Inferring the number of floors of building footprints in the Netherlands

Ellie Roy
student #4290984

1st supervisor: Hugo Ledoux
2nd supervisor: Giorgio Agugiaro
External supervisor: Maarten Pronk (Deltares)

January 8, 2020

1 Introduction

In the Netherlands, detailed information on buildings and addresses is provided as open data via the *Basisregistratie Adressen en Gebouwen*, or BAG for short. This forms part of the Dutch government's system of key registers. Within this dataset, the footprints of all buildings are stored as 2D polygons. Each footprint is associated with a number of attributes, such as the building's construction date and current use. By combining the geometry of these footprints with point cloud data, building height can be determined, as shown in Figure 1. In the Netherlands, the AHN point cloud can be used for this purpose.[1] This is a nationwide elevation model obtained through airborne lidar. The 3D Geoinformation research group at TU Delft has automated the process of extracting building heights from point cloud data and uses this to extend the BAG dataset with additional height attributes. The extended dataset is openly available through their 3D BAG service (Dukai et al., 2018). Height information provides increased value to the BAG dataset, as it allows 3D city models to be generated. These have a wide range of applications, such as the simulation of noise propagation for traffic planning and the estimation of solar irradiation for solar panel placement (Biljecki et al., 2015).

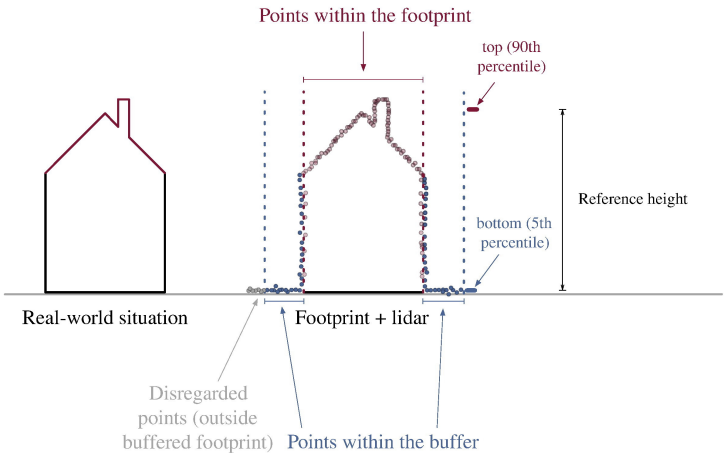


Figure 1: Determination of building height from aerial lidar data and building footprints. Adapted from Biljecki et al. (2017).

Many applications of 3D city models require data on the number of floors. For example, flood response plans require this information to determine the amount of inhabitable storeys remaining during a flood (M. Pronk, personal communication, July 28th, 2020). This is a particularly relevant topic for the Netherlands; a country largely located below sea level with approximately half of the population living at flood risk. For this reason, the Dutch government has developed a website that indicates the expected water height at any location given a major breach of flood defences.[2] This site also shows whether any dry storeys remain in each flooded building. Data on the number of floors is used to provide this information. Another relevant application is energy demand estimation, which is used to assess the benefit of energy retrofitting (Agugiaro, 2015; Biljecki et al., 2015). In addition, the number of floors can be used to estimate building population, which is useful for a variety of network analysis and urban planning applications (Lwin and Murayama, 2009).

Despite the wide range of applications, the number of floors is currently not included as an attribute of the BAG dataset. In some cases, this data is collected at a municipal-level as part of the BAG "plus" (BAG+). This is a more extensive version of the BAG, which indi-

[1] Actueel Hoogtebestand Nederland
[2] www.overstroomik.nl

vidual municipalities may choose to maintain for internal use (Heeres, 2016). The BAG+ is generally not made openly available. Lack of open data on the number of floors means that it must be inferred from other available data. Automatic methods are often based on building geometry, which usually involves dividing the estimated height of a building by an average storey height, as shown in Figure 2. Given the strong correlation between building height and storeys (Biljecki et al., 2017), this approach can often perform well. However, in certain cases this over-simplification also limits the accuracy of the results, which has adverse consequences for the intended application. This is further explained in Section 2.2.

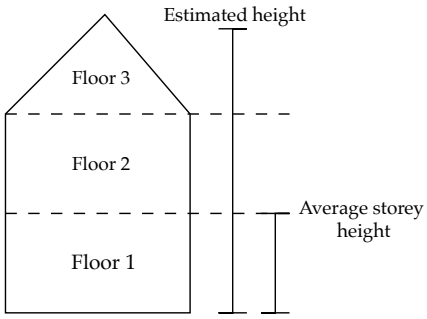


Figure 2: Determination of floor count using building height

Therefore, this thesis aims to develop an alternative method to automatically infer the number of floors. This method will be based on the building footprints available in the BAG dataset. Similar studies related to inferring building properties have used machine learning to obtain accurate results. A number of examples are provided in further detail in Sections 2.3.2 and 2.3.3. These studies show that combining multiple attributes has a greater potential than using a single predictor. For this reason, the alternative method will also focus on using machine learning to combine multiple attributes, such as roof shape and construction year, in order to obtain a more realistic estimate of the number of floors. If the accuracy of the results is sufficient, they may be integrated as a new attribute of the 3D BAG, allowing the data to become openly available for use in the previously mentioned applications. In addition to these applications, the results may be useful in two other main areas. Firstly, as input to algorithms related to reconstructing the interiors of 3D city models (Boeters et al., 2015). Secondly, for the automatic identification of mistakes in the floor count included in the BAG+, which is collected manually and is thus prone to errors.

The following sections explain the context, motivation and approach of this research in further detail. Section 2 provides an overview of the relevant background and related work, focusing on (1) 3D city models, (2) geometric approaches to estimate the number of floors and (3) machine learning. The main objectives of the research and the proposed methodology are presented in Sections 3 and 4. This is followed by a more in-depth explanation of the tools and datasets that will be used in Section 5. Preliminary results are presented in Section 6 and finally, the project planning is outlined in Section 7.

2 Background and related work

2.1 3D building reconstruction

One of the most important concepts in the 3D reconstruction of buildings is the level of detail (LOD). This concept defines the geometric and semantic complexity of a 3D city model, in order to describe its degree of resemblance to the real-world situation (Biljecki et al., 2016b). The most widely used standard for categorising the level of detail is the OGC CityGML 2.0

specification, which defines 5 levels of detail (OGC, 2012; Gröger and Plümer, 2012). This categorisation has been further refined by Biljecki et al. (2016a) to consist of four sub-categories per level of detail, with the exclusion of LOD4 which is rarely used in practice. The refined LOD specification is shown in Figure 4. It reduces ambiguity in the CityGML categorisation by providing a more precise definition of the geometric detail required within each sub-category (Biljecki et al., 2016a). For the purpose of this thesis, LOD1.2, 1.3 and 2.2 are the most important. These models are readily available as part of the 3D BAG 2.0 and could be used to extract geometric features relevant to estimating the number of floors.

