Technical University of Munich

Chair of Transportation Systems Engineering

Master's Thesis

Predicting and measuring venue popularity using crowd-sourced and passive sensor data

Authored by Stanislav Timokhin

Supervised by Univ.-Prof. Dr. Constantinos Antoniou

30 October 2018

Abstract

It is hard to underestimate the importance of transport planning and general research related to people mobility patterns. A lot of current research in this field relies heavily on data. However sometimes data availability issues due to system properties or some endogenous factors may limit study potential. Therefore, it was decided to discover the possibilities of use of auxiliary information sources that received limited attention previously.

A methodology to retrieve and predict data available for public and related to mobility patterns (i.e. shares of people attending particular venue from Google "Popular Times" section of maps) was developed and tested. Several sources were used in this study: Google Maps, Yelp, OpenStreetMap, Google API, government data on workplaces and population.

Certain scripts were developed for information retrieval and filtering for each data source. Additional procedures were developed to prepare highly aggregated data for use in prediction models. Special procedure was developed for combining venue specific and spatial data, which involved spatial operations (intersects/within) and spatial indexing to increase speed of spatial operations.

Clustering algorithm was developed for data exploration part. The algorithm is based on visual exploration of data projection with reduced dimensionality that is achieved with the help of t-SNE method.

Two classes of prediction models with and without transformation of dependent variables were tested: linear regression with lasso regularization and gradient boosted regression (GBR). Each model group tested consisted of 168 dependent variables (i.e. number of hours in a week), number of place parameters (like rating, number of related comments, type of service provided) and locational properties (like number of stores, hotels, attractions etc. nearby).

In general, it was found that prediction power of both classes of models increased with transformation of dependent variable.

GBR models with applied transformations were better, comparing with linear ones. In at least 50% of cases the difference is relatively low (R^2 difference of 0.02), increasing higher than 0.20 for certain hours.

As Google "Popular Times" data defines only venue shares, microcontroller setup to measure actual number of people attending particular venue by WIFI device presence detection was developed and tested. Real world tests showed that such setup is useful in practice and could be recommended in future research.

Declaration of Authorship

I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here.

This paper was not previously presented to another examination board and has not been published.

Munich, 30 October 2018

Stanislav Timokhin

Table of contents

List of tables

List of figures

1. Introduction

A lot of work have been done in travel demand modeling domain and transport planning. However, it is possible to say that data availability is one of the limiting factors of research. Availability of data depends on several factors for example on price associated with its collection (survey or sensor installation) or purchase from service providers, privacy issues and lack of certain types of data (electronic ticket validation in transport).

Currently available data sources may be also limited, especially in terms of social or recreational activities, that are not very well represented in traditional travel surveys. On the other hand, some more detailed data may be unavailable, extremely aggregated or expensive with mobile phone meta data as an example.

It is necessary to mention that there is some research of alternative data sources that could be used as a complimentary to traditional travel surveys or traffic data and could be retrieved at practically no cost. One of the most important alternative data sources are location based social networks (LBSNs) like Twitter, Foursquare, Facebook etc.
A lot of research is based on Foursquare and Twitter data for ex. (Gu, Qian et al. 2016). Other sources are less popular, mostly because of their less opened application programming interfaces (APIs) or their irrelevance to some planning tasks. However in (Chaniotakis, Antoniou et al. 2016) authors tried to investigate relationships of traditional travel survey and data from several sources like Foursquare, Facebook and Twitter. Some basic research was also done with the use of Google “Popular Times” where researchers tried to correlate “Popular Times” info with traffic and other auxiliary data see (Tafidis, Teixeira et al. 2018). It was also referenced in (Nunes, Ribeiro et al. 2017).

2. Literature review

Topics related to prediction of venue popularity patterns and, in particular, number of people attending exact venue have received limited attention in research (at least available for public). However, several fields may be considered as relevant.

In (Kisilevich, Keim et al. 2013) authors used geospatial data from OpenStreetMap (OSM)[1] as well as business features to predict hotel room prices. In the review of related research, distance to city center was considered as quite important variable in determining room rates in significant part of mentioned articles.
In the proposed model, apart from parameters directly related to hotels, authors used museums, historical places, places of worship, transportation, restaurants and pubs.

In (Wang, Gopal et al. 2015) authors analyzed the influence of Foursquare check-ins on business failure[2]. Several features from Yelp and Foursquare were studied, including business features (price range, rating, number of direct competitors within certain area, number of special promotions of business and competitors within certain area) and check-in data (average daily check-ins of business and neighbors, growth rate and number of days growth rate increased). Authors found out that "increase in a restaurant's average daily check-in growth rate and the number of days that the growth rate has increased are associated with a significant decrease of the odds of failure [i.e. business]". They also found out that rating was positively correlated with failure which was attributed to higher business costs and therefore lower profit margins. Interaction between rating and number of competitors within area was also significant. And finally, the more competitors there were nearby and the higher business price range the less likely is failure.

