# GLA Methodology Report

# Tree Canopy & Green Cover Mapping

# June 2024

# Contents

For more information about this report, please contact:

environment@london.gov.uk

With thanks to:

Alice Giroul, GLA/TfL
Justin Yiu, TfL
John Hobbs, TfL
Yiran Wei, GLA

# 1 Introduction

## 1.1 Project Aims and Report Structure

The London Environment Strategy 2018 set out targets to increase tree canopy cover in London by 10% by 2050 and to increase green cover to at least 50% by 2050. Canopy cover was estimated at 21.06% ± 0.2% in 2018, with green cover estimated at 47.6 to 50.7% in 2019 (in both instances, underlying aerial imagery was from 2016). The London Environment Strategy Implementation Plan commits the Greater London Authority (GLA) to reporting on canopy and green cover every 5 years. This report details the methodology followed to produce the 2024 updates to the London Tree Canopy Cover and London Green Cover maps.

While other assessments of canopy cover exist[1], this project set out to produce mapped layers of tree canopy and green cover (as opposed to approaches that produce statistical assessments only) that were:

- freely available (for further use of results);
- detailed, rather than aggregated to administrative boundaries such as LSOA or wards;
- produced via methods similar to those taken in 2018 and 2019, to enable comparison of results.

This report comprises a summary of the work and results in non-technical language, followed by a more detailed methodology. The report is intended as an explainer to those who are interested in how the updates were produced; to those who will need to work with the results and thus understand the limitations of the modelling approach; and to those who may need to repeat such modelling in future.

The detailed Detailed Methodology includes: a description of the data and software used; the steps needed in this kind of modelling; and information regarding the

[1] https://www.forestresearch.gov.uk/research/i-tree-eco/uk-urban-canopy-cover/, https://friendsoftheearth.uk/nature/trees-map-where-could-we-create-woodland-england, https://insights.sustainability.google/places/ChIJdd4hrwug2EcRmSrV3Vo6llI/trees?ty=2023&hl=en-US, https://bluesky-world.com/ntm/

3 to assess the model's performance. Some technical details necessary for replicability are found in Appendices rather than the main body of the report.

## 1.2 Summary Methodology

The modelling uses three main types of data:

- normal photographic images (captured via aircraft flight).
- infrared images (as plants reflect infrared light invisible to the human eye).
- measurements of the height of surfaces within London.

Two maps of the Greater London area are created: one showing tree canopy (areas covered by the crowns of trees), and one showing areas covered by any type of vegetation (i.e., plants; trees, bushes, grass etc).

The maps are created using a computer model that is 'trained' using data manually labelled by humans. Informed by what it has learned from this manual sample, the computer repeats that learning across all of the pixels that cover London. The resulting maps are then tested against 4000 random, manually categorised, points to see how well the model performed. As the modelling approach struggles to pick up bare grass or natural surfaces due to the lack of vegetation, the green cover layer is supplemented using additional land use data. The map of tree canopy was correct in around 93% of places, and the map of vegetation (or 'green cover') correct in around 91% of places.

Using the maps, we are able to estimate that approximately 51.7% of London is covered by vegetation (including trees) and approximately 19.6% of London is covered by tree canopy. The modelling approach has limitations. With tree canopy for example, the model struggles to detect young trees or those that are not in leaf at the time images are captured. With green cover meanwhile, the model struggles to identify very bare areas of grass. Despite these limitations, the modelling provides a practical way to achieve maps of tree canopy and green cover over such a large area. It is important to remember that the resulting percentages are estimates only.

## 1.3 Method Basis and Concept

Aerial images in most cases clearly show the presence of trees or other vegetation in any given area. Green plants absorb visible light and reflect infrared light[2]; this enables calculation of the 'Normalised Difference Vegetation Index' (NDVI), a widely used metric for quantifying the health and density of vegetation. A high NDVI value indicates the presence of healthy vegetation[3]. To calculate the NDVI, infrared imagery is needed.

---

[2] https://science.nasa.gov/ems/08_nearinfraredwaves
[3] https://gisgeography.com/ndvi-normalized-difference-vegetation-index/

Another measurement that is useful to detect vegetation is height data in the form of Digital Terrain Models (DTM) and Digital Surface Models (DSM). DTM and DSM that are photogrammetrically derived work in a similar way to human vision, using photographs from at least two different vantage points to assess depth and perspective. Height data is particularly useful to distinguish trees from other types of vegetation.

We have used colour and infrared aerial imagery in combination with height data to detect vegetation, including tree canopy, across Greater London. This has advantages over on-the-ground surveying, which is time and resource intensive, making it impractical at the London-wide scale. Additionally, aerial imagery is available London-wide, un-restricted by private ownership of land.

Identification of vegetation via aerial imagery is still time consuming. To obtain estimates of percentage coverage of tree canopy and vegetation, statistical methods are often used, such as the iTree Canopy method. In that approach, a number of points are manually labelled, with the resulting data used to estimate the range of plausible total figures. This provides a numerical figure to estimate the amount of tree canopy or vegetation, but does not enable the creation of maps to indicate the location of tree canopy or vegetation. To create a map of canopy and vegetation it is necessary to use some form of automation to categorise all data points (in this case, imagery pixels). We have used 'machine learning'[4] for this purpose.

Machine learning is a form of artificial intelligence whereby algorithms are used to detect patterns in data without following explicit (human) instructions. But to reliably detect tree canopy and vegetation, the model (or algorithms) must be trained, and tested for accuracy. For our purposes, the 'Random Forest'[5] algorithm is appropriate (the name has no relation to the subject at hand) as it can be trained to classify data points into categories and has been successfully used to detect canopy in aerial imagery before. A Random Forest (RF) model is a type of *ensemble* model, meaning it relies on a number of other models to make a prediction. The RF model collates multiple predictions from several 'Decision Trees'[6] (name also unrelated to the subject at hand) to make a final prediction. A decision tree classifier (DT) is a machine learning algorithm that can classify data points (in this case, pixels) into categories by creating a set of rules for the attributes of the data points. These attributes are known as 'features'. For example, a pixel's features include its colour; brightness; and associated height. To create the rules that should be followed,

[4] What is machine learning? | Royal Society
[5] The name 'Random Forest' is not related to the detection of canopy, and this method can be used on many types of data. See https://builtin.com/data-science/random-forest-algorithm
[6] The name 'Decision Tree' is also not related to the detection of canopy.

the DT needs to be trained on data points where the categories are provided. In practice, this means we manually assign categories to a sample of pixels, so that the model can later make predictions on other pixels automatically.

Pixels are given one of three categories (or 'classes'):

1) 'Canopy': Trees, either individual or part of a wooded area.

2) 'Green': Vegetation that is <u>not</u> tree canopy[7] (shrubs, grass etc).

3) 'Neither': Everything else, including roads, buildings, and water.

Note that these categories, especially 'canopy' do not make use of definitions that are universal. Rather, the model, and the way it is trained, defines what is considered 'canopy'. Using aerial imagery and height data it is not always possible to distinguish between trees and large shrubs. The model *usually* categorises any vegetation surpassing 4m in height as 'canopy'. In practice, large shrubs may have similar benefits to 'true' tree canopy while smaller trees may not provide the benefits associated with tree canopy.

Testing is done by manually labelling 4000 random sample points across London. Points were labelled by looking at the underlying aerial imagery; infrared imagery; and height data to make an informed decision. Then, the human label of a point can be compared to the model's classification of a point to see if the model was correct. These 4000 points were also used to obtain a separate estimate of tree canopy and vegetation cover (using the iTree method) as an additional comparison to the results of the model.

---

[7] While the machine learning model needs to distinguish these non-canopy aeras from trees, later the vegetation map will include canopy as well as we will combine both canopy and 'green' areas.

# 2 Detailed Methodology

## 2.1 Data & Software

This work makes use of the following data:

- Aerial colour imagery at 25cm resolution (meaning each pixel corresponds to a square of 25cm by 25cm on the ground).
- Colour Infrared (CIR) imagery at 50cm resolution.
- Digital Terrain Model (DTM) height data at 5m resolution.
- Digital Surface Model (DSM) height data at 2m resolution.
- The Ordnance Survey MasterMap (OSMM) Topography Layer[8]

The aerial data (colour, CIR, DTM and DSM) are provided by Bluesky International Ltd[9]. Colour and CIR imagery is in in JPEG format; height data (DTM and DSM) are TIFF files; the OSMM Topography Layer is in vector format. Excluding OSMM, the data is organised by British National Grid 1km$^2$ tiles. Each image (whether in JPEG or TIFF format) corresponds to one tile. Greater London is made up of around 1600 tiles[10], but for each tile location there are multiple images of each type corresponding to different measurement dates (i.e., dates the images were captured). There is no single date for which all of London is fully photographed[11]; instead, data is captured on a rolling basis, with all of London photographed every three years. As such, the data used in this update (the most recent data available at the point the modelling work began) is as below, with the bulk from 2022, but some tiles from 2021 and 2020 (this includes colour images, CIR images, and height data for each Ordnance Survey National Grid (OSGB) tile). Data was captured between April and October, meaning most trees should have leaves on the imagery.

*Table 1 – Distribution of data used.*

| Year | Month | Number of Tiles | % of overall data |
|---|---|---|---|

[8] https://www.ordnancesurvey.co.uk/products/os-mastermap-topography-layer the version used in this project was dated 19/02/2024
[9] https://bluesky-world.com/ (note that Bluesky have their own tree mapping product that details the location, height, and crown of individual trees - https://bluesky-world.com/ntm/ - as such it may be of interest and use to those managing tree stock)
[10] The number of tiles referenced in Table 1 add up to 1741. This is because tile boundaries do not match the Greater London boundary, so to cover all the area you need tile data that extends beyond its boundary in some instances.
[11] This may be due to time or weather limitations, i.e., a single aircraft cannot fly over all of London in a single interval of good lighting and no cloud cover.

| | | | |
|---|---|---|---|
| 2020 | April | 20 | 1 |
| 2021 | April | 51 | 3 |
| | May | 140 | 8 |
| | June | 20 | 1 |
| 2022 | April | 1510 | 87 |

This type of modelling requires extensive use of a coding language such as Python, more detail about what ‘packages’ were used can be found in Appendix 1. The project also requires image editing software (to label the training data) and use of GIS software such as ArcGIS to facilitate labelling of the test data.

## 2.2 Process Steps

The following flowcharts illustrate the steps needed to produce tree canopy and green cover maps using the computer modelling approach. The first shows the overall process, the second shows how a machine learning model is trained.

