and linear algebra operations on arrays. NumPy will be used in the project when dealing with arrays. Arrays will be used to store multiple sets of data. This may include but is not limited to, house objects, site objects, and neighboring sites. The Linear algebra (NumPy.linalg) will also be utilized to deal with any linear algebra used in the dealing with geometry of the sites and possibly also in the convolution neural network section of the project if needed.

Re

Re is Python's regular expression package. A regular expression (Regex) is a sequence of characters that specifies a search pattern, also defined as a "string of text that allows you to create patterns that help match, locate and manage text" (Admin, 2021). Python's RE provides regular expression matching operations. It will be used in the project to quickly search through strings to find a pattern. This may be finding a postcode or house number in an address string or when searching the output of the many PostGIS functions used in the project.

Mathplotlib

Mathplotlib's Pyplot is another useful open-source tool easily accessible through Python. It is a "comprehensive library for creating static, animated, and interactive visualizations in Python". The project uses Pyplot to simply plot any graphs or diagrams needed throughout the project. One such instance of this is the need to plot the sites and geocoded addresses together. This would give good visual feedback on how that part of the system works. (python.org, 2021)

Pandas

Pandas is an extremely powerful open-source package that is built on top of the Python programming language as well as NumPy. Pandas has mainly been used as a library for data analysis. It "makes it simple to do any of the time-consuming, repetitive tasks associated with working with data" (Admin, 2021). This includes data cleansing, data fill, data normalization, data inspection and merges and joins of data. This will grant the project easily accessible functions which can be used to deal with large data sets. It will also be used to read in a CSV file containing a column which is a list of addresses, later to be used to start the geocoding process. (activestate, 2021)

SciPy

SciPy is a free and open-source library under the BSD license, SciPy is developed and maintained publicly by the community which uses it. SciPy provides fundamental mathematical algorithms for optimization, integration, and algebraic and differential equations. It also extends the previously discussed numerical functionality by adding even more tools for dealing with arrays and other data structures. This project will utilize SciPy when dealing with any linear algebra which may arise.

Pathlib

Pathlib is another Python package used in the project. It offers classes representing filesystem paths with semantics for different operating systems. Specifically, the Path module which "instantiates a concrete path for the platform the code is running on". This will allow both users of the project, me and my mentor Luke from my workplace who may need to run or modify the code on his system, to have consistent paths whilst on different systems.

Time

'time' is a built in python module which allows users to easily access various time-related functions, such functions as `gmtime()` and `localtime()` will allow the project to time its various tasks and make improvements where needed.

Pickle

Pickle is a module which "implements binary protocols for serializing and de-serializing a Python object structure". This allows the project to "pickle" its objects saving them to opened later to be "unpickled". This could be used to save time, as the output from the site finding process could be pickled allowing the data to be input into python instantly after it was run the first time. (python.org, 2021)

Plotly

Plotly is python's interactive graphing library. Its capable of producing "publication-quality graphs" such as line plots, scatter plots and importantly map-boxes. Plotly will be used in the project to create a map-graph which will be able to go to the geographic location of an inputted site and take clickable input.

(Plotly, 2021)

SKlearn

SciKit-learn (Sklearn) is a useful and robust python library for machine learning. It "provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering" and importantly also neural networks. It was built on the previously discussed NumPy, SciPy, and matplotlib. This module is vital for the project as it will allow for easily accessible convolutional neural networks functions. (tutorialspoint, 2019)

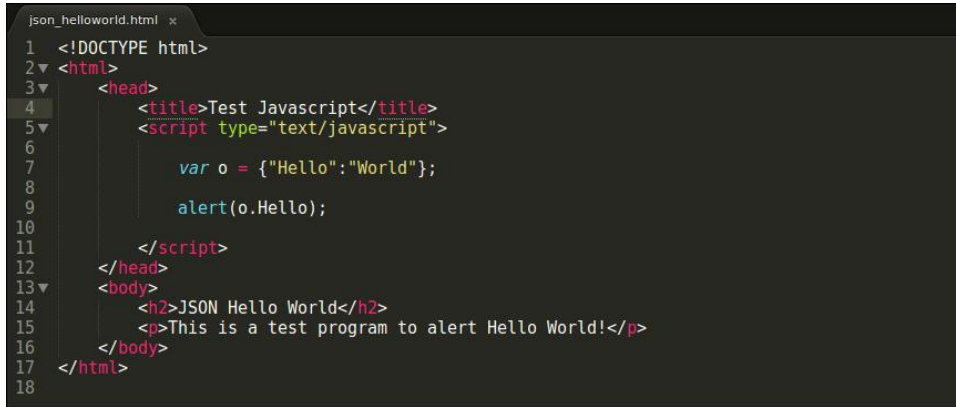
Other languages

Python isn't the only choice that could have been made as to the programming language for this project. It had fierce competition from R, Java, and Julia.