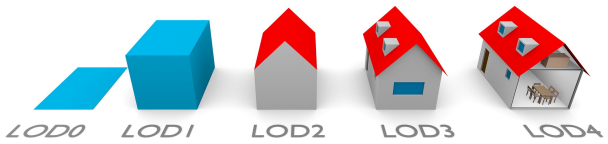


Figure 3: The five LODs defined by the OGC CityGML 2.0 standard. Reprinted from Biljecki et al. (2016a).

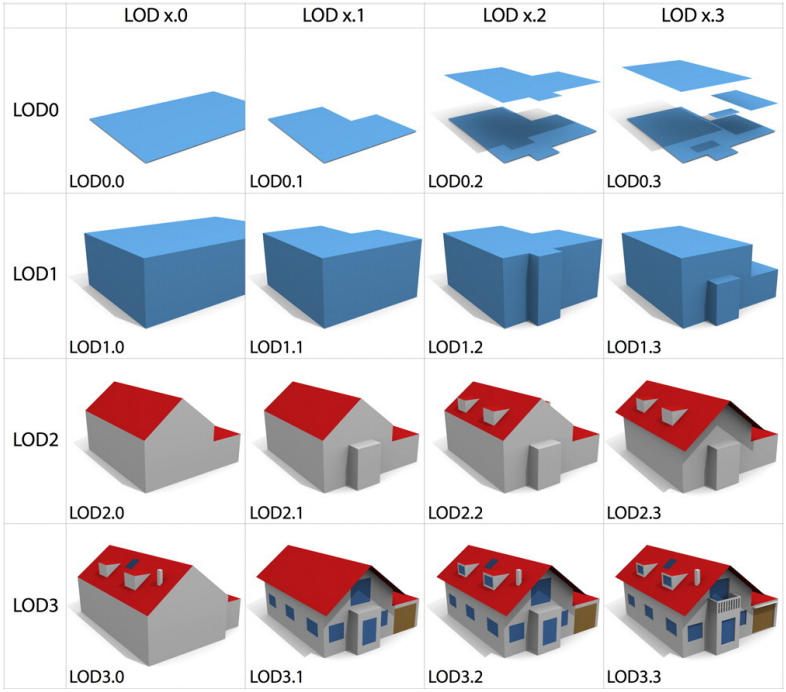
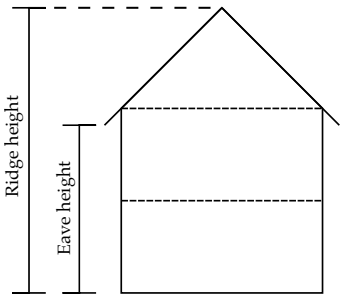


Figure 4: Improved LOD specification. Reprinted from Biljecki et al. (2016a).

Despite the apparent simplicity of the LOD1 block model, there is a high level of ambiguity in its geometric representation (Biljecki et al., 2014). As illustrated in Figure 5a, the position of the top surface varies significantly depending on the geometric reference chosen to represent the building's height. The 3D BAG takes geometric references into account by including six different roof heights as attributes, based on the calculation of different percentiles from the point cloud of each building (Dukai et al., 2019). In the context of this thesis, the difference between geometric references representing the ridge and eaves of the roof could enable any storeys beneath slanted roofs to be identified, as shown in Figure 5b.

(a) Different height references for the top surface of an LOD1 model. Reprinted from Biljecki et al. (2014).



(b) Ridge vs eave height

Figure 5: Geometric references

2.2 Geometric approaches to estimate number of floors

Geometric approaches are common in the estimation of the number of floors. These can be divided into two main types, based on either building height or internal area.

2.2.1 Height-based

This approach requires the height of a building to be known, allowing the number of floors to be estimated by dividing this by an average storey height and rounding to the nearest integer (as shown in Figure 6). This is currently the main approach used to determine the number of floors for flood response plans in the Netherlands (M. Pronk, personal communication, July 28th, 2020). It is also cited in a number of research papers (Shiravi et al., 2015; Alahmadi et al., 2013). For Dutch flood response plans, building height is calculated as the difference between the 75th roof percentile and 50th ground percentile, as recorded in the 3D BAG dataset. An average storey height of 2.65 metres has been derived from the Dutch building code (*Bouwbesluit*), based on an average from standards before and after 2003. The reliability of this method was assessed by manually inspecting a sample of buildings in Google Street View (M. Pronk, personal communication, December 2nd, 2020). However, the overall accuracy is currently unknown. This thesis will therefore also focus on evaluating the performance of this method.

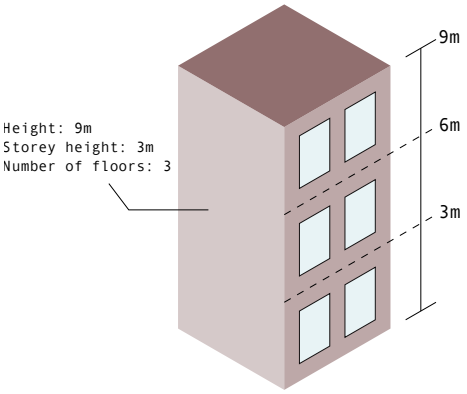


Figure 6: Schematic illustration of the height-based method

For buildings with flat roofs a height-based approach can work well, as they are essentially equivalent to their LOD1 representation. However, for slanted roofs the results are highly dependent on the geometric reference chosen to represent building height. In the Netherlands only 34% of buildings are estimated to have a truly flat roof (Dukai et al., 2019), so using an appropriate height reference is important. In addition, is also important to select a representative average storey height. This is difficult because buildings have different ceiling heights and floor thicknesses, as well as variable ceiling heights from floor to floor. For example, a ground floor lobby may be much taller than the floors above. This means that, despite a strong correlation between the number of floors and building height, there is also substantial variation in the number of floors of buildings with the same height. However, residential buildings are generally more consistent than non-residential buildings (Biljecki et al., 2017).

2.2.2 Area-based

Alternatively, the number of floors can be estimated by dividing the internal area of a building by its footprint area and rounding to the nearest integer. In the Netherlands, the net internal area (NIA) is documented within the BAG and can be used for this purpose. The NIA does not represent the gross area of a building, but rather the usable area of individual building units. This means that it excludes areas related to, for instance, stairs and elevator shafts or places where ceiling height is lower than 1.5 metres (Biljecki et al., 2017). This approach is used for Dutch flood response plans when building height information is not available (M. Pronk, personal communication, July 28th, 2020). For example, for buildings constructed more recently than the latest AHN point cloud. The height-based method is preferred because NIA is often incorrectly registered in the BAG. In addition, the footprint area includes walls while the NIA does not, which can lead to errors (M. Pronk, personal communication, July 28th, 2020). There are also discrepancies in the calculation of NIA before and after the introduction of the Dutch standard for areas and volumes of buildings (NEN 2580:2007) (Boeters, 2013). Furthermore, in some cases the building footprint registered in the BAG represents multiple buildings or is much larger than the floors above. However, a potential benefit is that underground floor area is included in the NIA, meaning that underground storeys can be inferred (Biljecki et al., 2017).