*Figure 1: Overall Process*

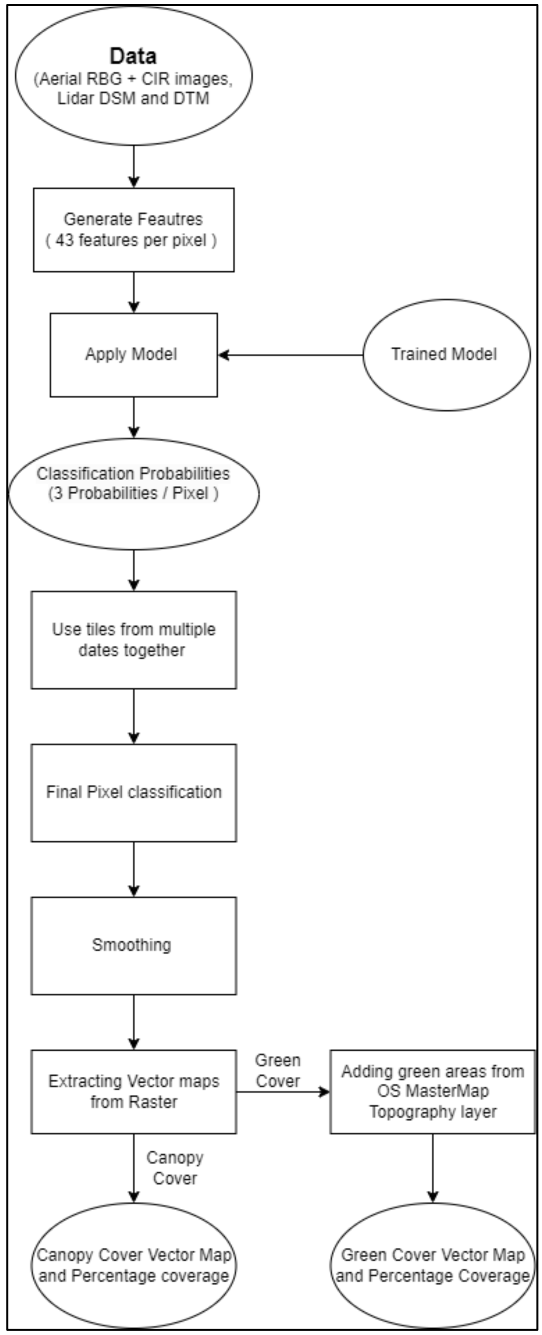
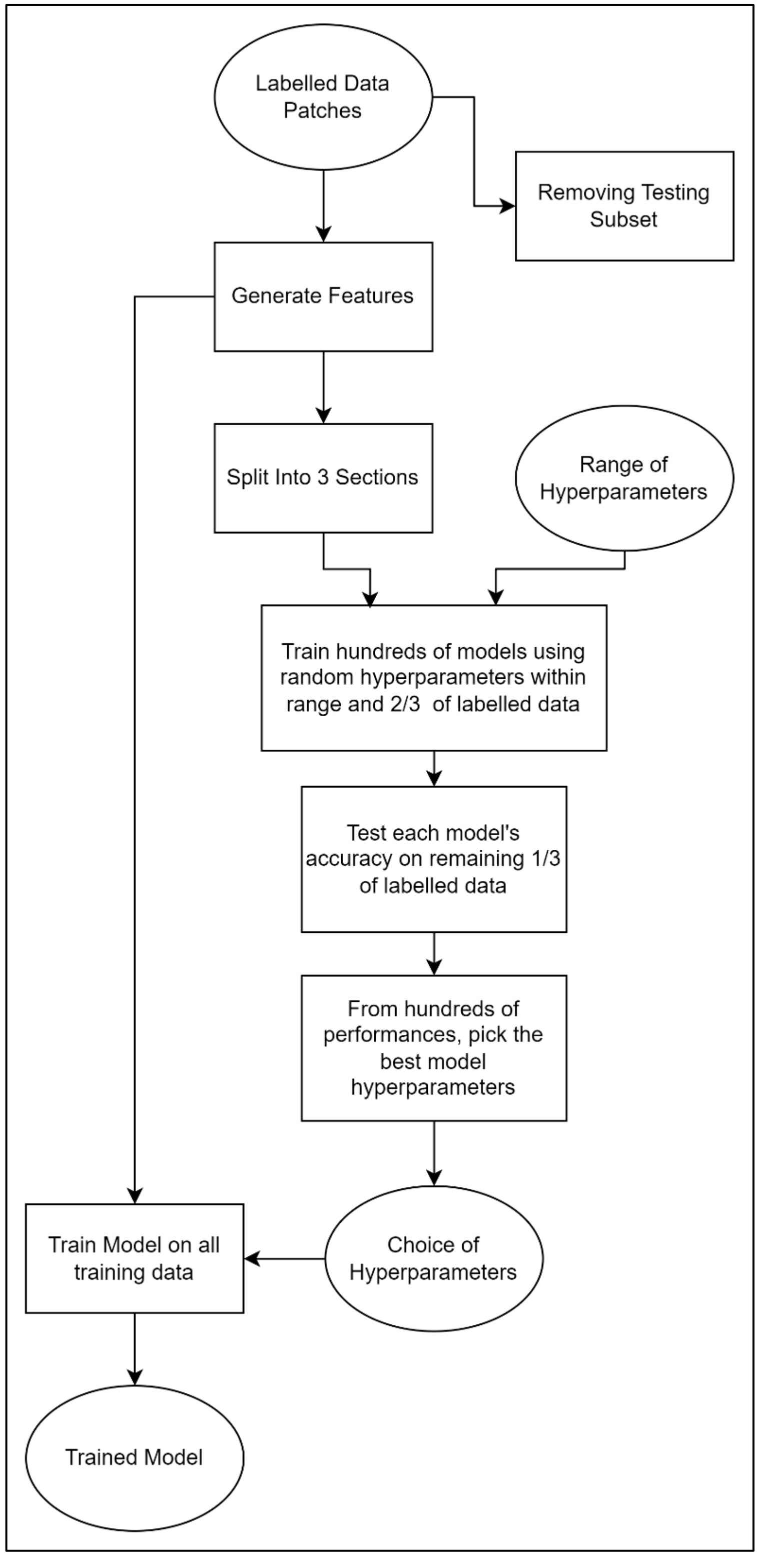


*Figure 2: Training a 'machine learning' model to automatically categorise pixels*



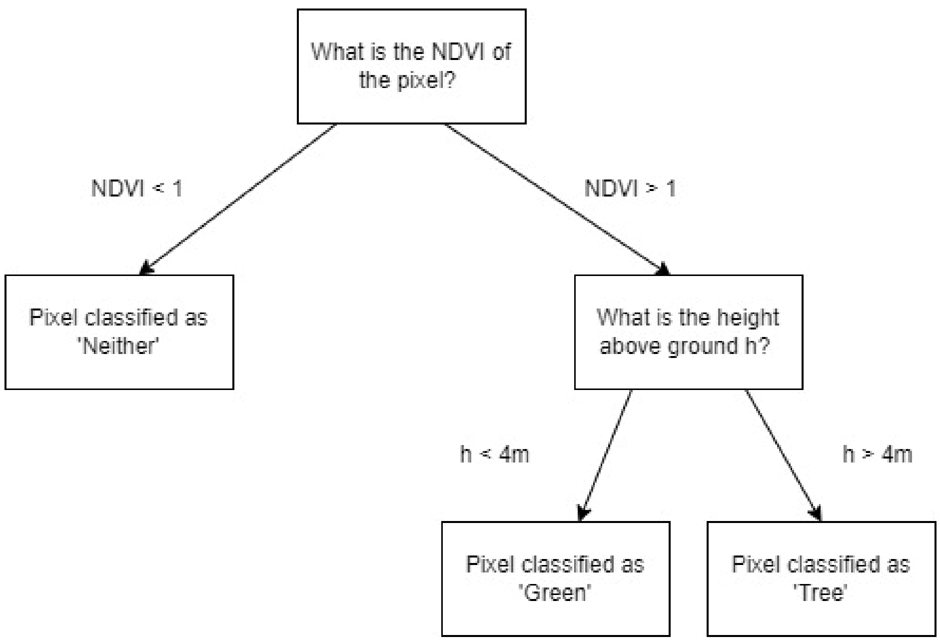
## 2.3 Pixel Classification using Machine Learning

### 2.3.1 General Concept

The classification of pixels is done via a machine learning algorithm called a Random Forest (RF) classifier. An RF classifier is a commonly used tool for which many tutorials and references exist, but a short description of how it works is as follows.

An RF model is an 'ensemble' model, where the predictions of many models are used together. The RF combines multiple Decision Tree[12] (DT) classifiers, with each DT having a 'vote' on the outcome of the classification. A Decision Tree is also a machine learning algorithm, which can be used individually, but combining the results of multiple DTs gives better results. The simplest way to think of a DT model is as a flowchart: looking at a single item to classify (in this case, a pixel), it asks a set of yes/no questions. A simplified example of what that might look like is given below.

*Figure 3: Simplified example of a Decision Tree (DT)*




The Decision Tree splits the data at each 'node' (or question), ending in a set of 'leaves' (or results) – in this example there are two nodes and three possible results.

In reality, a DT would be more complex, and is created by a DT classifier algorithm (rather than a human), based on training data that is labelled manually. With training data, the true classification of a pixel is already known, the DT classifier algorithm uses that

[12] https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991

information to try to create DT's that will have the most coherent results, whereby pixels ending up in one classification or another are similar, with as few as possible outliers. Once the DT is created, it can be used on any new pixel to determine its classification (with some accuracy).

A Random Forest algorithm creates multiple DTs. To ensure the DTs are all different, only *some* features of the pixels are given to each DT algorithm (see section 2.3.4 for an explanation of 'features'). For example, some DTs will be created without knowing the colour value of a pixel or its height. This ensures a more thorough use of the data as it forces the model to use all the information available. It also stops the model being overly reliant on limited aspects of the data (such as the height), which can cause 'overfitting'. 'Overfitting' is when a model is overly specialised to the data it was trained on but does not generalise well for new data.

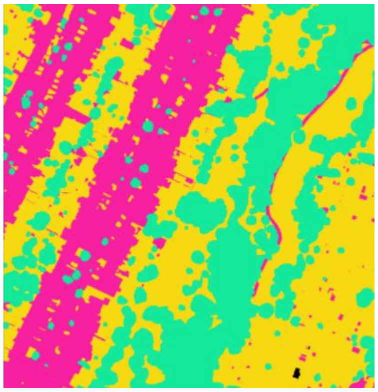
Each DT is given one 'vote' towards the final classification of a pixel. For example, if in 100 DTs, 80 classify the pixel as 'Tree'; 5 as 'Neither'; and 15 as 'Green', the pixel will be classified as 'Tree'.

### 2.3.2 Labelling

To train and test the model, a sample of pixels are manually classified (or labelled). The labelled data is split (for training or testing, but not both) so that the test data is different from the training data (the model is obviously likely to perform well on the data it was trained with). The labelled data should form a representative sample of the overall set of pixels.

To do this, we selected 16 sections of tiles of roughly 1000×1000 pixels (~250x250m on the ground) that featured all three possible classes ('Canopy', 'Green', or 'Neither'). We chose tiles from heavily urbanised and suburban locations, and ones with different capture dates. Referring to each image of an identified tile (RGB, CIR and height, both DSM-DTM), the labeller uses image editing software to colour-code a new, labelled, tile.

*Fig 4 – Labelled data, for training and testing*



In this example of labelled data, green is ‘canopy’, yellow is ‘other vegetation’, and pink is ‘neither’.