R is a popular open-source visualization-driven language that concentrates on statistical computing and rules high in the machine learning environment. R has a strong pool of resources, thanks to its prominent features that help develop machine learning applications.

Java is a versatile high-level, object-orientated, and class-based programming language that has proven successful in machine learning applications and algorithms. Java has robust frameworks such as weka and rapid miner which support machine learning algorithms such as decision trees and regression

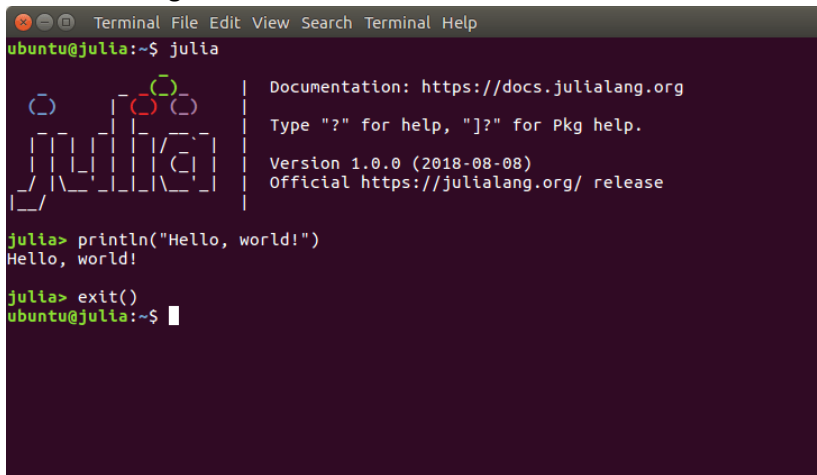
techniques. An example of a JavaScript which outputs “Hello World” can be seen in Figure 7 below



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Test Javascript</title>
5     <script type="text/javascript">
6
7       var o = {"Hello":"World"};
8
9       alert(o.Hello);
10
11     </script>
12   </head>
13   <body>
14     <h2>JSON Hello World</h2>
15     <p>This is a test program to alert Hello World!</p>
16   </body>
17 </html>
18
```

Figure 7, Hello world in JavaScript

Julia is another popular high-level programming language that was made for creative efficient model analytics. This goes especially well with machine learning applications. Julia is versatile as it executes seamlessly on all platforms. It also has a large selection of libraries like Python. Figure 8 shows a simple hello world algorithm written in Julia.



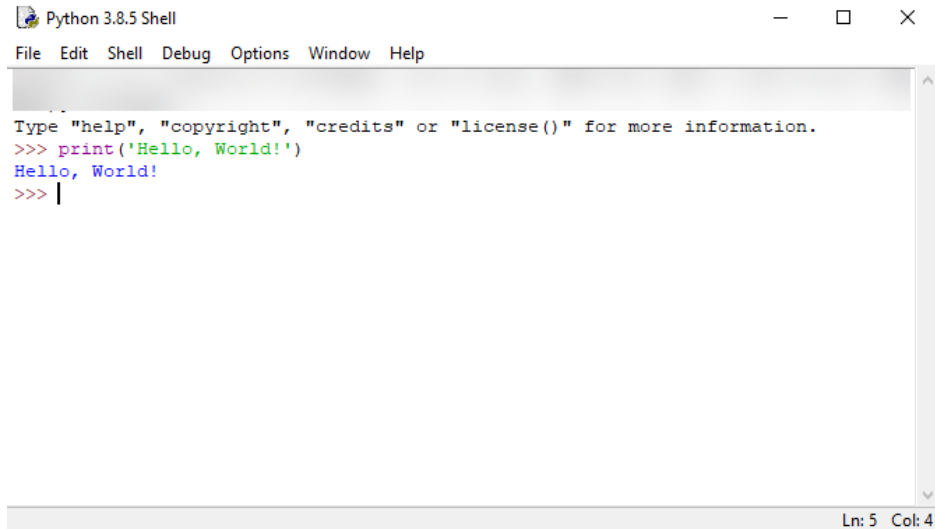
```
Terminal File Edit View Search Terminal Help
ubuntu@julia:~$ julia
Documentation: https://docs.julialang.org
Type "?" for help, "]"? for Pkg help.
Version 1.0.0 (2018-08-08)
Official https://julialang.org/ release

julia> println("Hello, world!")
Hello, world!