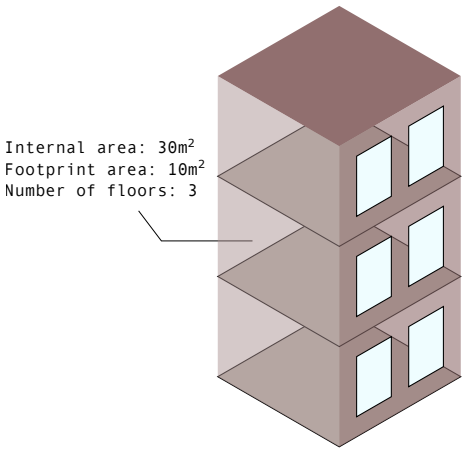


Figure 7: Schematic illustration of the area-based method

2.3 Machine learning

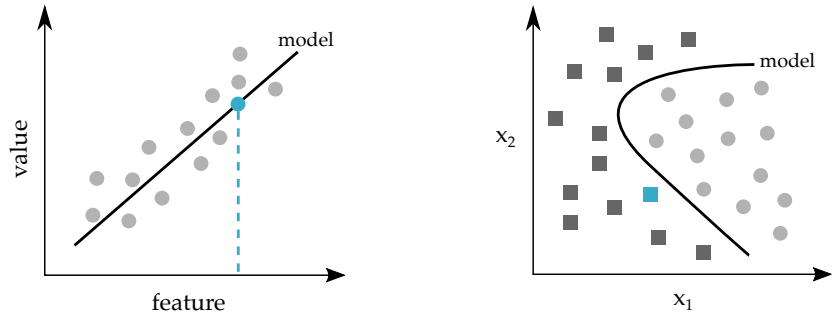
The following section provides a background on machine learning and outlines two relevant examples of its application. These examples are chosen because they are based on using building footprints and attributes to infer a building property. This provides a useful reference for predicting the number of floors from similar data, as further elaborated on in Section 4.2.

2.3.1 Background

Machine learning is the field of study that enables computers to learn from data without being explicitly programmed (Géron, 2019). Since they do not rely on hard-coded rules, machine learning systems can be used to gain insights into large, complex datasets and to solve problems for which conventional approaches do not perform well. It is a technique used for a wide variety of tasks ranging from email spam filters to facial recognition.

Machine learning techniques can be divided into different categories. One major distinction is made between supervised and unsupervised methods. Supervised algorithms require the input data to consist of a mapping between features and labels, whereas unsupervised algorithms determine this mapping independently and require only features. In the case of inferring the number of floors, the labels are the building floor count (ground truth), while the features are any relevant building properties, such as roof shape and construction year. Supervised algorithms can be further sub-divided into regression and classification tasks. Figure 8 illustrates the difference between these tasks. Regression algorithms are used to predict continuous values (e.g. building height), whereas classification is used to predict discrete classes (e.g. roof shape categories) (Müller and Guido, 2016). For the estimation of the number of

floors, it is not clear which of these tasks is more appropriate. This is discussed in further detail in Section 6.2.



(a) Regression (b) Classification

Figure 8: Two main types of supervised machine learning algorithms, with the predictions made for new instances shown in blue

Random Forest (RF) is one example of a supervised machine learning algorithm that can be used for both classification and regression tasks. This is the main algorithm used in the examples provided in the following sections. It works by combining multiple decision trees, which are each trained on different random subsets of the training data (Géron, 2019). A schematic representation is shown in Figure 9. Decision trees are essentially a hierarchy of if/else questions, which the algorithm constructs by finding the best feature to split each node. One main drawback of decision trees is their tendency to over-fit the data. Through the combination of many slightly different trees, over-fitting can be averaged out, allowing more reliable results to be obtained (Müller and Guido, 2016). The main advantage of RF is that the feature importance is calculated, which enables predictive models to be constructed using the optimal subset of features. Other examples of supervised machine learning algorithms include linear regression, Support Vector Machines (SVMs) and neural networks. The comparison of different models plays an important role in the development of a machine learning algorithm. Multiple models can also be combined through ensemble learning to improve the accuracy of the results (Géron, 2019).

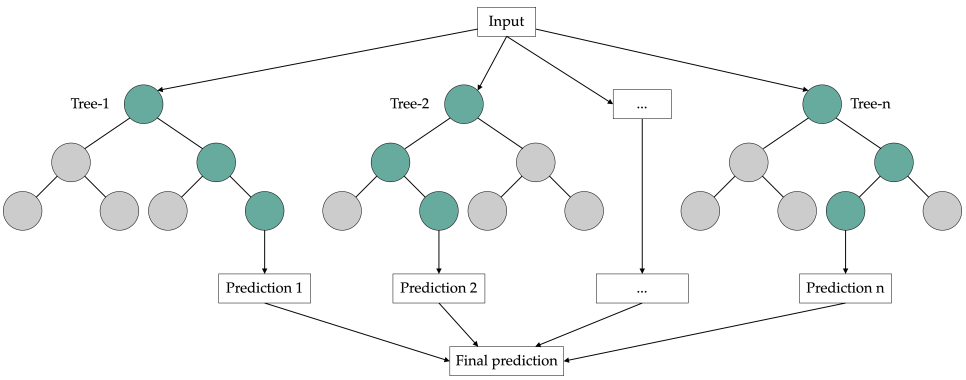


Figure 9: Random forest algorithm

2.3.2 Building height

Biljecki et al. (2017) used machine learning to infer building heights using only 2D features derived from building footprints and attributes. This research highlighted the potential to estimate building height without elevation data. Furthermore, the results provided a higher

level of accuracy than the commonly used geometric approach, which is based on multiplying the number of floors by an assumed storey height. Ten different features were derived and different combinations were used as input to a Random Forest regression algorithm. These combinations were selected in order to take into account scenarios where not all features are available. The number of storeys, building age and net internal area were found to be the features with the highest importance.

Lánský (2020) further extended on this research by inferring the height of all buildings in the USA, using similar features derived from 2D building footprints and attributes. The level of accuracy achieved for rural and suburban areas was relatively good, but the mean absolute error (MAE) for central business districts in cities was high. This is potentially because the training data was not representative enough. Similar research has also been conducted by Anh et al. (2018) using the same geometric predictors as Biljecki et al. (2017) to infer the height of buildings in Hanoi, Vietnam. However, the performance of the model is lower, which the authors attribute to the limited amount of training data used by the algorithm.

2.3.3 Roof shape

Biljecki and Dehbi (2019) explored the use of machine learning to infer roof shape from LOD0 and LOD1 models without the acquisition of lidar data. Twelve features were used as input to a Random Forest classifier, around half of which were derived from the geometry of the building footprint. The model achieved an accuracy of 85% for the prediction of six roof classes and 92% for distinguishing flat roofs from slanted roofs. Footprint area and building height were found to be the two most important features in this case. This research is closely related to the previously described work of Biljecki et al. (2017), as it forms part of a possible pipeline for constructing LOD2 models from footprints without elevation data (Figure 10).

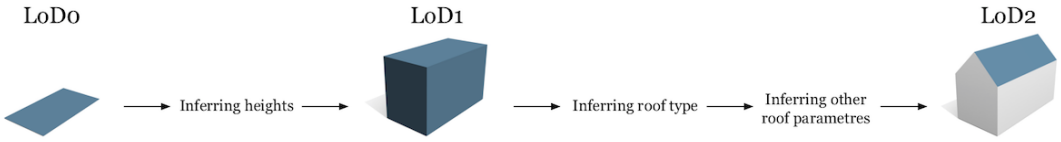


Figure 10: Pipeline for generating LOD2 models from building footprints without elevation data. Adapted from Biljecki and Dehbi (2019).