Note that this approach to labelling differs from the approach followed during the 2018 tree canopy modelling by Curio, where randomly sampled, individual points were labelled. There are advantages and disadvantages of either approach. Labelling a section allows for the quick labelling of many points, and provides a way to estimate error margins of the resulting estimates (see *3.3 Cover* Estimate Error Margins , p.25)[13]. However, since our labelled data is not randomly sampled, and nearby points are not independent from each other, it is not as well suited for use as testing data. To mitigate this, final testing of the model was performed using different, randomly sampled data (see *3.2 Map Accuracy* Testing, p.22).

### 2.3.3 Splitting Labelled Data

The labelled data is split into training and testing datasets. The testing set is used during the development of the model, and to calculate error margins, while separate, randomly selected labelled points are used to test the model’s performance (see Assessing Accuracy).

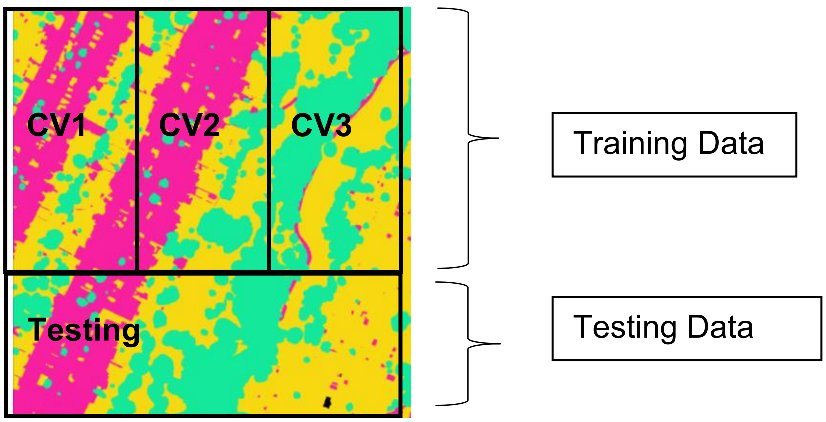
To be able to test the model’s performance *while* it is being trained, the training set is additionally split in to three ‘cross validation’ (CV) sets. The model is trained using two of the three cross validation sets and tested on the third, with the model’s predictions for the third CV set compared to the manually labelled data. This can be done in 3 variations, i.e., train using set ‘a’ and ‘b’ but test on ‘c’, or train using ‘b’ and ‘c’ but test on ‘a’ (etc). The performance of the model is assessed by averaging how well it performed across the

[13] An additional advantage is that the labelling of complete sections makes the labelled data usable by other types of model (e.g. ‘computer vision’ models which learn to interpret entire images at a time). We have not made use of that approach in this update, but it’s use could be explored in future.

three variations (this is the 'Cross Validation' process) and can be compared to the performance of *other* models (or rather, the same model tuned differently).

*In this project, 30% of the pixels are retained for testing, with the remaining 70% used for training and split into 3 CV sets. This is a typical ratio used for testing/training data.*

*Figure 5: Labelled data split in to training and testing sets.*




A few rows of pixels between the different sets are discarded[14], to minimise the chance that pixels of the same feature (e.g., the same tree) are present in multiple sets. This helps to reduce the risk of 'overfitting'.

### 2.3.4 Feature Generation

The model is trained to classify each pixel (Canopy, Green, or Neither) according to its attributes, or 'features'. Features inherent to the source data include things like the RGB (red, green, blue) value in the colour image; the value of the infrared band in the CIR image; or the height value in the DSM and DTM variations of the same pixel. To obtain the best possible classification for each pixel, additional features are created and assigned to each pixel. Together with the original pixel attributes, this will constitute the full set of features the model uses to the classify the pixel.

The full list of features can be found in Appendix 2, there are 43 overall. Examples of additional features include:

- Alternative representations of the pixel's colour, and its greyscale value.
- NDVI (Normalised Difference Vegetation Index) value (see p.5)

---

[14] 20 rows are discarded between the training and testing areas, and 10 rows discarded between the CV groups

- Height above ground, calculated as the difference between DSM and DTM. (This should be the height of objects above terrain level which is useful in distinguishing trees)
- Height changes *around* the pixel (such as minimum and maximum difference in DSM height between a pixel and its neighbours).
- Average values around the pixel, e.g., average NDVI, Hue, Saturation etc.
- Measures of image texture around the pixel. There are ways to measure the 'texture' around a pixel, such as with the Gray-Level Co-Occurrence Matrix[15]. This can inform the model on how 'smooth' or 'rough' the image patch around the pixel is.

The additional features were chosen by considering what could help distinguish vegetation and canopy from other things seen in aerial imagery, such as buildings, roads etc. We endeavoured to use a similar set of features to those used by Curio in the 2018 analysis, minus the height data we have additionally used in this update. It is possible to investigate how useful the individual features used in the modelling are and use that information to either: remove features that are less useful; or add more features that are similar to the useful ones (this is done using a software package called 'scikit-learn'[16]).

Once a set of features is chosen, their values can be computed for each pixel. Note that this stage is computationally intensive (i.e., it takes a computer a long time to execute) and requires a large amount of storage to save the resulting '3-dimensional arrays'[17] (3-D arrays enable the storage of 43 values per pixel). The features of the labelled data are computed and stored, but when the model is running classification of new pixels, the values of each feature are generated, used immediately for pixel classification, but not stored.

### 2.3.5 Hyperparameter Tuning

To build the model, 'hyperparameters' need to be provided. Hyperparameters are simply 'settings' for the model. When using a Random Forest classifier, one such hyperparameter is the number of Decision Trees (DT) to be used. Another is the number of features each DT is able to use. Another hyperparameter might tell the model when to stop splitting the data further into smaller groups (this can be done by setting a maximum 'depth' for each DT, the example DT illustrated by Fig.3 on page 12 has a depth of 3). The full list of hyperparameters, as well as the ranges tested for each, can be found in Appendix 3.

---

[15] GLCM Texture Features — skimage 0.22.0 documentation (scikit-image.org).
[16] Feature importances with a forest of trees — scikit-learn 1.4.1 documentation
[17] https://www.educative.io/answers/what-is-a-3-d-array

To determine which hyperparameters to use, we train multiple versions of the model with a range of different hyperparameter values and compare results using the three cross validation (CV) groups the labelled training data is split into (see 2.3.3 *2.3.3 Splitting Labelled* Data). Each potential model is tested three times, by being trained on two of the CV groups and tested on the other[18]. To test the model, the predictions for the third CV group are compared to the labelled data, resulting in a value for how many pixels were predicted correctly, expressed as a fraction. We keep only the models that have high accuracies, but within those we pick a model with the *lowest* DT depth and number of features – this is because a simpler model is less likely to be 'overfitted' (see p.13). In this project, the final model has a maximum DT depth of 11; uses 7 features (out of 43) for each DT; and uses 97 DTs (see Appendix 3 for all hyperparameters).

Now that hyperparameters have been chosen, the three cross-validation (CV) sets can be abandoned, and a final version of the model is trained using a random sub-sample of *all* available training data[19]. At this stage, the model can be saved to file, and applied to any new data.

### 2.3.6 Model and Outputs

Once a model is saved, it can technically be applied to any aerial image (and corresponding infrared / height data) to obtain a modelled version of that imagery. Of course, while it *can* be applied to any data of the same format (aerial image, infrared image, height data, and extracted/additional features from all these) the model will only perform well if the source data is sufficiently similar to the data it was trained on. So, if an aerial image is from another region or time, the model may not perform well.

Two examples of the raw model outputs are shown below. In these examples 'Canopy' is shown in white; 'Green' is shown in grey; and 'Neither' is shown in black.

---

[18] To ensure the model is performing well across all classes (canopy, green, neither), the 'balanced accuracy' can also be computed. The balanced accuracy is the average of the 'producer accuracy' for each class (see 3.2 *223.2 Map Accuracy* Testing for a definition of producer accuracy). Balanced accuracy can be automatically computed using scikit-learn, a free and open-source machine learning library for the Python programming language.

[19] This random sub-sample comprises 5 million pixels. Using *all* the available training pixels was not possible with the computer memory available.

*Figure 6: TQ5786 (Upminster, in Havering). Image taken August 2018*[20].

Image © Bluesky International, 2024

*Figure 7: TQ1984 (Stonebridge, in Brent). Image taken April 2022*

Image © Bluesky International, 2024

These images show that the model performs as expected. Patches of woodland, street trees, and private gardens are identified.

[20] The example image is outside the date range used in this update only because TfL did additional / internal-use modelling.

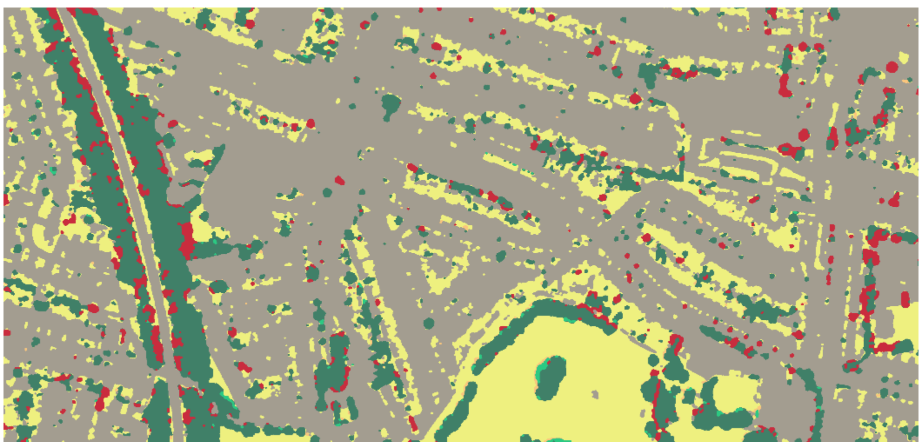
*Figure* shows that fields that are bare are not identified as ‘green’, this issue is addressed by a later stage (see section 2.4.3).

## 2.4 Post-Processing

### 2.4.1 Smoothing the outputs

As the model classifies individual pixels, the output layer can be 'noisy' (or, appear erratic) with many pixels being classified differently from their neighbours, in a manner that is unlikely to be true to the situation on the ground. To remove these outlier, or 'lonely' pixels, and make the resulting map easier to host and make use of, the results are 'smoothed'. This is done via 'morphological image processing'[21] using OpenCV[22]. The areas depicting tree canopy or green cover are successively shrunk and grown. In this way they return to their intended size except that very small patches that are shrunk to 'nothing' disappear, which is the intended result. After this step a smooth map layer is created. An example of the 'noisy' output, before smoothing, is shown below. Note that while this step could in theory decrease the map's accuracy, all the testing (see *3.2 Map Accuracy Testing*, p.22) is performed on the final, smoothed map.