julia> exit()
ubuntu@julia:~$
```

Figure 8, Julia ‘Hello world’ Program

Python was chosen for this project as it offers “concise and readable code “while also offering “complex and versatile workflows “Python is also “easy to learn” and “understandable by “humans”[which is important as the creator of the project is relatively new to topics such as convolutional neural networks. Python’s hello world program can be seen in Figure 9. (Beklemysheva, 2021)

A screenshot of a Python 3.8.5 Shell window. The window title is "Python 3.8.5 Shell" and it has standard window controls (minimize, maximize, close). The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area shows the following text:

```
Type "help", "copyright", "credits" or "license()" for more information.  
>>> print('Hello, World!')  
Hello, World!  
>>> |
```

The status bar at the bottom right indicates "Ln: 5 Col: 4".

Figure 9, Python's hello world

Blender

For this project, 3D models will need to be created. Preferably automatically. These models will be models of accurate, semi-realistic houses and possibly entire streets. The 3D animation software used needs to have a scripting feature allowing a preferably python script to control the actions taken. This is due to the time it would waste if each house was to be modeled manually. Therefore, this process needs to be automated. Blender was chosen for this project mainly because of its scripting feature. This feature is perfect for the needs of the project. Scripting is not the only reason for Blender to be chosen. Blender is also open source which means it will be free to use making it easily accessible to a college student. It also means there is a strong community behind its production as designers, animators and developers come together to make Blender a more powerful tool. Blender was also used widely in Lanu the company is doing the project. My project manager at Lanu is therefore very familiar with it and can help me if needed.

Blender is a free and open-source 3D modeling software that comes under the GNU GPL license. Blender was founded in 2002 by The Blender Foundation. It still has 24 employees who work on the blender software, validate, and test Blender in working environments. Blender supports the entirety of the 3D pipeline. Including modeling, rigging, and animation. Blender also allows for python scripting to customize the application and write specialized tools. Blender is cross-platform and runs on Linux, Windows, and Mac OS computers. Blender uses an OpenGL interface. This project will use Blender 2.93

on Mac OS. In Figure 10 a screenshot of the blender user interface can be seen.

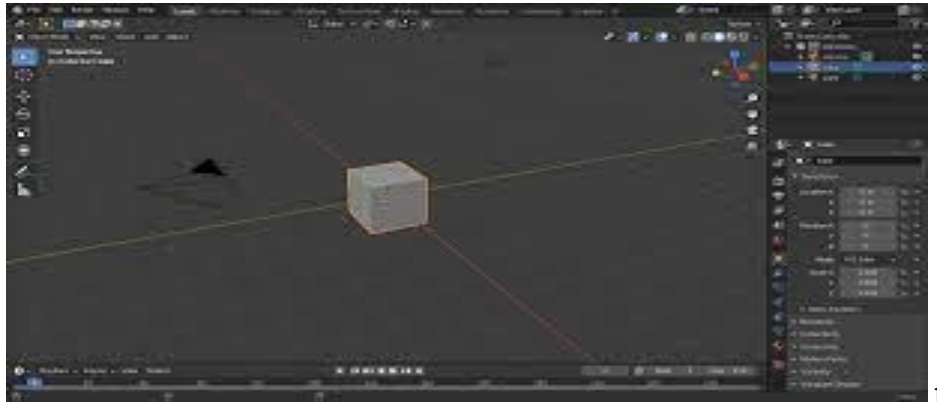


Figure 10, Blenders User Interface

Blender offers a wide range of both simple and complex features to its user base.