2.4 Conclusions from the literature review

The main conclusions from the literature review are:

1. Commonly used geometric approaches rely on a single feature (e.g. building height) to determine the number of floors, which can be prone to errors. The overall accuracy of these approaches is currently unknown and requires evaluation.

2. Machine learning may allow more accurate results to be obtained, since multiple attributes have a greater potential than the use of a single predictor. Features used to infer building height and roof shape in previous studies could be useful in the prediction of the number of floors from similar data.

3. The use of different combinations of features is an important consideration when developing a machine learning algorithm. This also allows situations where not all features are available to be taken into account.

4. The comparison of geometric height references could be an important aspect to investigate in relation to identifying storeys beneath slanted roofs (see Figure 5b).

3 Research objectives

The main research question of this thesis is:

To what extent can machine learning provide a better estimate of building storeys than a purely geometric approach?

The goal of this research is to develop a machine learning algorithm which improves on the results obtained from the geometric approaches described in Section 2.2. Rather than relying on a single predictor (e.g. height), this algorithm will be based on multiple building attributes and geometrical features. The focus will be on developing an algorithm which uses the optimal subset of building features to determine the number of floors, taking into account cases where not all attributes are readily available (e.g. buildings constructed after the most recent lidar survey). If a sufficient level of accuracy is achieved, the results may be integrated as a new attribute of the 3D BAG service.

3.1 Sub-questions

In order to achieve the main research objective, the following sub-questions are defined:

a. Which features are related the number of floors? Is there any overlap between these features and which subset is optimal?

b. Which supervised learning method provides the best results? Does combining different machine learning algorithms through ensemble learning improve the results?

c. What level of accuracy can be achieved? To what extent does this improve on the current estimation?

3.2 Scope

The following scope is defined for the research:

- The focus will be on buildings in the Netherlands due to:
  - the wide availability of open data
  - the aim to integrate the number of floors as a new attribute of the 3D BAG service
  - the fact that applications related to flooding are specifically relevant to the Netherlands, since a large portion of the country is under sea-level.
- The focus will be on (mixed-)residential buildings because:
  - this avoids over-complication of the problem, since non-residential buildings are more variable in design
  - some applications are most relevant to this type of building use (e.g the liveability of homes subject to flooding)
- The comparison of different machine learning algorithms will not be extended to neural networks/deep learning, in order to prevent the algorithm from becoming a ”black box”.
- The focus will be on extracting features based on building geometry and existing attributes, not on the modelling of buildings in LOD2. An LOD2 model of the Netherlands is already openly available as part of the 3D BAG 2.0.

4 Methodology

An overview of the proposed methodology is provided in Figure 11. It consists of three main stages: (1) data acquisition and pre-processing, (2) data preparation and (3) modelling.

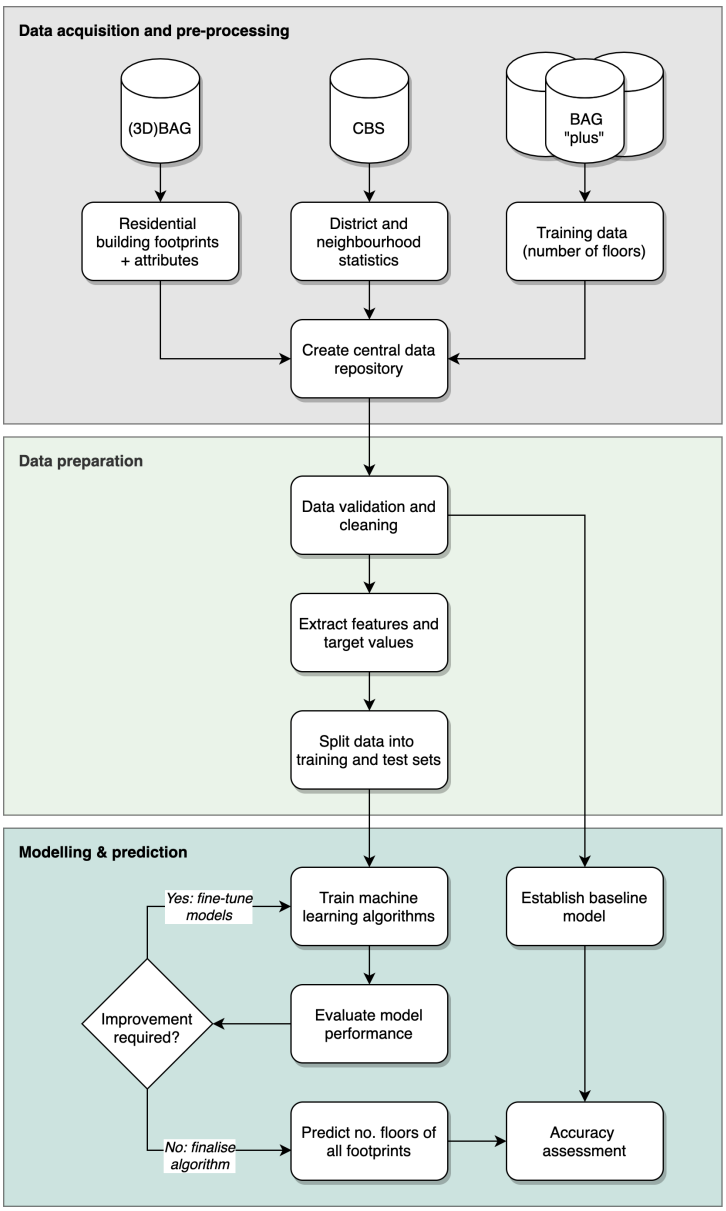


Figure 11: Flowchart of methodology

4.1 Data acquisition and pre-processing

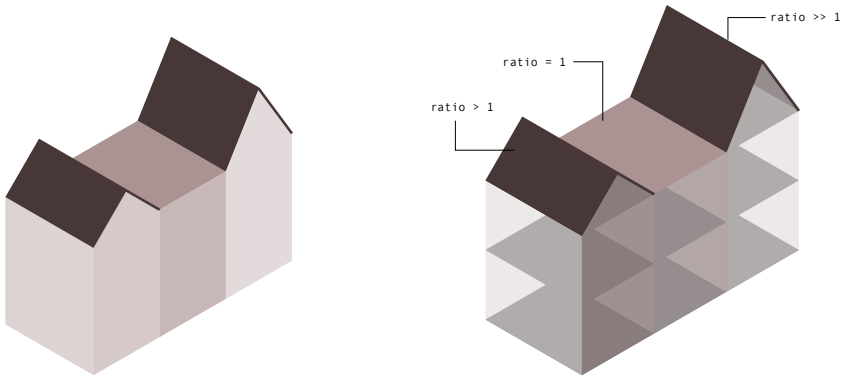
The first stage of the methodology involves collecting the data required for the analysis. This will be obtained from four main datasets, as outlined in further detail in Section 5.2. In addition to the BAG building footprints, this includes a dataset containing neighbourhood and district statistics. As explained in further detail in the following section, this data will be used to extract additional features which may be related to the number of floors.