*Figure 2: Model output before 'smoothing' (many very tiny patches of pixels)*



### 2.4.2 Extracting outputs and converting to shapefile

So far, the output has been in 'raster' format (images made up of pixels). Raster format is necessary for statistical analysis (i.e., to work out the accuracy of the results) but a 'vector' format (or 'shapefile') is needed for the next stage (see 2.4.3 below). Two maps are extracted using an algorithm from the Rasterio[23] software package (the extraction is possible because the images were already georeferenced, i.e., each pixel was mapped to a location via a relevant coordinate system). The two extracted maps (tree canopy cover, and green cover which includes canopy) are saved in vector/shapefile format meaning they are now stored as shapes (or polygons) rather than images. Using vector

---

[21] https://www.educative.io/answers/what-is-morphological-image-processing
[22] https://docs.opencv.org/3.4/db/df6/tutorial_erosion_dilatation.html
[23] https://rasterio.readthedocs.io/en/latest/api/rasterio.features.html#rasterio.features.shapes

maps it is possible to identify the canopy cover and green cover percentage of any other shape, such as Borough or Ward.

### 2.4.3 Improving the Green Cover Map

As illustrated by Figure 6 (p.17) the modelling approach does not do well at categorising areas of dry grass, or bare field, as vegetation. (The lack of greenery means there is less infra-red light reflected for the model to detect). This was a known issue in the 2019 modelling of green cover[24], overcome in a similar manner to the approach taken in this update, though a different data source was used[25].

The problem is addressed by supplementing the automatically detected green areas with additional mapping products. This is especially important for agricultural land, which should be counted as green cover but are not always 'green' due to the cycle of planting and harvest. In this update, categories from the Ordnance Survey MasterMap (OSMM) Topography Layer[26] deemed likely to identify land comprising vegetation were added to the final green cover layer. The OSMM Topography Layer is updated every six weeks[27].

The full list of categories extracted from the OSMM Topography Layer can be found in Appendix 4. A sample of the polygons corresponding to the chosen categories were inspected alongside aerial imagery to ensure they identified green areas.

This stage raises the question of why the OSMM Topography Layer (or other similar products) cannot replace the whole analysis of green cover. The issue is that such datasets do not typically map private gardens or many other very small areas of green cover.

---

24 How Green is London – Report and Methodology, 2019

25 The 2019 release made use of *Verisk GeoInformation Group UK Map*, but use of OSMM data is preferred given the GLA's ongoing licence

26 https://www.ordnancesurvey.co.uk/products/os-mastermap-topography-layer

27 Version used in this project was dated 19/02/2024

# 3 Assessing Accuracy

The modelling approach to assessing tree canopy and green cover generates two types of output, the mapped layers, and the percentage figure for each type of cover. Both outputs are estimates, so it is important to conduct some analysis of how accurate those estimates are likely to be. The way we assess accuracy is different for each of the two types of output. For map accuracy, the research question is 'how many points on the map are correct', for accuracy of the cover estimates, the question is 'how far outside this estimate is the true answer likely to be' (or, what are the 'margins of error'). This section of the report looks at each in turn.

## 3.1 Obtaining Additional Testing Data

The labelled data used to train the model (see section 2.3.2) has some drawbacks as a way to test the accuracy of the map. The training data was not randomly sampled, and as sections of image (rather than individual pixels) were labelled, many of the labelled pixels have similar features. Some of the originally labelled data was used to test the model at intermediate steps[28], but to test the final maps as objectively as possible, we created new labelled data comprising 4000 randomly selected pixels. These form a representative sample and additionally allowed us to obtain an independent estimate of tree canopy and green cover using the iTree Canopy[29] tool (see section 3.4 below)

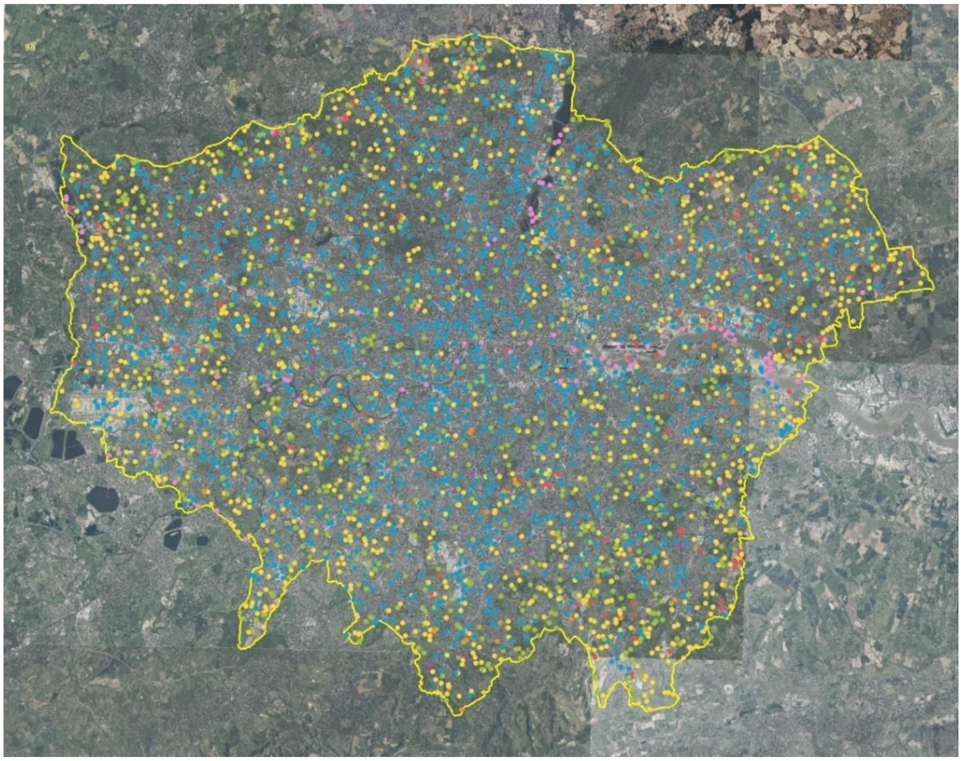
Appendix 6 gives more information about how the labelling interface was set up. Labellers were able to look at colour aerial imagery; infrared imagery; and height data to help classify a point (the same information the model worked with). The following classifications were given as options:

- Tree
- Vegetation that is not a Tree (used for any other plant including grass, shrubs, flowers, hedges)
- Vegetation but unsure whether tree or other
- Bare ground (used for areas of soil without vegetation or buildings / manmade surfaces)
- Manmade (non-natural surfaces, including buildings, roads etc)
- Water (rivers, lakes, ponds etc)
- Unsure (this option was only used where the labeller could not determine what the surface was, for example if the point is completely in shadow in the available imagery)

---

[28] E.g. during *2.3.5 Hyperparameter* Tuning (p.13) and to check that the post-processing (p.16) of the map did not impact accuracy negatively.
[29] [i-Tree Canopy (itreetools.org)](https://canopy.itreetools.org)

*Figure 3: 4000 randomly selected points, labelled manually. (Colours indicate how they have been labelled)*



Labellers were encouraged to use the height layer to determine whether a point was a tree or other type of vegetation. As a guideline it was indicated that trees should be at least 4 meters high unless there were other ways of identifying. (Individual / street trees are often easier to identify than the elements that make up dense green cover).

## 3.2 Map Accuracy Testing

To assess the accuracy of the two cover maps, the modelled classification of each point was considered alongside the manual classification (i.e., what the model thinks compared to what the human thinks).

*Table 1: Manual classification of 4000 points compared to modelled classification.*

| Number of Points | Percentage of total points | Manual Classification | Modelled Classification |
|---|---|---|---|
| 1644 | 41.1 | Manmade | Neither |
| 959 | 24 | Vegetation (not Tree) | Green |
| 694 | 17.4 | Tree | Tree |
| 135 | 3.4 | Vegetation (not Tree) | Neither |
| 91 | 2.3 | Manmade | Green |
| 90 | 2.3 | Water | Neither |
| 81 | 2 | Vegetation (not Tree) | Tree |
| 62 | 1.6 | Bare Ground | Green |
| 49 | 1.2 | Tree | Green |
| 37 | 0.9 | Bare Ground | Neither |
| 34 | 0.9 | Tree | Neither |
| 26 | 0.7 | Manmade | Tree |
| 26 | 0.7 | Unsure | Neither |
| 24 | 0.6 | Vegetation (unsure if Tree) | Green |
| 18 | 0.5 | Vegetation (unsure if Tree) | Tree |
| 10 | 0.3 | Unsure | Green |
| 6 | 0.2 | Unsure | Tree |
| 8 | 0.2 | Vegetation (unsure if Tree) | Neither |
| 5 | 0.1 | Water | Green |
| 1 | 0 | Bare Ground | Tree |

In the table above, green indicates that the modelled prediction matches the manual classification; red indicates the modelled prediction does *not* match the manual classification; yellow indicates unknown whether they match (as the human labeller has chosen 'unsure'). Obviously, there is always a chance of human error, but the manual classification is taken to be true. ('Green' is taken to be correct when matched with a 'Bare Ground' manual label as we want the green cover layer to incorporate such areas)

From this, we can see that most points are classified correctly by the model (green rows), and calculate as a fraction, the accuracy of the tree canopy and green cover maps. The approach raises the question of how to account for pixels that have been manually classified as 'Unsure' or 'Vegetation (unsure if Tree)'. One option is to count them all as modelled incorrectly, but that is unlikely to be the case (especially in the instances where the model has classified as 'green' or 'tree', and the labeller has also classified as vegetation, just unsure which type). It was decided to calculate the accuracy in two ways, a 'worst case scenario', and a 'realistic scenario'. In the latter, it is assumed that *half* the pixels classified as 'Unsure' are modelled correctly in the green cover map, and half the pixels classified as 'Vegetation (unsure if Tree)' are modelled correctly in the canopy cover map. The resulting accuracies can be found in Table 2 below ('overall' columns, 5 and 6). For the realistic scenario the Canopy Map is found to have an accuracy of 94% and the Green Cover Map an accuracy of 91% (figures in bold).

Two complementary and common ways of assessing the performance of a classification map are known as the 'consumer' (or 'user') accuracy and the 'producer' accuracy[30]. The consumer accuracy looks at it from the point of view of the user of the map and answers the question 'If a point is mapped as a tree, what is the probability that it really is a tree?'[31]. The producer accuracy looks at it from the point of view of the creator of the map and answers the question 'If a point is *actually* a tree, what is the probability that it gets mapped as a tree?'. In this project, 'actually a tree' is known via the manual (human) labelling of points, taken to be the 'ground truth'.

*Table 2: Overall accuracy (realistic and worst-case); consumer/user accuracy; and producer accuracy of each map*

| Map | Option | Consumer Accuracy | Producer Accuracy | Overall (worst case scenario) | Overall (realistic scenario) |
|---|---|---|---|---|---|
| **Canopy Cover** | Tree | 0.84 | 0.89 | 0.93 | 0.94 |
| | Not Tree | 0.95 | 0.97 | | |
| **Green Cover** | Vegetation | 0.93 | 0.90 | 0.91 | 0.91 |
| | Not Vegetation | 0.88 | 0.93 | | |

These accuracy results are comparable to the modelling of canopy cover and green cover produced in 2018/19 and confirm that the model performs well. A final step in assessing the accuracy of the maps, is to check that the accuracy does not vary too much across

30 https://gsp.humboldt.edu/olm/Courses/GSP_216/lessons/accuracy/metrics.html
31 Or 'If a point is mapped as *not* a tree, what is the probability that it is really not a tree?'. This alternative version of a similar (but not identical) question is why there are two options under the consumer and producer accuracy of each map.