Blender gives users the modeling tools and abilities needed to complete creating, sculpting, editing, and transforming tasks on their models. Some of these tools include keyboard shortcuts for a fast workflow, N-Gon support, edge slide collapse and dissolve, and grid and bridge fill. This functionality makes the blender easier and more attractive to use. N-Gon is a polygon with n sides, with support for the use of these in a blender the possibilities surge. Modifiers are also included in a blender. Modifiers are automatic operations that affect an object in a non-destructive way. These tasks which would normally be tedious are completed automatically in a blender. Another feature of blender modeling is UV unwrapping. This allows users to easily unwrap your mesh right inside Blender and use custom images or textures as well as the ability to paint directly onto the model. Blender's modeling features will aid the project in the design of the house models which should be the final product of this project.

Another key feature of blender is the digital sculpting tool that is provided. "By offering the sculpting and polygonal modeling toolsets side by side, Blender greatly simplifies the transition between conceptual research and final model production" (Admin, 2021). Some examples of sculpting tools that blender users enjoy including 20 different brush types, multi-res sculpting support, mirrored sculpting, and dynamic topology sculpting. This is a dynamic tessellation sculpting method that can add or remove details on the fly.

Blender also allows users to turn their still objects into animations. Blender's animation toolset contains a character animation pos editor, non-linear animation, inverse kinematics, sound synchronization, and rigging tools. Blender also allows for simulations. These include animations like a crumbling building, rain, fire, smoke, or even the destruction of another object. The project will most likely not use any of these features.

Blender's viewport acts as its graphical user interface and is an important feature. It allows new and experienced users alike to customize their experience on blender using python or a drag and drop system. This includes splitting the viewport and adding or deleting windows when needed. Custom preferences can also be set for the user's keymap.

One of the key features of Blender and what makes it a good choice in this project is its scripting capabilities. Blender's python scripting is a versatile way to extend its functionality. It allows users to