Quite interesting study was presented in (Willing, Klemmer et al. 2017) where authors researched factors that influence car sharing usage and predicted areas usage density. They used point of interest (POI) data from Google API as well as data from car sharing provider. Prediction "without the POI" … "yielded a 50% hit rate" comparing to 80% with POI. Gradient boosting was used to select important variables which were used to build linear regression.

[1] For details see http://openstreetmap.org/ and Methodology part
[2] Defined by authors as less than one check-in per day in Foursquare

From review of current research related to bicycle and car sharing systems in (Rodas 2017), taking into account that transport demand is induced by various activities, we may get some insights of factors that could potentially influence venue demand patterns. Majority of studies in the review considered population density (7 of 9), job density was considered only in 2 studies, possibly because of data availability reasons, also location of public transport stations / sharing stations and lengths of roads were used in several papers, as well as numbers of possible activities available and land use type.

Venue popularity research with Foursquare only data in (Li, Steiner et al. 2013) explored 3 aspects: venue profile information, venue category, and venue age (date of profile creation in Foursquare). The main findings include that popularity of an older venues is higher comparing to newer ones (with some exceptions), more complete profiles suggest higher popularity, most commented (tipped) venues belong to food category; as well as transport category (for ex. airports) have most check-ins.

Big slice of literature is devoted to venue recommendation systems, with some papers developing prediction mechanisms of venue appropriateness to a specific user or group of users depending on the context[3] (like user's location, time of day), user and venue features, see for ex. (Deveaud, Albakour et al. 2015, Manotumruksa, Macdonald et al. 2016). Article (Noulas, Scellato et al. 2012) suggests several "global mobility features" that are used in next place prediction: popularity (total number of venue check-ins), geographic distance, rank distance (measures relative density of venues that are closer to user than proposed destination), activity transitions (for ex. work –> supermarket) and place transitions. This article also mentions that popularity of a place as well as geographic distance are quite important factors of user decision.

Another direction of studies with high business focus is presence detection and tracking systems (location analysis systems) for example monitoring of shopping and street activity as well as vehicle traffic monitoring to name a few. One of the main reasons of using these technologies by commercial entities is the improvement of corporate decision making especially related to spatiotemporal analysis, consumer behavior analysis and estimating marketing campaigns effectiveness. On the other side, city planning organizations and authorities may also benefit from these technologies as they may improve data provided by local conventional sensors, like

[3] Context – any information that can be used to characterize the situation of an entity that is considered relevant to the interaction between a user and an application

for example loop detectors, with better understanding of traffic flow, faster accident and congestion detection and provide better inside for city planning and event management.

In general it is possible to classify users of spatiotemporal data as governmental and non-governmental (Meeks and Dasgupta 2004). Apart from transport such data is useful in a variety of areas. According to (Garber 2013) the most common types of location analytic applications include:

- finding the best place to locate stores, warehouses, cell towers, or other structures
- identifying high- and low-performing stores
- allocating and recruiting staff
- targeting sales and marketing efforts to different regions
- offering products and prices most suitable for specific areas
- managing insurance risk based on the potential of disasters in given locations
- streamlining supply chains, shipping, and distribution
- analyzing the competition
- network design
- enhancing disaster forecasting and emergency preparedness
- asset and customer-relationship management
- urban planning
- evaluating crime data to focus law enforcement efforts in problem areas
- tracking infectious diseases

There are a lot of papers related to location analytics with the help of different wireless technologies like WIFI[4] and Bluetooth[5] as well as video recognition systems. GSM data is also used for similar purposes however it is not relevant for this study as data is usually coarse due to technological and privacy reasons plus the significant price of it.

In conference paper (Abrishami, Kumar et al. 2017) authors collected data with WIFI monitoring devices in over 100 places in USA and used it to predict actual foot traffic for the next 168 hours (week). Data from past traffic observations was used to predict future states. Some factors that

[4] WIFI – wireless local area networking technology based on IEEE 802.11 standards
[5] Bluetooth – wireless personal area networking technology for exchanging data over short distances

may affect results were also mentioned (although only holiday effects were implemented), dependence on location was also checked.

Previous article also mentioned study (Cortez, Matos et al. 2016), where researchers used camera and facial recognition systems to “detect foot traffic to a sports store” and then used time series to predict traffic up to one week ahead. Number of additional factors was also quite limited.

In (Yoshimura, Krebs et al. 2016) authors used passive Bluetooth sensing data to analyze museum visitors behavioral patterns. However, it is possible to say that this approach had certain drawbacks. And the main one is the fact that it is possible to detect only mobile devices with Bluetooth turned on. According to estimates mentioned in paper, averagely, 8.2% of visitors had this function activated. This results in possible biases as mentioned in paper. On the other hand, sensor setup at entrance/exit points as well as inside museum building let them collect relatively reliable data on length of stay.