London. To do this we have calculated the accuracies for each borough, which can be found in Appendix 5. While the accuracy does vary, it remains good across all boroughs.

## 3.3 Cover Estimate Error Margins

In addition to estimating the accuracy of the maps, it is useful to obtain margins of error for the estimates of overall canopy and green cover, both for the whole Greater London area and for each borough. Errors in the model can compensate for each other when looking at sufficiently large areas (vegetation might be missed in some locations, but mistakenly detected in others). This means that the margin of error tends to decrease the larger the area.

To assess how accurate the cover estimates are, we need to look at the modelled results for areas where we know what the *true* results are. In this case, we can look at the labelled data described in section 2.3.2 of this report. The 'testing set' of this labelled data, reserved specifically for testing purposes, has not been used to train the model and is therefore a fair way to assess the accuracy of the cover estimates. We can compare the modelled vs labelled canopy cover and green cover of each patch.

First, the error (in percentage points) is calculated for each patch. This is the difference in coverage between the labelled patch and the modelled patch. For example, if the labelled patch has a green cover of 45%, but the modelled patch has a green cover of 39%, the error is -6% (39 minus 45). Then, the error is calculated for any two given patches added together, then three, and so on. There were seven available patches of labelled data. This approach allows us to identify the errors across different sizes of area, which is important given the issue noted above (larger areas have lower margins of error) and is what enables us to estimate the margin of error for each borough.

Once we know the true errors for a range of different areas (and size of areas), we look at the variation (or 'deviation') of those errors and use that information to calculate the Standard Error (SE)[32]. Standard error is commonly used in statistics and gives an indication of how close a true value is likely to be to an estimated value. Under the 'empirical rule' (or '68-95-99.7 rule', which looks at normal distribution), 68% of the data falls within one standard error. (Or, the true value has a 68% chance of being within one standard error of the estimate). This is why standard error is expressed as a plus or minus (±).

[32] https://www.investopedia.com/terms/s/standard-error.asp

The SE represents a 68% 'confidence interval', higher confidence intervals can be obtained by multiplying by the relevant factor[33]. Use of the 68% confidence interval matches the approach taken by Curio in the 2018 modelling of tree canopy cover,

Here are the standard errors for some example areas of interest:

| Area Type | Canopy Cover Typical SE (%) | Green Cover Typical SE (%) |
|---|---|---|
| Greater London | 0.30 | 1.11 |
| Average size borough | 0.34 | 1.13 |
| Average size ward | 0.56 | 1.23 |

*Table 3: Standard errors for London, average sized borough, and average sized ward.*

The results section of this report (4.1) gives the standard error of each estimate, including borough breakdown.

It is important to remember that just like the model and cover estimates themselves, the calculations of error margins and accuracy are also estimates.

## 3.4 Additional iTree Estimates

The i-Tree canopy[34] method uses an equation to estimate the canopy cover of a specified area using randomly sampled, manually labelled points. As we had 4000 randomly selected and manually labelled points from the testing of the map accuracy (explained above, in section 3.2) we were also able to use them to obtain separate estimates of London-wide canopy and green cover. This is not a direct assessment of the modelled / mapped layer or its outputs (cover estimates), rather it is an entirely independent estimate, to enable comparison with the modelled estimate.

The i-Tree method relies on the fact that the more points sampled, the less likely it becomes that their population is far from the real underlying distribution (e.g., the real canopy or green cover of London).

[33] E.g. for a 95% confidence interval, the standard error should be multiplied by 1.96

[34] See References i-Tree Help (itreetools.org). iTree have their own point-labelling interface and aerial imagery, however the method works with any accurately labelled, randomly sampled points. This link also contains the formula used for the standard error.

The most important part of the iTree canopy method is the calculation of error margins. The standard error (SE) is given by the formula:

$$SE = \sqrt{\frac{pq}{N}}$$

N = total number of sampled points

p = proportion of points classified as tree/vegetation

q = proportion of points *not* classified as tree/vegetation

The 4000 labelled points were distributed as follows:

| Label Classification | Raw Percentage |
|---|---|
| Bare Ground | 2.5% |
| Manmade | 44.0% |
| Tree | 19.4% |
| Unsure | 1.1% |
| Vegetation (not Tree) | 29.4% |
| Vegetation (unsure whether Tree) | 1.3% |
| Water | 2.4% |
| Total | 100% |

*Table 4: Distribution of classifications (manual labelling)*

To calculate green cover via this method, we combined points labelled 'Bare Ground'[35], 'Tree', 'Vegetation (not Tree) and 'Vegetation (unsure if Tree)'. Points labelled 'Unsure' were assumed to have the same proportion of vegetation to the rest of the sample, so we factored a proportion in to obtain a green cover estimate of 53.1±0.8%. For canopy cover, we included points labelled 'Tree' plus a fraction of the points labelled 'Vegetation (unsure if Tree)' (for this it was assumed that the proportion of trees to vegetation remained the same as in the overall sample). This resulted in a Tree cover estimate of 19.9±0.6%. In each case, the standard error equation was used, and the margins of error shown are one standard deviation, corresponding to a 68% confidence interval.

| iTree Results | Greater London |
|---|---|
| Canopy Cover (iTree) | 19.9 ± 0.6 % |
| Green Cover (iTree) | 53.1 ± 0.8 % |

[35] For completeness, the calculation of vegetation cover *excluding* Bare Ground is 50.6% ±0.8 (this includes 'unsure' points)

# 4 Results

## 4.1 Tree Canopy and Green Cover

From the modelled maps an estimate of the percentage cover of canopy and vegetation can be extracted. London-wide and borough estimates are included here, ward estimates are available via the London Datastore[36] and on the corresponding web map. The standard error (SE) is given against each result, see section 3.3 for further explanation.

| Results | Greater London |
|---|---|
| Canopy Cover | 19.56 ± 0.30 % |
| Green Cover | 51.68 ± 1.11 % |

| Borough | Area (ha) | Canopy Area (ha) | Canopy Cover | Canopy Cover SE | Vegetation Area (ha) | Green Cover | Green Cover SE |
|---|---|---|---|---|---|---|---|
| Barking and Dagenham | 3780.8 | 311.4 | 8.24% | 0.36% | 1336.3 | 35.30% | 1.14% |
| Barnet | 8678.2 | 2093.6 | 24.12% | 0.33% | 4892.6 | 56.40% | 1.13% |
| Bexley | 6430.0 | 1040.6 | 16.18% | 0.34% | 3153.1 | 49.00% | 1.13% |
| Brent | 4325.1 | 661.0 | 15.28% | 0.35% | 1597.8 | 36.90% | 1.14% |
| Bromley | 15017.1 | 4075.0 | 27.14% | 0.32% | 11059.8 | 73.60% | 1.13% |
| Camden | 2179.7 | 618.3 | 28.36% | 0.38% | 887.3 | 40.70% | 1.15% |
| City of London | 315.0 | 12.7 | 4.03% | 0.52% | 18.8 | 6.00% | 1.22% |
| Croydon | 8652.3 | 1796.4 | 20.76% | 0.33% | 4419.0 | 51.10% | 1.13% |
| Ealing | 5556.9 | 1048.3 | 18.86% | 0.34% | 2496.5 | 44.90% | 1.14% |
| Enfield | 8222.7 | 1406.5 | 17.11% | 0.33% | 4478.2 | 54.50% | 1.13% |
| Greenwich | 5045.5 | 1040.2 | 20.62% | 0.35% | 2403.1 | 47.60% | 1.14% |

[36] https://data.london.gov.uk/dataset/green-cover-2024 & https://data.london.gov.uk/dataset/canopy-cover-2024

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Hackney | 1905.5 | 343.2 | 18.01% | 0.38% | 650.8 | 34.20% | 1.15% |
| Hammersmith and Fulham | 1716.1 | 258.4 | 15.06% | 0.39% | 500.6 | 29.20% | 1.16% |
| Haringey | 2960.8 | 589.3 | 19.90% | 0.36% | 1143.3 | 38.60% | 1.14% |
| Harrow | 5048.7 | 1074.8 | 21.29% | 0.35% | 2758.7 | 54.60% | 1.14% |
| Havering | 11447.8 | 1610.9 | 14.07% | 0.33% | 7134.6 | 62.30% | 1.13% |
| Hillingdon | 11575.6 | 2314.6 | 20.00% | 0.33% | 6628.9 | 57.30% | 1.13% |
| Hounslow | 5661.1 | 1012.4 | 17.88% | 0.34% | 2800.1 | 49.50% | 1.13% |
| Islington | 1486.2 | 318.2 | 21.41% | 0.39% | 429.5 | 28.90% | 1.16% |
| Kensington and Chelsea | 1238.9 | 288.2 | 23.26% | 0.40% | 388.5 | 31.40% | 1.16% |
| Kingston upon Thames | 3727.7 | 773.6 | 20.75% | 0.36% | 1950.5 | 52.30% | 1.14% |
| Lambeth | 2725.9 | 480.8 | 17.64% | 0.37% | 847.7 | 31.10% | 1.15% |
| Lewisham | 3532.8 | 692.6 | 19.60% | 0.36% | 1379.9 | 39.10% | 1.14% |
| Merton | 3763.9 | 747.4 | 19.86% | 0.36% | 1720.6 | 45.70% | 1.14% |
| Newham | 3858.8 | 448.6 | 11.62% | 0.35% | 1162.0 | 30.10% | 1.14% |
| Redbridge | 5645.6 | 928.7 | 16.45% | 0.34% | 2837.1 | 50.30% | 1.13% |
| Richmond upon Thames | 5878.6 | 1605.7 | 27.31% | 0.34% | 3817.8 | 64.90% | 1.13% |
| Southwark | 2992.3 | 647.4 | 21.63% | 0.36% | 1123.8 | 37.60% | 1.14% |
| Sutton | 4386.3 | 668.1 | 15.23% | 0.35% | 1973.6 | 45.00% | 1.14% |
| Tower Hamlets | 2158.2 | 317.6 | 14.72% | 0.38% | 543.2 | 25.20% | 1.15% |
| Waltham Forest | 3881.9 | 720.9 | 18.57% | 0.35% | 1573.4 | 40.50% | 1.14% |
| Wandsworth | 3523.4 | 787.5 | 22.35% | 0.36% | 1366.1 | 38.80% | 1.14% |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Westminster | 2203.8 | 472.4 | 21.43% | 0.38% | 713.4 | 32.40% | 1.15% |

*Table 5: Results from the Canopy and Vegetation maps for all the London boroughs. Standard errors (SE) are provided for the area of each borough and for each map.*

## 4.2 Limitations and Successes

While section 3 of this report gives statistical assessments of the accuracy of this update, it is worth exploring some of the specific limitations and successes of the machine-learning approach to estimating tree canopy and green cover. The statistical assessments presented in section 3 are conducted on the final maps, and so are inclusive of the issues explained here.