write python code to control the actions taken in the UI. Most of the functionality of Blender can be scripted. This includes importing geometry or blend files, rendering, animation, objection creation, and destruction. Using this an entire scene can be generated automatically. To interact with blender, scripts can use its integrated API. Blenders Python API allows users to edit any data the user interface can, modify user preferences, run tools with its own settings, create new tools and also interactive tools, create new rendering engines that interact with python, and draw in Blender’s 3D viewport. Python scripts can be executed using the built-in text editor or by entering commands into the built-in Python console. From the text editor, you can also open other .py files from your system and run and edit them in blender. An example screenshot of the python scripting UI alongside the viewport can be seen in Figure 11. (Admin, 2021)

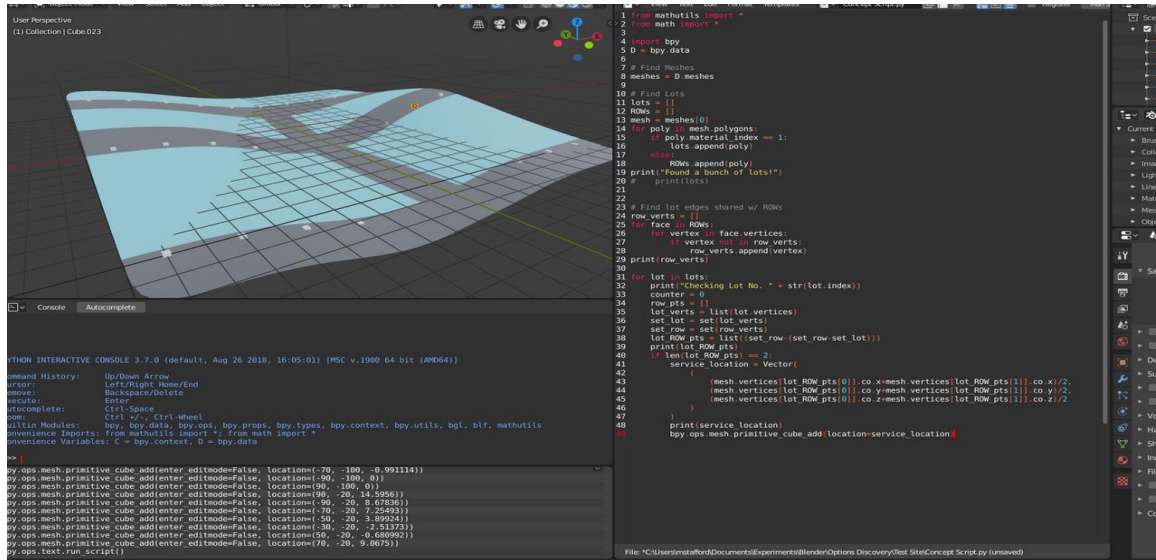


Figure 11, Blender’s viewport

Dash

“Dash apps give a point and click interface to models written in python, vastly expanding the notion of what’s possible in a traditional “dashboard”. (Plotly, 2021)

For this project dash will be used to create a prototype web application which will allow users to use the functionality of the project. Dash will provide a UI experience that will interact with back-end python and Postgres databases. The end models will also be displayed in dash once they are created in blender.

“Dash is a python framework created by plotly for creating interactive web applications. Dash is written on top of Flask, Plotly.js and React.js”. Dash is also “open source” and viewable locally on the “web browser”.

Dash is comprised of layouts and callbacks. Layouts, like CSS and html, is responsible for the content, design, or style of the website. This includes elements like graphs, input boxes and dropdowns. Dash also implements HTML components such as headings and paragraphs behind the scenes using solely python. Callbacks bring interactivity and functionality to dash applications. These python functions which carry very similar functionality to the uses of JavaScript in standard website and web applications. (Tomar, 2021).

Dash was chosen for this project due to it being easy and quick to assemble and its availability through python.

Machine Learning

Convolutional Neural Networks

This project will use a convolutional neural network to determine the features of a roof when given an image of the roof. This may be the type of ridgeline the roof has, whether the roof has a chimney, a rear or front dormer (attic extension), and rear extensions which appear in the images collected.

“A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data” (TechTarget, 2018) These use deep learning, machine vision, and NLP to perform both generative and descriptive tasks. (Al-Zawi, 2017)