This stage also involves a number of pre-processing steps. Firstly, the 2D building footprints from the BAG dataset will be filtered by current use to consist of only (mixed-)residential

buildings. Since the focus is on estimating the number of floors of residential buildings, discarding any other building types will help to reduce data storage. After filtering the data, the net internal area of each building will be calculated based on the usage area of its constituent building units. This data is available as an attribute of the *verblijfsobjecten* contained in the BAG dataset. Furthermore, since the statistical data is only available at a neighbourhood and district level, it will need to be associated to each building contained within the boundaries. Finally, the training data from different municipalities will be combined into a uniform format. After the pre-processing steps, the data will be loaded into a PostgreSQL/PostGIS database.

4.2 Data preparation

The second stage of the methodology involves the preparation of the data for the machine learning algorithm. The first step is to validate and clean the data. This is expected to be the most time-consuming step, as it is essential to ensure that the input data is clean for the algorithm to perform well (Géron, 2019). This mainly involves verifying the accuracy of the training data and removing any instances that are incorrectly labelled. It also involves checking the consistency of the 3D BAG footprints. For example, some buildings are marked as having valid height while they do not have any ground points, which leads to incorrect height calculation. Furthermore, any missing data or anomalies will be identified and handled. The handling of missing data could either involve ignoring these instances completely, setting the values to zero or using the mean/median as a replacement value.



(a) Exterior (b) Interior

Figure 12: Schematic illustration showing the potential use of the roof to footprint area ratio for identifying floors below slanted roofs. A ratio equal to 1 indicates a flat roof, as shown by the middle building. A ratio larger than 1 indicates a slanted roof. The larger this value, the higher the likelihood of a full storey located below the roof, as shown by the buildings on the left and right.

Once the data has been validated and cleaned, the next step will be to extract features. These features will be used as input to the machine learning algorithm. A list of attributes that may be useful for feature extraction is provided in Table 1. This list categorises the attributes into cadastral, geometrical and statistical. After feature extraction, feature scaling may be required to normalise the feature ranges, as machine learning does not perform well if the input features have very different scales (Géron, 2019). After this, the data will be split into training and test sets. The training set will be used to train the machine learning algorithm, whereas the test set will be used to evaluate its performance. Usually 20% of the dataset is set aside to create the test set. A simple way to split the data is through random sampling. Alternatively, building characteristics could be used to stratify the split, helping to make the training set more representative.

Table 1: List of useful attributes for feature extraction

Category	Attribute	Explanation (and reference)
Cadastral	Construction year	Building age is related to storey height to some extent (Biljecki et al., 2017). Possibly useful for determining storey height from the Dutch building code.
	Net internal area	Calculated as summed area of building units, used to compute ratio with footprint area.
Geometrical	Footprint area	For comparison with net internal area.
	Measured height	Roof percentile – ground percentile. Different geometric references could be used for comparison of roof eaves and ridge, in order to determine the ceiling height under slanted roofs, as shown in Figure 5b.
	Number of neighbouring residential buildings	Building height has been linked to the number of neighbouring buildings, shorter buildings (less floors) typically have more neighbours (Biljecki et al., 2017)
	Roof shape	Either based on binary classification (flat/not-flat) or a more complex shape metric if results require improvement. The roof flat attribute in the 3D BAG implements quite a strict definition of flat (Dukai et al., 2019).
	Roof area	Based on LOD2 model. As shown in Figure 12, this could be compared with footprint area to identify full storeys below a slanted roof.
Statistical	Population density	Buildings with more floors accommodate more people. Volumetric approaches for calculating building population are based on the number of floors (Lwin and Murayama, 2009), suggesting a strong relationship.
	% of multi-household buildings	Multi-household buildings are generally taller (contain more floors).
	Distance to supermarkets, etc	Taller buildings (with more floors) accommodate more people, potentially leading to higher demand for supermarkets and other amenities (Biljecki et al., 2017).

4.3 Modelling and prediction

The third stage of the methodology consists of implementing the actual machine learning algorithm itself. As part of this stage, a baseline model will also be established. This is a simple

model that can be used to evaluate the performance of the machine learning algorithm. In this case, a reference model will be created using the geometric approaches described in Section 2.2. This is the same as the method currently used to determine the number of inhabitable storeys used in flood response plans. After creating the baseline mode, different machine learning algorithms will be selected and trained. Once a number of promising models have been developed, each one will be iteratively evaluated and fine-tuned, taking into account feature importance, cross-validation scores and hyperparameter values. Once the best combination is found, the implementation will be finalised and used to predict the number of floors of all buildings. Multiple models may potentially be combined through ensemble learning. Finally, the baseline model will be used to evaluate the model performance.

5 Tools and datasets

5.1 Tools

In order to read, store and process the data used in this project several open-source tools are required. These are summarised below:

- **Python** will be used to develop the main machine learning algorithm using the `scikit-learn` library. It will also be used to read and (pre-)process the data using the `(geo)pandas` and `numpy` libraries. Furthermore, plots will be created using `matplotlib`.
- **PostgreSQL** will be used to manage the data from different sources in a central database. The (3D)BAG datasets will be restored from Postgres backup files and the training data will be loaded using the `ogr2ogr` command line utility provided by GDAL. **PostGIS** will be used to extend the database with spatial analysis tools in order to extract geometric features from the data.
- **QGIS** will be used for the visualisation of 2D data and **Azul** for the visualisation of 3D city models in CityJSON/CityGML format.

5.2 Datasets

The following datasets will be used:

1. *Basisregistratie Adressen en Gebouwen* (BAG). As explained in Section 1, this dataset consists of polygons representing the footprints of all buildings in the Netherlands. It also consists of points representing each address. It is maintained at the municipality level and distributed as a national dataset by *Kadaster*, the Dutch Cadastre. Each building and address is associated with a number of attributes, such as the current use and construction year. The building footprints are based on the projection of the roof outline, rather than the true footprint on the ground.
2. *3D BAG* (Dukai et al., 2018). This is a dataset developed and maintained by the 3D Geoinformation research group at TU Delft. It provides additional height information about the roof and ground surfaces of each building in the BAG dataset. This information is derived at multiple reference heights from the AHN point cloud of the Netherlands. Parameters related to the quality of LOD1 models generated from the data are also included as attributes (Dukai et al., 2019). The dataset is updated monthly but is currently in the process of being updated to version 2.0, which will include LOD1.2, 1.3 and 2.2 models of all buildings in CityJSON format (a beta version is already available).

3. *CBS wijken en buurten*. This is a dataset maintained by the Dutch central bureau for statistics. It consists of census data collected at a district and neighbourhood level on a yearly basis, such as the population density and percentage of single household dwellings.