### 4.2.1 Detection of Bare Ground

This issue is explained in section 2.4.3 as it necessitated an additional stage in the process.

### 4.2.2 Detection of Young Trees

We know that the model does not typically detect young and newly planted trees as canopy. Rather than being a limitation, this should be seen as a success as the purpose of the modelling is to detect true canopy cover, rather than trees. Young trees do not contribute to canopy cover, and its benefits, in the same way established and mature trees do. While the management of London's overall tree stock (including new planting) within a 5-year interval may lead to canopy increase in the longer term, it is understandable that the impacts of young trees / new planting will not be detected within the same timeframe.

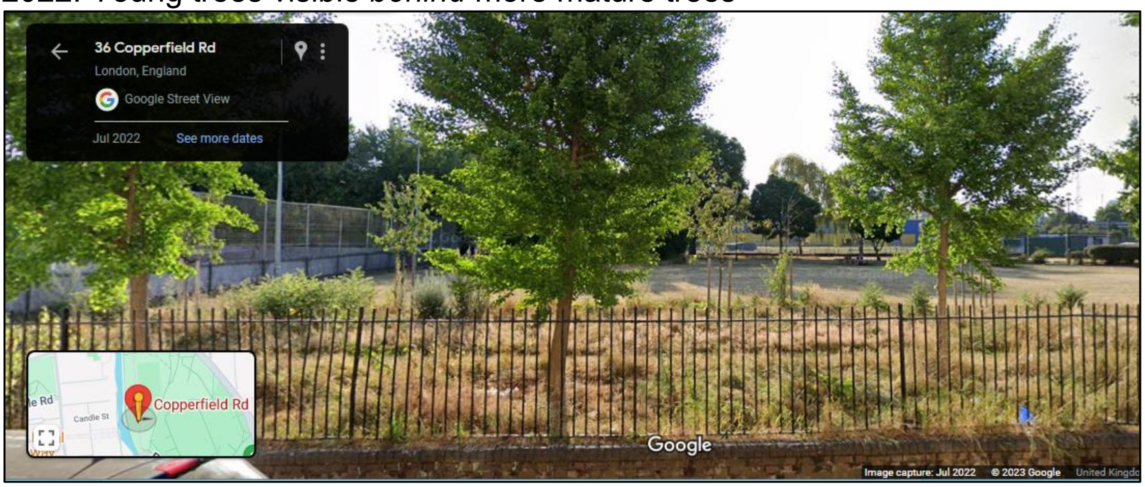
There are multiple reasons why the modelling does not detect young trees as canopy:

1) The model was trained to detect canopy at around 4m in height, young or recently planted trees are usually below this height. (New planting often consists of 'Whips' that can be less than 1m in height)
2) Even if a young tree had a height above 4 metres, it would not necessarily be picked up by the model. The resolution of height data is relatively low, so a tree taller than 4m would need to cover the bulk of a 2m x 2m area to be picked up.
3) Young trees have fewer branches and are not as leafy as mature trees. This means they emit less infrared and have a smaller NDVI signature (a key component in the model's ability to make classifications).

Example – Mile End Park

2018: trees recently planted

2022: Young trees visible *behind* more mature trees



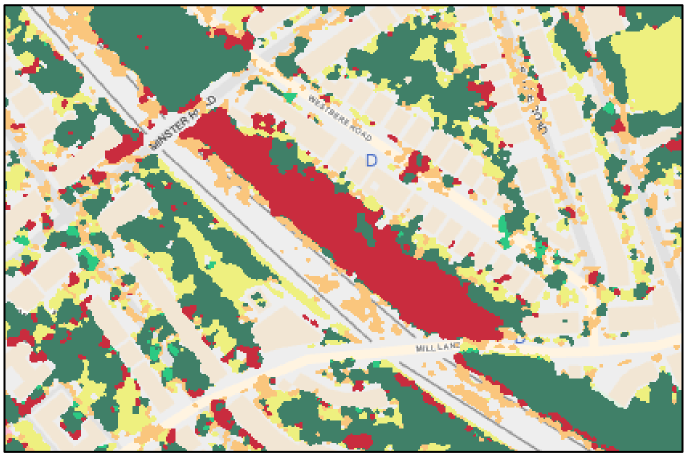

2022: Almost impossible to see in aerial imagery (left) and not detected by model (right)

### 4.2.3 Detection of Lost Canopy

We know that the modelling successfully identified instances of lost canopy cover, and that in other instances, it indicated lost canopy cover when trees are still present but have (for example) been pollarded. Some examples are included here.

Any indications of change as result of this work (either in the mapping, or the cover estimates) should be investigated further (for example by reviewing aerial or street imagery across a range of dates, or via additional use of the iTree sampling methodology) bearing in mind all outputs are estimates only.

<u>1) Railway-adjacent land in West Hampstead</u> (51.551840, -0.204460)

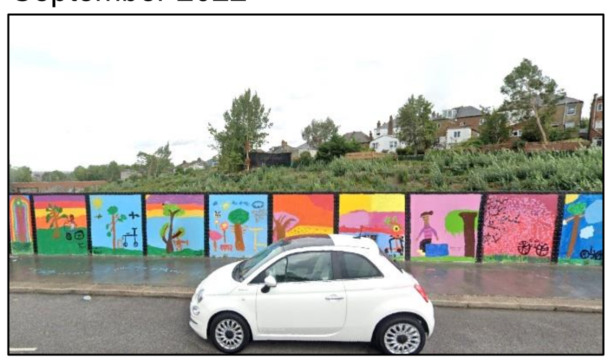



The modelled map shows a patch of missing canopy (red) next to railway tracks.

Google Street View images below show that trees have been cut back either entirely (with new shrub growing) or significantly.

August 2021

September 2022



## 4.3 Comparison to previous results

As a means of comparing this update to the estimates produced in 2018, borough level canopy cover results were compared, and instances where the estimates differed by more than 6% were investigated further. This process was not exhaustive, and as mentioned elsewhere in this report, any instances of apparent significant change should be validated further. The overall finding of this update is no statistically significant change in canopy cover across London within the timeframe (2016 to 2022).

Over-estimation of 2018 modelling

The modelling has picked up instances where the 2018 work appears to have over-estimated canopy cover as result of detecting general vegetation as tree canopy (see examples below). The key difference in the new update is the use of height data which we think explains why the new model has classified such areas differently, and why the overall canopy cover estimate appears slightly lower (19.56 ± 0.30% compared to 21.06 ±0.2%, though we do not consider this an indication of true change).

Note that while we have examples of this issue, we have no practical way of analysing its overall impact on results.

- Barking (Beam Valley Country Park) – continuous area modelled as tree canopy (green in first image) in 2018 (based on 2016 data), whereas in aerial imagery from 2015 and 2017 (e.g., second image) it appears to be a mix of trees, shrub, and grass. New model (third image) makes more distinction, but potentially under-estimates.

- Harrow (Mead Wood) – as above

# Appendices

## Appendix 1 - Python Packages (Software)

Modelling and testing via Python[37] (a coding language) made use of the following 'packages':

- OpenCV (cv2) for image processing, reading and writing;
- Scikit-image (for feature generation);
- Numpy, pandas;
- Scikit-learn (sklearn) for the Random Forest classifier;
- Pyspark for parallel processing of feature generation and model hyperparameter tuning;
- Shapely, Fiona and geopandas for geospatial analysis;
- Rasterio for counting pixels in specific parts of rasters;
- and joblib, folium, matplotlib

[37] https://www.python.org/

# Appendix 2 - Data Features

This is the full list of features computed for each pixel and given to the model to classify it. They are computed using the OpenCV and Scikit-Image code packages.

1. Blue
2. Green
3. Red
4. Blue (CIR)
5. Green (CIR)
6. NIR (Near Infrared band from the CIR image)
7. Hue – with cv2 cv2Color (see [OpenCV: Color conversions])
8. Saturation – cv2Color
9. Value – cv2Color
10. CIR_Hue – cv2Color computed HSV with Blue and Green from the CIR and NIR instead of Red
11. CIR_Saturation – as 11
12. CIR_Value – as 11
13. Flipped Hue – Hue is a circular quantity, flipping it by 180 shows that. This is necessary because the DTs cannot take that into account itself. *Flippped_hue = Hue+127, hue_f[hue_f>255] = hue_f[hue_f>255]-255*
$$Flipped\ Hue = Hue + 127\ if\ Hue \leq 128$$
$$Flipped\ Hue = Hue - 128\ if\ Hue > 128$$
14. Greyscale – cv2Color COLOR_RGB2GRAY
15. NDVI $NDVI = \frac{NIR-Red}{NIR+Red}$
16. NDVI numerator by value $NDVInbV = \frac{NIR-Red}{Value}$
    This was used in addition to the NDVI to avoid large values in the NDVI when the denominator approaches 0.
17. Smooth NDVI – nr 15 smoothed with cv2.medianBlur and kernel size 5 (see [OpenCV: Image Filtering])
18. Same with kernel size 3
19. Kernel size 7
20. Note: DSM and DTM are **not** included raw into the model
    DSM Mindiff cutoff – we compute the smallest difference between a pixel and its 4 neighbours. Then, we set a cut off to that value at 2m. The aim is to differentiate particularly flat areas (e.g. roads, roofs) vs natural surfaces.
21. DSM Maxdiff – same but the largest difference, and no cut off
22. Height above ground : DTM-DSM

23. Function of max height change in disk(2) footprint of the height above ground (22 above)
24. Entropy of NDVI (scikit-image entropy function (skimage.filters.rank) with footprint = disk(5), see skimage.filters.rank — skimage 0.22.0 documentation (scikit-image.org))
25. Same as 24 with disk(20)
26. Same as 24 with disk(27)
27. Entropy of Hue with disk(10)
28. NDVI neighbourhood minimum. Minimum NDVI in disk(5) using skimage.filters.rank.minimum(), see skimage.filters.rank — skimage 0.22.0 documentation (scikit-image.org)
29. Same with disk(10)
30. Same with disk(20)
31. Smooth NDVI (17) neighbourhood minimum with disk(10)
32. Saturation (8) neighbourhood median with disk(5)
33. Hue (7) neighbourhood median with disk(5)
34. Gray Level Co-Occurrence Matrix (GLCM) Dissimilarity for the NIR (6) for a 25x25 pixel patch, angle = 0 and distance = 2
35. Same as above but with 10x10pixel patch , distance = 1
36. GLCM Correlation for the NIR for a 25x25 pixel patch, angle = 0 and distance = 2
37. Same as 36 but for the NDVI (15)
38. GLCM Dissimilarity for the NDVI (15) for a 25x25 pixel patch, angle = 0 and distance = 2
39. Neighbourhood Minimum of GLCM Dissimilarity for the NDVI (15) for a 10x10 pixel patch, angle = 0 and distance = 2
40. Neighbourhood Median of GLCM Dissimilarity for the NDVI (15) for a 10x10 pixel patch, angle = 0 and distance = 2
41. Blurred blue band (1) – using cv2.blur and a 9x9 pixel patch (OpenCV: Image Filtering)
42. Same for green (2)
43. Same for red (3)

# Appendix 3 - Hyperparameters Tested

Random Forest classifiers involve several possible hyperparameters, or settings[38]. In this project, five were specified:

- Number of DTs: The number of Decision Trees used to form the Random Forest. Each DT gets one 'vote' towards a prediction.