Neural networks are a subset of machine learning algorithms that are at the heart of deep learning and reflect the behavior of the human brain, allowing the computer to recognize patterns and solve problems. Neural networks are comprised of layers. There is an input layer, node layer, output layer, and more hidden layers. “Each node or artificial neuron connects to another and has an associated weight and threshold. If the output of an individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along the next layer of network” (IBM-Cloud-Education, 2020). An example of this node interaction can be seen in Figure 12 Companies like Yelp, Google, Facebook, and Twitter use neural networks in their search engines, recommend systems, and other algorithms which are often criticized by the public discourse (Dickson, 2018).

A simple neural network

input layer hidden layer output layer

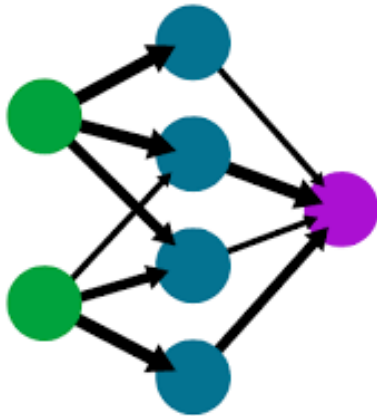


Figure 11, Simple neural network example

“Convolution is a simple mathematical operation which is fundamental to many common image processing operators. Convolution provides a way of `multiplying together two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality”

(Admin, 2003)

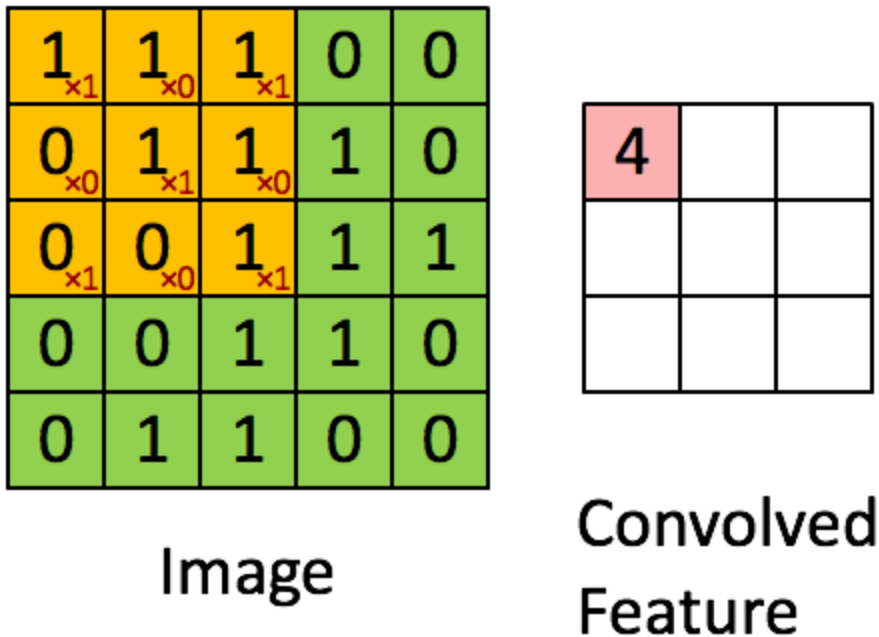


Figure 12, Convolution gif explanation

As seen in Figure 12 above the green represents our input image which has been degraded into a 5x5 matrix. We then slide the K matrix (the 3x3 matrix) over the input images matrix, creating our third matrix which is seen in red. This red matrix represents the convoluted image. An image with the same features as the input but in a 3x3 matrix instead of a 5x5. This simplified matrix saves massively on computing power. “The objective of the convolution operating is to extract high-level features such as edges, from the input image” (Admin, 2018)

This project will utilize a convolutional neural network by inputting a large dataset of images and getting a Boolean response to specific questions about the image. The images will be the Vu city height data transformed into a color-coded tile, each color on the tile will represent a cluster of similar angles on the roof. This way the convolutional neural network will learn what each color will represent. For instance, if blue was determined to be a portion of the roof that was flat, then CNN has two clear choices. It is either a chimney or a rear dormer as these are the only flat parts on a typical roof. Now the CNN must decide based on the size of the flat blue portion which box, chimney, or roof, it falls into. The same process will continue step by step until the roof and entire site have been mapped. For this, to work a form of clustering might have to take place as well.

Clustering

Clustering is a type of unsupervised machine learning used to find a meaningful structure within data.