4. *Training data*. Data on the number of floors is required to train the machine learning algorithm. This data can be obtained from the BAG "plus", which is an extended version of the BAG maintained by individual municipalities. Since the maintenance of this dataset is optional, the content varies per municipality and the definition of attributes is not standardised. This means that additional work will be required to transform the data into a uniform format. Furthermore, any data on the number of floors is generally not made publicly available, meaning individual municipalities must be contacted to request its use. Training data has already been obtained from three municipalities in Randstad (Rotterdam, The Hague and Amsterdam), as well as a rural municipality in the East of the Netherlands (Rijssen-Holten). More data may be required from municipalities outside of the Randstad in order to avoid potentially introducing bias.

6 Preliminary results

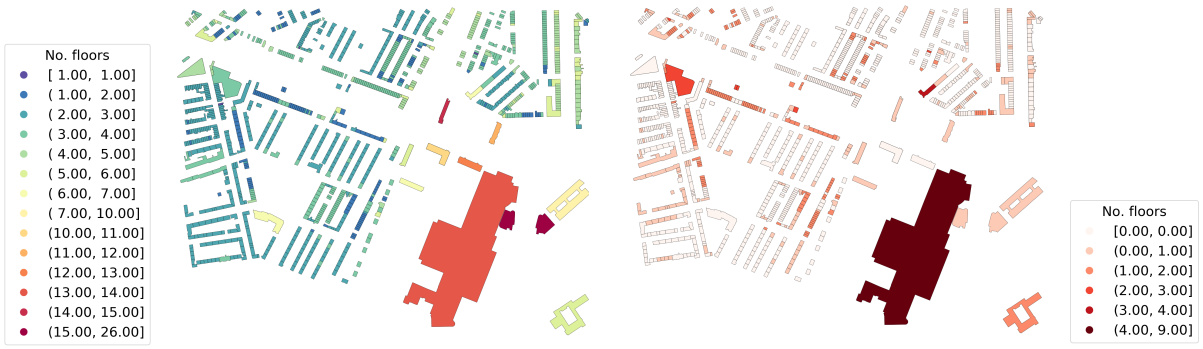
Preliminary results have been obtained for three case study areas in Rotterdam and The Hague. The table below provides an overview of the study areas.

Case study	**Area (m²)**	**No. buildings**
Zuidplein (Rotterdam)	9.62×10^5	1431
Hoogvliet (Rotterdam)	6.04×10^5	1749
Mariahoeve (The Hague)	2.18×10^6	671

6.1 Reference model results

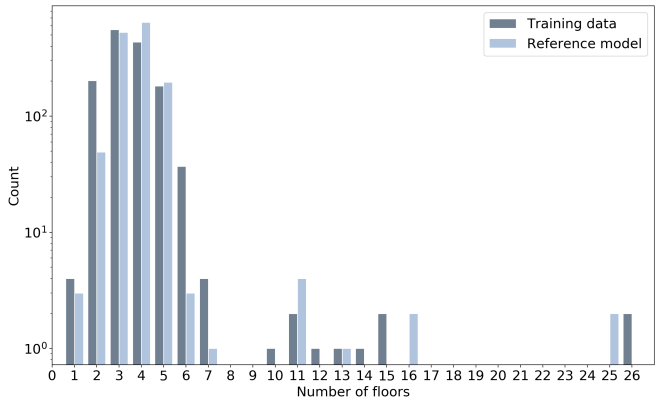
Firstly, results were obtained using a simple geometric approach. The method implemented corresponds to the calculation of the number of floors currently used for flood response plans in the Netherlands. These results will later be used as a reference model to evaluate the performance of the final machine learning algorithm. In order to gain an understanding of the accuracy of these results, they were compared to the training data available for each case study. An example is shown in Figure 13 for Zuidplein in Rotterdam. The highest absolute difference between the training data and reference model is obtained for the large building at the bottom right of the area. This is actually a mixed-residential shopping mall with a much larger footprint than the rest of the building, which causes the geometric approach to greatly underestimate the number of floors. This also occurs in a number of cases for the other study areas. Aside from this one exception, the reference model obtains reasonably similar results for the majority of buildings in this study area. In particular, buildings below six storeys correspond very well. However, further inspection using Google Street View to validate the results has led to some interesting discoveries about the quality of the training data. In quite a few cases, the training data for Rotterdam is incorrectly labelled and the reference model actually provides better results. A number of examples are shown in Figure 14.

Due to concerns about the quality of the training data for Rotterdam, a second study area was investigated. Hoogvliet was chosen because Boeters et al. (2015) referred to this neighbourhood as being a good choice for a case study due to the variation in different building



(a) Training data

(b) Absolute difference with reference model



(c) Distribution of number of floors

Figure 13: Zuidplein, Rotterdam

types. Since Hoogvliet is a very large neighbourhood, a smaller area was selected consisting of mainly 3-storey terraced houses with slanted roofs. This provided a good test for the reference model. Figure 16a provides a comparison between the training data and reference model results. In most cases, the reference model underestimates the number of storeys of 3-storey buildings by one floor. This seems to be due to sloped roofs and relatively low floor-to-ceiling heights, as shown in Figure 15. This highlights the need for a more advanced model to estimate the number of floors. In the case of Hoogvliet, it seems that a predictor related to building age may be a useful feature, as this could indicate a particular architectural style.

Finally, part of the Mariahoeve neighbourhood in The Hague was analysed. This area consists of multiple blocks of high-rise flats, as well as terraced housing. Figure 16b provides an overview of the distribution in the number of floors, for both the training data and reference model. Overall, the results of the reference model correspond quite well. However, there are some discrepancies due to the standard used to define the number of floors by the municipality of The Hague. The training data includes a distinction between whether the first building storey starts at ground level (*etage*) or above/below ground (*woonlaag*). This could be a useful attribute, but also leads to some ambiguity. In some cases a *woonlaag* starts half a storey above/below ground and in other cases this is a full storey, meaning that the ground floor may be completely excluded from the floor count, as shown in Figure 17. It unclear whether this is consistently the case for apartment blocks, which could be a major limitation and requires further investigation. Another limitation of this training dataset is that it only includes single-household buildings connected to the ground and flats accessed via a gallery, meaning that the data is a lot sparser.

(a) Flakkeesestraat 85-89 (training data: 2 floors; reference model: 4 floors; actual: 4 floors)

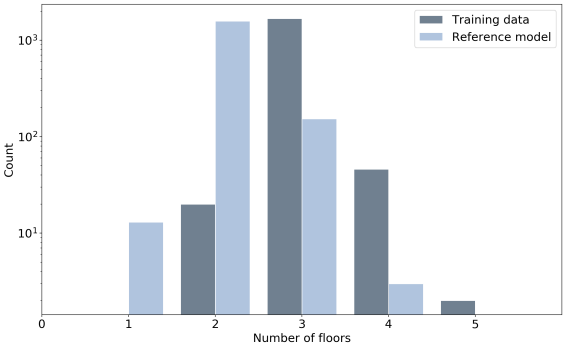
(b) Pleinweg 108–110 (training data: 2 floors; reference model: 5 floors; actual: 5 floors)

Figure 14: Examples of incorrectly labelled buildings in Zuidplein, Rotterdam
Imagery © Google