- Maximum DT Depth: The maximum depth of each DT (maximum number of successive nodes).

- Minimum Result Size: The minimum number of pixels required to form a possible result (i.e., if the data will be split into smaller sections than this, the model stops splitting the data).

- Maximum Samples: The maximum fraction of samples (pixels) used to train each DT. In our case, this is almost 1, so all samples are used each time.

- Maximum Features: The maximum number of features (out of all the features available) used to train each DT.

For each hyperparameter, we tested a range of possible values before settling on a final choice for the model:

| Hyperparameter | Range tested | Final choice |
|---|---|---|
| Number of DTs | 30-100 | 97 |
| Maximum Tree Depth | 6-18 | 11 |
| Minimum Result Size | 50-1000 | 84 |
| Maximum Samples | 0.3-1.0 | 0.99 |
| Maximum Features | 3-20 | 7 |

In the testing of hyperparameters, a large number (1200) of random combinations are used. This has advantages over specifying the exact combinations of hyperparameters to test (known as a grid search)[39].

[38] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#
[39] https://scikit-learn.org/stable/modules/grid_search.html# (RandomizedSearchCV)

# Appendix 4 - OS MasterMap Topography Categories

The table below contains the list of all categories included as part of the green cover mapping. How and why this additional data was used is explained in the main body of the report (p.21). The version of the OSMM Topography Layer used was dated 19/02/2024.

| DescGroup | DescTerm | style_description |
|---|---|---|
| Natural Environment | Nonconiferous Trees (Scattered),Rough Grassland,Scrub | Nonconiferous Tree Fill |
| Natural Environment | Heath,Scrub | Scrub Fill |
| Natural Environment | Mineral Workings (Inactive),Scrub | Scrub Fill |
| Natural Environment | Rough Grassland | Rough Grassland Fill |
| Natural Environment | Coniferous Trees,Nonconiferous Trees | Mixed Woodland Fill |
| General Surface | Spoil Heap (Inactive) | Natural Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Rough Grassland | Nonconiferous Tree Fill |
| Natural Environment | Rough Grassland,Scrub | Scrub Fill |
| Natural Environment | Coniferous Trees (Scattered) | Coniferous Tree Fill |
| Natural Environment | Marsh,Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Coniferous Trees (Scattered) | Mixed Woodland Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Scrub,Coniferous Trees (Scattered) | Mixed Woodland Fill |
| Natural Environment | Nonconiferous Trees,Scrub | Nonconiferous Tree Fill |

| | | |
|---|---|---|
| Natural Environment | Nonconiferous Trees,Coppice Or Osiers | Nonconiferous Tree Fill |
| Natural Environment | Coniferous Trees (Scattered),Rough Grassland,Scrub | Coniferous Tree Fill |
| Natural Environment | Nonconiferous Trees,Scrub,Coniferous Trees | Mixed Woodland Fill |
| Natural Environment | Coppice Or Osiers | Coppice Or Osiers Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Heath | Nonconiferous Tree Fill |
| Natural Environment | Marsh,Scrub | Scrub Fill |
| General Surface | Spreads | Natural Fill |
| Natural Environment | Heath,Nonconiferous Trees (Scattered) | Nonconiferous Tree Fill |
| Natural Environment | Heath,Nonconiferous Trees (Scattered),Scrub | Nonconiferous Tree Fill |
| Natural Environment | Heath,Nonconiferous Trees (Scattered),Rough Grassland,Scrub | Nonconiferous Tree Fill |
| General Surface | | Natural Fill |
| Natural Environment | Nonconiferous Trees,Coniferous Trees,Scrub | Mixed Woodland Fill |
| Natural Environment | Marsh,Nonconiferous Trees,Scrub | Nonconiferous Tree Fill |
| Natural Environment | Scrub,Nonconiferous Trees,Coppice Or Osiers | Nonconiferous Tree Fill |
| Natural Environment | Mineral Workings (Inactive),Nonconiferous Trees,Scrub | Nonconiferous Tree Fill |
| Natural Environment | Scrub | Scrub Fill |

| | | |
|---|---|---|
| General Surface | Reservoir | Natural Fill |
| Natural Environment | Coppice Or Osiers,Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment | Coniferous Trees (Scattered),Nonconiferous Trees (Scattered),Scrub | Mixed Woodland Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Scrub,Rough Grassland | Nonconiferous Tree Fill |
| Natural Environment | Scrub,Nonconiferous Trees (Scattered) | Nonconiferous Tree Fill |
| General Surface | Tank | Natural Fill |
| Natural Environment | Scrub,Coppice Or Osiers | Coppice Or Osiers Fill |
| Natural Environment | Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment | Lock,Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment | Scrub,Rough Grassland | Scrub Fill |
| Natural Environment | Coniferous Trees (Scattered),Nonconiferous Trees (Scattered),Rough Grassland,Scrub | Mixed Woodland Fill |
| Natural Environment | Scrub,Nonconiferous Trees,Coniferous Trees | Mixed Woodland Fill |
| Natural Environment | Scrub,Coppice Or Osiers,Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment | Orchard | Orchard Fill |
| Natural Environment | Scrub,Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment | Saltmarsh | Marsh Fill |

| | | |
|---|---|---|
| Natural Environment | Nonconiferous Trees (Scattered) | Nonconiferous Tree Fill |
| Natural Environment | Heath | Heath Fill |
| Natural Environment | Coniferous Trees (Scattered),Scrub | Coniferous Tree Fill |
| Natural Environment | Coniferous Trees | Coniferous Tree Fill |
| General Surface | Agricultural Land | Agricultural Land Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Coniferous Trees (Scattered),Scrub | Mixed Woodland Fill |
| Natural Environment | Rough Grassland,Nonconiferous Trees (Scattered) | Nonconiferous Tree Fill |
| Natural Environment | Marsh,Rough Grassland | Rough Grassland Fill |
| Natural Environment | Coniferous Trees (Scattered),Nonconiferous Trees (Scattered),Rough Grassland | Mixed Woodland Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Scrub | Nonconiferous Tree Fill |
| Natural Environment | Scrub,Nonconiferous Trees (Scattered),Coniferous Trees (Scattered) | Mixed Woodland Fill |
| Natural Environment | Scrub,Coniferous Trees,Nonconiferous Trees | Mixed Woodland Fill |
| Natural Environment | Scrub,Coniferous Trees (Scattered) | Coniferous Tree Fill |
| Natural Environment | Coniferous Trees (Scattered),Nonconiferous Trees (Scattered) | Mixed Woodland Fill |

| | | |
|---|---|---|
| Natural Environment | Heath,Rough Grassland | Rough Grassland Fill |
| Natural Environment | Coniferous Trees (Scattered),Scrub,Nonconiferous Trees (Scattered) | Mixed Woodland Fill |
| General Surface | Landfill (Inactive) | Natural Fill |
| Natural Environment | Nonconiferous Trees (Scattered),Coniferous Trees (Scattered),Rough Grassland | Mixed Woodland Fill |
| Natural Environment | Coniferous Trees,Scrub | Coniferous Tree Fill |
| Landform | | Landform Natural Fill |
| Natural Environment, General Surface | Scrub,Nonconiferous Trees | Nonconiferous Tree Fill |
| General Surface | Mineral Workings (Inactive) | Natural Fill |
| Natural Environment | Nonconiferous Trees,Coniferous Trees | Mixed Woodland Fill |
| Natural Environment | Coniferous Trees (Scattered),Rough Grassland | Coniferous Tree Fill |
| Natural Environment | Coniferous Trees,Nonconiferous Trees,Scrub | Mixed Woodland Fill |
| Natural Environment | Scrub,Coniferous Trees (Scattered),Nonconiferous Trees (Scattered) | Mixed Woodland Fill |
| Natural Environment | Marsh | Marsh Fill |
| Natural Environment | Coppice Or Osiers,Nonconiferous Trees,Scrub | Nonconiferous Tree Fill |
| Natural Environment | Nonconiferous Trees,Scrub,Coppice Or Osiers | Nonconiferous Tree Fill |

| | | |
|---|---|---|
| Natural Environment | Coniferous Trees,Scrub,Nonconiferous Trees | Mixed Woodland Fill |
| Natural Environment | Scrub,Coniferous Trees | Coniferous Tree Fill |
| Natural Environment,Rail | Nonconiferous Trees (Scattered) | Nonconiferous Tree Fill |
| Natural Environment,Rail | Rough Grassland | Rough Grassland Fill |
| Natural Environment,Rail | Coniferous Trees (Scattered),Nonconiferous Trees (Scattered) | Mixed Woodland Fill |
| Natural Environment,Rail | Nonconiferous Trees,Scrub | Nonconiferous Tree Fill |
| Natural Environment,Rail | Rough Grassland,Scrub | Scrub Fill |
| Natural Environment,Rail | Coniferous Trees (Scattered),Scrub | Coniferous Tree Fill |
| Natural Environment,Rail | Coniferous Trees (Scattered) | Coniferous Tree Fill |
| Natural Environment,Rail | Coniferous Trees,Nonconiferous Trees | Mixed Woodland Fill |
| Natural Environment,Rail | Nonconiferous Trees (Scattered),Scrub | Nonconiferous Tree Fill |
| Natural Environment,Rail | Scrub,Nonconiferous Trees | Nonconiferous Tree Fill |
| General Surface,Rail | | Natural Fill |
| Natural Environment,Rail | Scrub | Scrub Fill |
| Natural Environment,Rail | Coniferous Trees,Scrub | Coniferous Tree Fill |
| Natural Environment,Rail | Coniferous Trees | Coniferous Tree Fill |

| | | |
|---|---|---|
| Natural Environment,Rail | Coniferous Trees (Scattered),Nonconiferous Trees (Scattered),Scrub | Mixed Woodland Fill |
| Natural Environment,Rail | Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment,Rail | Coniferous Trees,Nonconiferous Trees,Scrub | Mixed Woodland Fill |
| General Surface,Rail,Structure | | Natural Fill |
| Natural Environment,Roadside | Coniferous Trees | Coniferous Tree Fill |
| Natural Environment,Roadside | Nonconiferous Trees (Scattered) | Nonconiferous Tree Fill |
| Natural Environment,Roadside | Rough Grassland,Scrub | Scrub Fill |
| Natural Environment,Roadside | Scrub | Scrub Fill |
| Natural Environment,Roadside | Coniferous Trees (Scattered) | Coniferous Tree Fill |
| Natural Environment,Roadside | Nonconiferous Trees (Scattered),Scrub | Nonconiferous Tree Fill |
| Natural Environment,Roadside | Coniferous Trees,Nonconiferous Trees | Mixed Woodland Fill |
| Natural Environment,Roadside | Nonconiferous Trees,Scrub | Nonconiferous Tree Fill |
| Natural Environment,Roadside | Nonconiferous Trees | Nonconiferous Tree Fill |
| Natural Environment,Roadside | Rough Grassland | Rough Grassland Fill |

| | | |
|---|---|---|
| Natural Environment,Roadside | Coniferous Trees,Nonconiferous Trees,Scrub | Mixed Woodland Fill |
| General Surface,Roadside,Structure | | Natural Fill |
| General Surface,Structure | Aqueduct | Natural Fill |
| General Surface,Structure | | Natural Fill |
| Natural Environment,Structure | Scrub | Scrub Fill |