It does this by dividing the data into groups with other data with similar data points. Separating data points based on similarity and dissimilarity. Clustering determines the intrinsic grouping among the unlabeled data points in a dataset. This is shown in Figure 1 below. The data points on this graph were put into three clear and separate clusters by a machine learning algorithm implementing clustering.

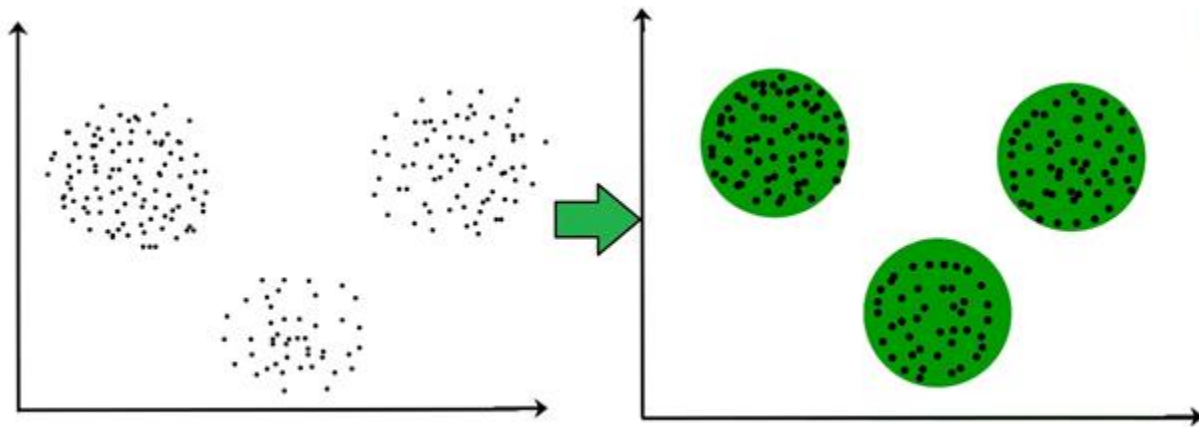


Figure 13, CNN in action

Clustering will be used in the project to quickly gather the points of height on the roof together, separating them from the other points which may have a height within the site such as the back garden shed. The process used in the project should be like the one shown in Figure 1, with two or three groups of high points. For the project, the larger set of points which will almost definitely be the house and not the shed will be selected.

K-Means

K-means clustering is one of the simpler clustering implementations. It starts with a group of randomly selected centroids, which are used for the beginning points for every cluster. It then performs iterative calculations to optimize the position of these clusters. It stops creating and optimizing clusters when the centroids have stabilized, or the defined number of iterations (k) has been achieved. (Garbade, 2018)

To determine the right number of clusters (value of k) that be best suited for the project the 'elbow method' should be implemented. This method involves plotting the distortion or inertia of the data against the number of clusters and picking the point after which the value of inertia decreases in a linear line. Figure 14 shows an example on an implementation for the Elbow method

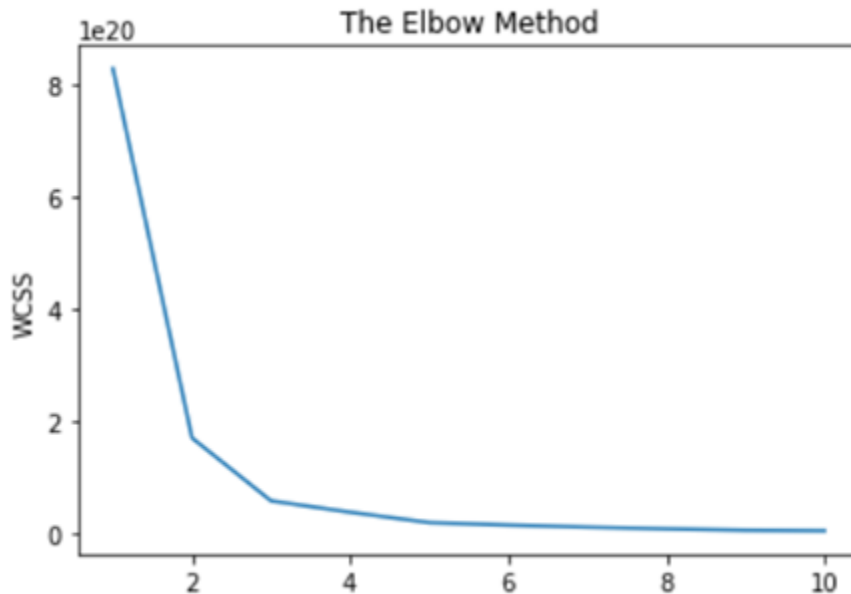


Figure 14, Elbow method example

Hierarchical Clustering

Hierarchical clustering is another method that could be used to cluster on the data. It is “an algorithm that groups similar objects into groups called clusters. The endpoint is a set of clusters, where each cluster is distinct from each other cluster, and the objects within each cluster are broadly similar to each other” (Bock, 2020). What makes the hierarchical method distinct from k means clustering is that it seeks to build a hierarchy of clusters without having a fixed number of clusters, while k-means clustering does hence the ‘k’.

To visualize how this method might cluster the data a dendrogram can be implemented. This tree-like diagram shows the relationship and Euclidean distances between the data points. An example of a dendrogram can be seen in Figure 15. The code above Figure 15 shows a python implementation of this. Here linkage refers to the distance between the sets of observation. In the example, it is set to war. This minimizes the variance of the clusters. This feature of the hierarchical method could be key in this

project's use case.