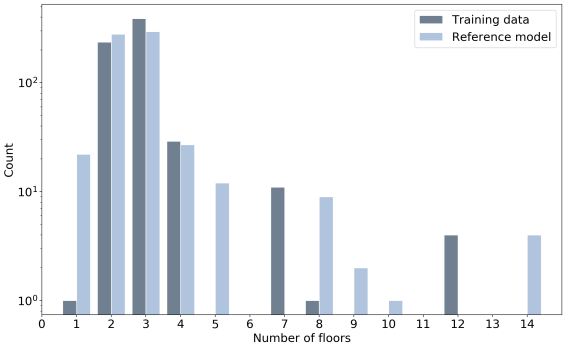
(a) Janswaal 28–32 (training data: 3 floors; reference model: 2 floors; actual: 3 floors)

(b) Wederik 1–17 (training data: 3 floors; reference model: 2 floors; actual: 3 floors)

Figure 15: Examples of incorrect reference model results in Hoogvliet, Rotterdam
Imagery © Google



(a) Hoogvliet, Rotterdam



(b) Mariahoeve, The Hague

Figure 16: Distribution of the number of floors

(a) Isabellaland (Training data: 7 floors; reference model: 9 floors; actual: 8 floors)

(b) Robertaland 42 (Training data: 2 floors; reference model: 3 floors; actual: 3 floors)

Figure 17: Examples from The Hague where floors are counted from above ground level.
Imagery © Google

6.2 Machine learning results

After establishing the reference model, Random Forest regression and classification algorithms were implemented using two input features: (1) building height and (2) floor area ratio. The main purpose of this analysis was to gain an initial understanding of the `scikit-learn` library. Building height was calculated as the difference between the 75th roof percentile and 50th ground percentile, as recorded in the 3D BAG. Floor area ratio was calculated as the net internal area of the building units divided by the footprint area.

As mentioned in Section 2.3.1, it is not clear whether a regression or classification algorithm would be better suited to inferring the number of floors. Therefore, both were implemented during the preliminary analysis. Since the number of floors is a discrete attribute, a classification algorithm appeared more appropriate at first. The results of a regression algorithm would need to be rounded to the closest integer. However, for classification to work properly, the training data must include sufficient examples of each number of floors encountered within buildings in reality. If this is not the case, the algorithm would be unable to predict the number of floors correctly for the missing classes. It seems unlikely that each floor count will be sufficiently represented in the training data, meaning that a classification algorithm may only be useful for distinguishing broader ranges (e.g. whether a building has between 1–3 floors). This suggests that regression may be more suitable. The comparison between regression and and classification will be further investigated during the thesis in order to understand the limitations of each and to determine which performs best.

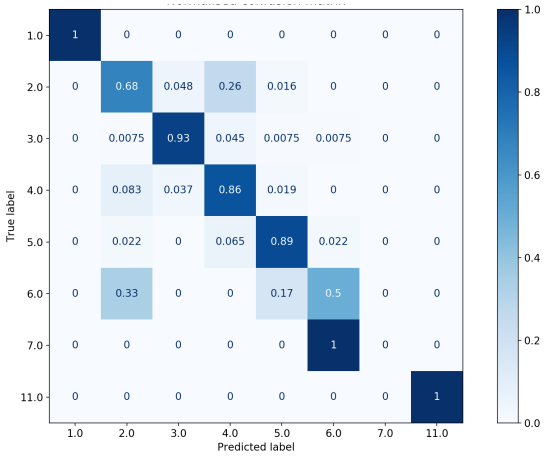
The importance of the two input features used by the algorithms is shown in Table 2. The importance is roughly the same for the classification algorithm in all cases, although area has slightly more importance for both Hoogvliet and Mariahoeve. This may be because there are more buildings in these areas with slanted roofs, making height a less good predictor. For the regression algorithm, height was a much more important predictor for both Zuidplein and Mariahoeve, while for Hoogvliet the feature importance was roughly equal.

The accuracy of the models was evaluated using a test set. An in-depth evaluation was not performed since the number of floors was only predicted for a small amount of buildings and because the training data from Rotterdam contains substantial gross errors. However, it was still useful to gain an initial understanding of the accuracy metrics that can be used. For classification algorithms confusion matrices can be used to compare the true labels to the

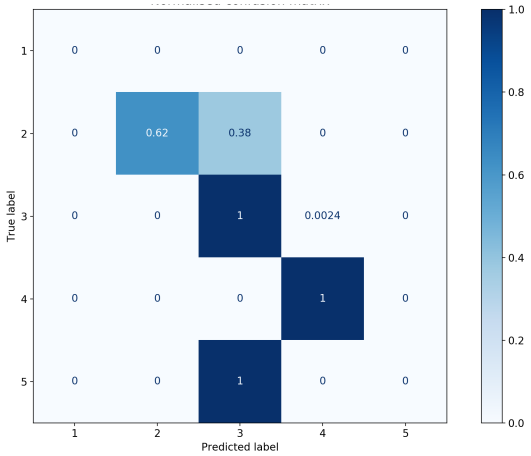
Table 2: Overview of feature importance

		Classification		Regression	
		Height	Area ratio	Height	Area ratio
Importance	Zuidplein	0.54	0.46	0.84	0.16
	Hoogvliet	0.45	0.55	0.55	0.45
	Mariahoeve	0.45	0.55	0.77	0.23

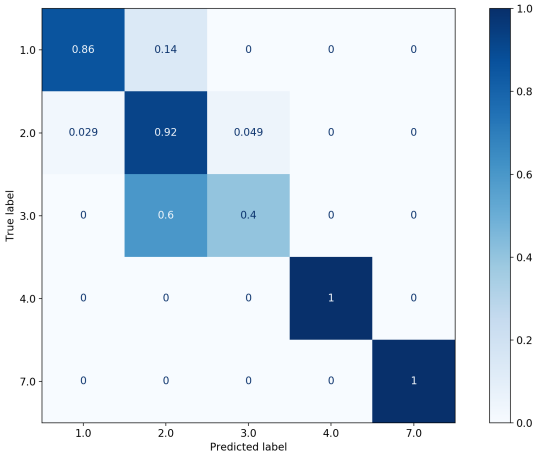
predicted labels. These are shown in Figure 18 for the three study areas. A value of 1 on the diagonal indicates that the class has been predicted correctly. It is interesting to note that the algorithm struggled most to correctly infer the number of floors for buildings with 2–3 storeys, similar to the reference model. The performance of the regression algorithm was difficult to assess. The mean absolute error (i.e. the average difference between true and predicted values) can be used, but this was zero on average in this case, using rounded predicted values.



(a) Zuidplein



(b) Hoogvliet



(c) Mariahoeve

Figure 18: Normalised confusion matrices showing the performance of the RF classification

7 Project planning

7.1 Tasks and deadlines

The project plan is outlined in the Gantt chart shown in Figure 19. This chart presents the order and duration of each of the main tasks. Deadlines are indicated with an orange diamond.