# Appendix 5 - Map Accuracies

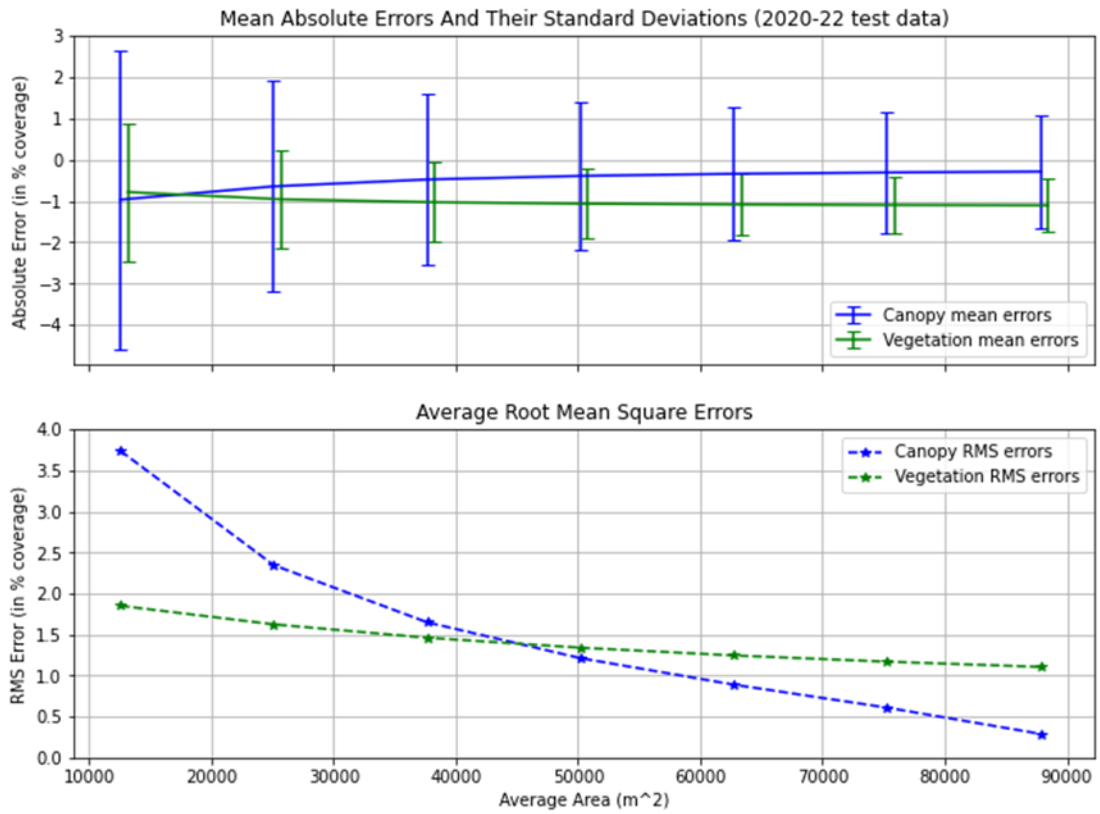
The following table contains the accuracy calculation for each London borough. Note that because each borough contains only a fraction of the labelled points, the accuracy estimate is less precise. The more points available, the better the estimate is. Standard errors are shown for each estimate of accuracy – this standard error is computed with the iTree equation (see *3.4 Additional iTree Estimates*, p.26). If the proportion of 'accurate' points is higher than 90%, then 0.9 is used in the equation for the proportion $p$, to avoid overestimating the precision of the accuracy estimate. The borough with the lowest accuracy for the vegetation map is Barking and Dagenham, with 78 ± 4% and the lowest accuracy for the tree map is 89 ± 2% in Croydon.

| Borough | Number of points | Vegetation Map Accuracy | Tree Map Accuracy | Vegetation Map Standard Error on Accuracy | Tree Map Standard Error on Accuracy |
|---|---|---|---|---|---|
| Barking and Dagenham | 88 | 78% | 94% | 4% | 3% |
| Barnet | 200 | 95% | 94% | 2% | 2% |
| Bexley | 137 | 92% | 93% | 3% | 3% |
| Brent | 110 | 90% | 96% | 3% | 3% |
| Bromley | 393 | 88% | 95% | 2% | 2% |
| Camden | 49 | 93% | 96% | 4% | 4% |
| City of London | 6 | 100% | 100% | 12% | 12% |
| Croydon | 227 | 88% | 89% | 2% | 2% |
| Ealing | 151 | 89% | 94% | 3% | 2% |
| Enfield | 205 | 90% | 97% | 2% | 2% |
| Greenwich | 115 | 89% | 94% | 3% | 3% |
| Hackney | 45 | 99% | 98% | 4% | 4% |
| Hammersmith and Fulham | 34 | 97% | 100% | 5% | 5% |
| Haringey | 64 | 94% | 92% | 4% | 4% |
| Harrow | 115 | 93% | 93% | 3% | 3% |
| Havering | 286 | 88% | 94% | 2% | 2% |
| Hillingdon | 312 | 93% | 95% | 2% | 2% |
| Hounslow | 129 | 94% | 95% | 3% | 3% |
| Islington | 39 | 94% | 92% | 5% | 5% |
| Kensington and Chelsea | 35 | 90% | 91% | 5% | 5% |
| Kingston upon Thames | 105 | 88% | 94% | 3% | 3% |
| Lambeth | 74 | 91% | 95% | 3% | 3% |
| Lewisham | 84 | 91% | 94% | 3% | 3% |

| | | | | | |
|---|---|---|---|---|---|
| Merton | 96 | 90% | 94% | 3% | 3% |
| Newham | 83 | 89% | 97% | 3% | 3% |
| Redbridge | 120 | 92% | 95% | 3% | 3% |
| Richmond upon Thames | 146 | 93% | 92% | 2% | 2% |
| Southwark | 76 | 94% | 97% | 3% | 3% |
| Sutton | 112 | 84% | 91% | 3% | 3% |
| Tower Hamlets | 38 | 79% | 95% | 7% | 5% |
| Waltham Forest | 109 | 87% | 96% | 3% | 3% |
| Wandsworth | 104 | 93% | 97% | 3% | 3% |
| Westminster | 58 | 95% | 98% | 4% | 4% |

# Appendix 6 - Calculation of Error Margins

Figure 4: Errors in vegetation (green) and canopy (blue) covers for the test data. The top panel shows average absolute errors, i.e. the average difference between the real covers and the modelled covers. The lower panel shows average root mean squared errors, i.e. the average size of the error independently of whether it is positive or negative (i.e. an over or under estimation).




**Calculation of error margins for Greater London canopy cover:**

For the smallest area (one patch, ~12,500 m$^2$ or 12.5 ha), the average absolute error in canopy is –0.97 and has a standard deviation of 3.61. As the areas become larger, the absolute error goes down to –0.29, and the standard deviation can be computed[40] as $SE(n) = \sigma/\sqrt{n} = 3.61/\sqrt{7}$ , where 7 is the number of patches looked at. Greater London is 157,000 ha, and therefore equivalent to 125,000 patches. So for Greater London we expect the standard deviation to be 3.61/√125000. Adding this to the absolute error, we

[40] This is the formula for the standard error for a number of samples $n$ and a standard deviation $\sigma$. see https://www.investopedia.com/terms/s/standard-error.asp

get $0.29 + 3.61/\sqrt{125000} = 0.30$. This is the standard error for the canopy estimate for Greater London. We can generalise this for any area as

$$\boldsymbol{SE} = |\boldsymbol{F}| + \frac{\sigma}{\sqrt{area/12.5\ ha}}$$

Where F is the absolute error for large areas (here taken as the absolute error for 7 patches, since this is the largest area investigated).

# Appendix 7 - Manual Labelling

## Set-up

The interface to enable manual labelling of points was set up using ArcGIS Pro desktop, ArcGIS Online, and ArcGIS Enterprise Server.

Data Preparation:

1. Generate 4000 points within the Greater London boundary in GeoDatabase feature layer format, using the *Create Random Points* tool available in ArcGIS Pro.
2. Divide the points into 40 labelling groups by adding a new column ('group_id').
3. Add a new column for 'landuse_label'.
4. Create an attribute domain called 'land use label' to constrain the values allowed in the 'landuse_label' attribute.

| Code | Description |
|---|---|
| 0 | Tree |
| 1 | Vegetation that is not tree |
| 2 | Bare ground |
| 3 | Vegetation but unsure whether tree |
| 4 | Manmade |
| 5 | Unsure |
| 6 | Water |

5. Select 'land use label' in the Domain Name field for the 'landuse_label' attribute.

| Domain Name | Description | Field Type | Domain Type |
|---|---|---|---|
| land use label | land use labelling options used for Green Cover Model 2023 accuracy accessment | Text | Coded Value Domain |

6. Upload the GeoDatabase layer to the GLA's ArcGIS Online platform and publish the data into a hosted feature layer service.
7. In settings, enable editing. To restrict what kind of editing is allowed, unselect 'Add' and 'Delete' options, only select 'Update – attributes only'.
8. Upload height data tiles to the GLA Postgres Database and publish to a cached tile service.

Creation of web map:

1. Create ArcGIS online web map and add the following feature layers:
   - 40 groups of random points (using the definition query 'group_id = 1' etc. to split the single random points layer in to 40 separate layers).
   - Height Tiles layer cached tile service.
   - Blue Sky API (Aerial Photography 12.5cm 2021/2022).
   - Blue Sky API (Colour Infra-red Imagery 50cm 2021/2022).

    - Ordnance Survey light style Web Map Tile service (for base map).
2. Configure the layer names, popup box, and layer visibility within the web map mode.
3. When the web map is ready, create a Web App on top of it (so more mapping functions can be added to the map within the Web App mode)
    - Add legend.
    - Add labelling instructions (to the details panel).
    - Add Edit function so users can label the land use category for each point.
    - Add Authentication (create ArcGIS Online user account and apply authentication to the web map so that only users with login credentials can access).

## Results

Classifications of the 4000 manually labelled points.

| Label Classification | Points (Greater London) |
|---|---|
| Bare Ground | 100 |
| Manmade | 1761 |
| Tree | 777 |
| Unsure | 42 |
| Vegetation (not Tree) | 1175 |
| Vegetation (unsure whether Tree) | 50 |
| Water | 95 |
| Total | 4000 |