```
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method = 'ward'))
plt.title('Dendrogram')
plt.ylabel('Euclidean distances')
plt.show()
```

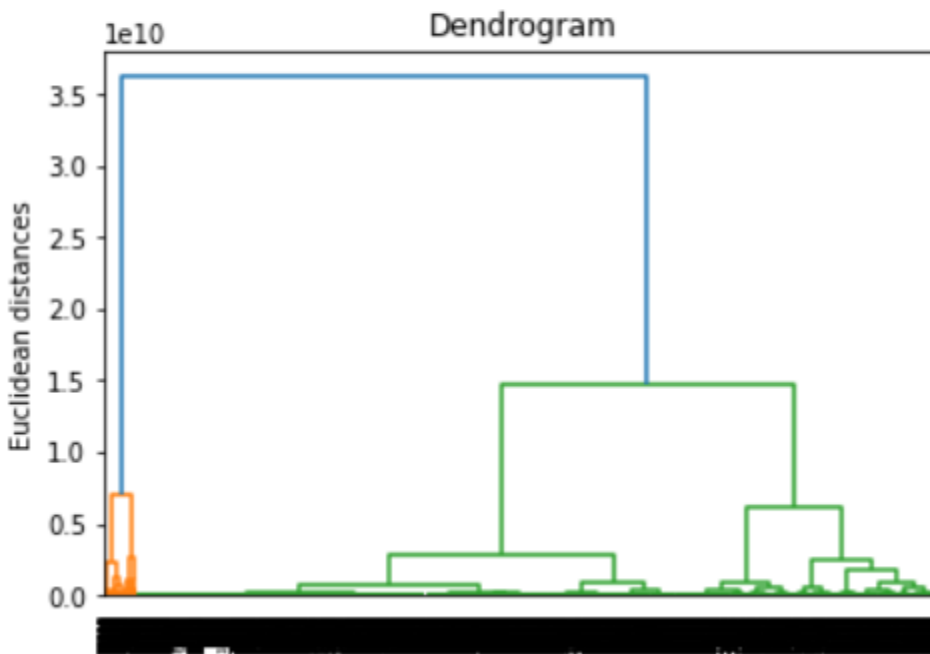


Figure 15, Dendrogram example

To conclude, both k-means and hierarchical methods will have to be tested on the data to properly discover which best suits the dataset. Although hierarchical feature set allows the user more choice of how to cluster the data and better visualization system. Another bonus is not having to have pre-determined value for the number of clusters.

(Garbade, 2018)

Bibliography

- activestate, 2021. *What Is Pandas In Python? Everything You Need To Know (Example)* - ActiveState, s.l.: s.n.
- Admin, 2003. *Convolution* <https://homepages.inf.ed.ac.uk/rbf/HIPR2/convolve.htm>, s.l.: s.n.
- Admin, 2018. *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, s.l.: s.n.
- Admin, 2019. *About PostGIS*, s.l.: s.n.
- Admin, 2019. *Chapter 1. Introduction*, s.l.: s.n.
- Admin, 2021. *Aiven for PostgreSQL*, s.l.: s.n.
- Admin, 2021. *Features*, s.l.: s.n.
- Admin, 2021. *GIS at UCD and on the Web: QGIS*, s.l.: s.n.
- Admin, 2021. *GoogleV3*, s.l.: s.n.
- Admin, 2021. *Matplotlib: Visualization with Python*, s.l.: s.n.
- Admin, 2021. *National Polygon dataset*, s.l.: s.n.
- Admin, Unkown. *ST_Area*, s.l.: s.n.
- Admin, Unkown. *ST_ConcaveHull*, s.l.: s.n.
- Admin, Unkown. *ST_Distance*, s.l.: s.n.
- Al-Zawi, S., 2017. *Understanding of a convolutional neural network*, s.l.: s.n.
- Anon., 2019. *What is PostgreSQL?*, s.l.: s.n.
- Beklemysheva, A., 2021. *Why Use Python for AI and Machine Learning?*, s.l.: s.n.
- Bock, T., 2020. *What is Hierarchical Clustering?*, s.l.: s.n.
- Dempsey, C., 2012. *Five Reasons to Start Using QGIS*, s.l.: s.n.
- Dickson, B., 2018. *The security threats of neural networks and deep learning algorithms*, s.l.: s.n.
- Docs, A. Q., 2021. *View Data*, s.l.: s.n.
- Garbade, D. M. J., 2018. *Understanding K-means Clustering in Machine Learning*, s.l.: s.n.
- Garbade, D. M. J., 2018. *Understanding K-means Clustering in Machine Learning*, s.l.: s.n.
- GISGeography, A., 2021. *27 Differences Between ArcGIS and QGIS – The Most Epic GIS Software Battle in GIS History*, s.l.: s.n.
- GOV-UK, A., 2021. *National Polygon Service*, s.l.: s.n.

Handcock, L., 2021. *iceni-projects*, s.l.: s.n.

IBM-Cloud-Education, 2020. *Neural Networks*, s.l.: s.n.

Kurt, 2012. *Cropping Shapefiles in QGIS?*, s.l.: s.n.

Menke, K., 2017. *At Least 10 Reasons You Should Be Using QGIS*, s.l.: s.n.

Plotly, 2021. *Dash Enterprise*, s.l.: s.n.

Postgis, Unkown. 5.10.2. *Distance Operators*, s.l.: s.n.

python.org, 2021. *pickle — Python object serialization*, s.l.: s.n.

PythonDocs, 2021. *Python Librarys* , s.l.: s.n.

TechStars.com, 2020. *Techstars London Accelerator*, s.l.: s.n.

TechTarget, 2018. *convolutional neural network*, s.l.: s.n.

Tomar, A., 2021. *Dash for Beginners: Create Interactive Python Dashboards*, s.l.: s.n.

tutorialspoint, 2019. *Scikit Learn Tutorial*, s.l.: s.n.

VU.CITY, 2021. *Farrells Architects*, s.l.: s.n.

VU.CITY, 2021. *Iceni Projects*, s.l.: s.n.

VU-City, 2021. *MAKE BETTER DESIGN AND PLANNING DECISIONS, FASTER*, s.l.: s.n.