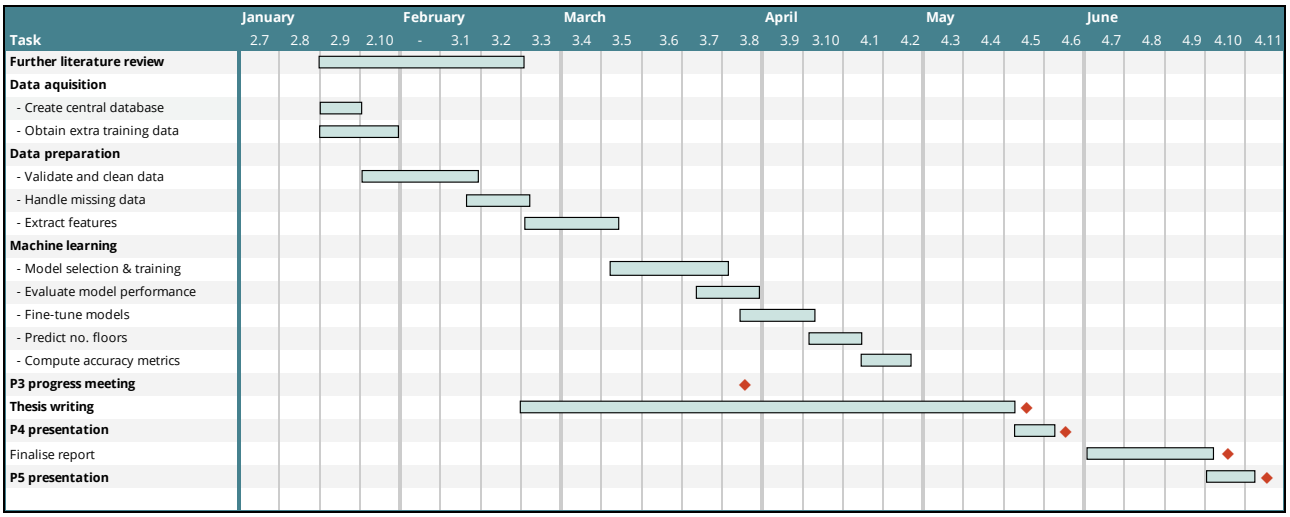


Figure 19: Gantt chart showing the main tasks, deadlines and overall project schedule

7.2 Meetings

A weekly 30 minute meeting will be held with the first supervisor. Meetings will be held with the second supervisor and external supervisor when additional guidance or feedback is required. The co-reader is still to be decided.

8 References

G. Agugiaro. Energy planning tools and CityGML-based 3D virtual city models. Experiences from Trento (Italy). *Applied Geomatics*, 8, 09 2015. doi:10.1007/s12518-015-0163-2.

M. Alahmadi, P. Atkinson, and D. Martin. Estimating the spatial distribution of the population of Riyadh, Saudi Arabia using remotely sensed built land cover and height data. *Computers, Environment and Urban Systems*, 41:167 – 176, 2013. ISSN 0198-9715. doi:https://doi.org/10.1016/j.compenvurbsys.2013.06.002. URL http://www.sciencedirect.com/science/article/pii/S0198971513000598.

P. Anh, N. T. N. Thanh, C. T. Vu, N. V. Ha, and B. Q. Hung. Preliminary Result of 3D City Modelling For Hanoi, Vietnam. In *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 294–299, 2018. doi:10.1109/NICS.2018.8606867.

F. Biljecki and Y. Dehbi. RAISE THE ROOF: TOWARDS GENERATING LOD2 MODELS WITHOUT AERIAL SURVEYS USING MACHINE LEARNING. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W8:27–34, 2019. doi:10.5194/isprs-annals-IV-4-W8-27-2019. URL https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-4-W8/27/2019/.

F. Biljecki, H. Ledoux, and J. Stoter. Height references of CityGML LOD1 buildings and their influence on applications. 11 2014.

F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin. Applications of 3D city models: state of the art review. *ISPRS International Journal of Geo-Information*, 4:284–2889, 2015. doi:10.3390/ijgi4042842.

F. Biljecki, H. Ledoux, and J. Stoter. An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59:25 – 37, 2016a. ISSN 0198-9715. doi:https://doi.org/10.1016/j.compenvurbsys.2016.04.005. URL http://www.sciencedirect.com/science/article/pii/S0198971516300436.

F. Biljecki, H. Ledoux, J. Stoter, and G. Vosselman. The variants of an LOD of a 3D building model and their influence on spatial analyses. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116:42 – 54, 2016b. ISSN 0924-2716. doi:https://doi.org/10.1016/j.isprsjprs.2016.03.003. URL http://www.sciencedirect.com/science/article/pii/S0924271616000605.

F. Biljecki, H. Ledoux, and J. Stoter. Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, 64:1–18, 2017. doi:10.1016/j.compenvurbsys.2017.01.001.

R. Boeters. Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination. Master's thesis, Delft University of Technology, 2013. URL https://repository.tudelft.nl/islandora/object/uuid:b22a2b93-4a0a-4aa7-8e3b-6e08e0027634?collection=research.

R. Boeters, K. A. Ohori, F. Biljecki, and S. Zlatanova. Automatically enhancing CityGML LOD2 models with a corresponding indoor geometry. *International Journal of Geographical Information Science*, 29(12):2248–2268, 2015. doi:10.1080/13658816.2015.1072201. URL https://doi.org/10.1080/13658816.2015.1072201.

B. Dukai, H. Ledoux, and J. Stoter. 3D Registration of Buildings and Addresses (BAG), 2018. URL http://3dbag.bk.tudelft.nl.

B. Dukai, H. Ledoux, and J. Stoter. A Multi-Height LoD1 Model of all Buildings in the Netherlands. In *14th 3D GeoInfo Conference 2019*, volume IV-4 of *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 51–57. ISPRS, 2019. doi:10.5194/isprs-annals-IV-4-W8-51-2019.

A. Géron. *Hands-on machine learning with scikit-learn, keras, and tensorflow*. O'Reilly Media, Inc., 2019.

G. Gröger and L. Plümer. CityGML – Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, 71:12–33, 07 2012. doi:10.1016/j.isprsjprs.2012.04.004.

E. Heeres. Exploring the 3D BAG: How to define it and to what extent can it automatically be created using open data. Master's thesis, Delft University of Technology, 2016. URL https://repository.tudelft.nl/islandora/object/uuid:bb4a1667-1d6f-41ac-b12e-fb70c0013881.

I. Lánský. Height inference for all USA building footprints in the absence of height data. Master's thesis, Delft University of Technology, 2020. URL https://repository.tudelft.nl//islandora/object/uuid:ddcae7d1-6cc8-42a7-8c1d-a922ec7551f0.

K. Lwin and Y. Murayama. A GIS Approach to Estimation of Building Population for Micro-spatial Analysis. *T. GIS*, 13:401–414, 08 2009. doi:10.1111/j.1467-9671.2009.01171.x.

A. C. Müller and S. Guido. *Introduction to Machine Learning with Python*. O'Reilly Media, Inc, 2016.

OGC. OGC City Geography Markup Language (CityGML) Encoding Standard 2.0.0. Technical report, Open Geospatial Consortium., 2012.

S. Shiravi, M. Zhong, S. A. Beykaei, J. D. Hunt, and J. E. Abraham. An assessment of the utility of LiDAR data in extracting base-year floorspace and a comparison with the census-based approach. *Environment and Planning B: Planning and Design*, 42(4):708–729, 2015. doi:10.1068/b130144p. URL https://doi.org/10.1068/b130144p.