In other paper (Nunes, Ribeiro et al. 2017) authors were using WIFI tracking technology to analyze tourist mobility patterns. In this research authors had ground truth data from tourist authorities and were able to correlate it with sensor data and, for certain places, with Google “Popular Times”. Their results showed strong correlation between ground truth and sensor data, as well as quite high correlation between sensor data and Google “Popular Times”. The ratio of people to detected devices was “approximately 2 at the airport and more than 5 in the other POIs”, that is 50% and 20% respectively.

It is also interesting to note that some commercial companies advertise even 95% detection rates at certain conditions (Libelium).

3. Research goal

In this paper author will try to investigate correlations of Google “Popular Times” data with actual place attendance and develop a model that would predict share of visitors at certain point of time based on object parameters like rating, price range, location and other factors.

For measuring of actual place attendance author will develop cost effective microcontroller setup as well as write a software interface necessary for a desired task.

This could be helpful in various fields be it traffic demand modeling, accessibility analysis and even more complex tasks either public or commercially related.

4. Methodology

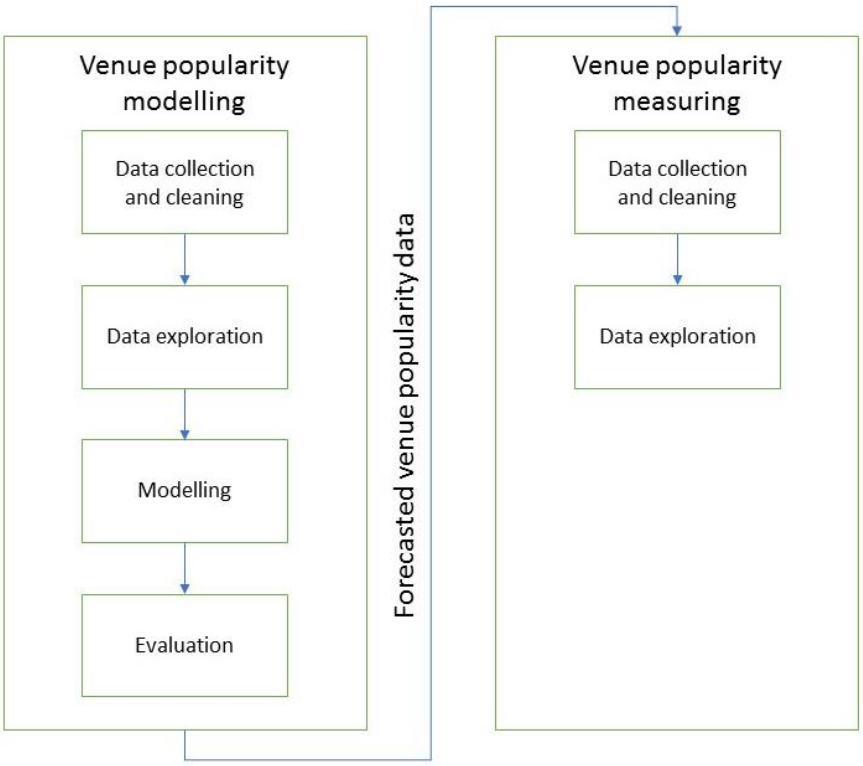

Figure 1: Project workflow (flow-chart).

This project consists of two significant parts: venue popularity modelling and venue popularity measuring. Although in terms of measuring of venue popularity only exploratory analysis is effected, in future, when enough data is collected and additional research is performed, it could be possible to develop a model of venue occupancy, that, with the help of forecasted venue popularity, may provide "real" occupancy numbers for all venues without necessity in additional measurements.

4.1. Venue popularity modelling

4.1.1. Data collection and cleaning

Dependent variables should be defined as occupancy level per hour, as there are 168 hours in a week, it is necessary to introduce 168 dependent variables.

Independent variables should include business factors related to exact venue (for ex. rating, type etc.) and spatial variables (for ex. number of stores nearby).

Significant amount of spatial information comes from OpenStreetMap (OSM).
OSM is built by international community of mappers that contribute and maintain data about roads, trails, cafes, railway stations and other geospatial objects.
The main advantage of OSM is opened data. Anyone is free to use it for any purpose as long as the license terms are not altered.

It is possible to interact with OSM directly via API. There are two of them, either special Editing API or read-only Overpass API. Majority research related task require only read-only access, so Overpass API is sufficient and acts as a database backend for various applications.
It is also possible to download extract from third party web site, for example from Geofabrik GmbH (www.geofabrik.de/data/ download.html) that is more efficient and helps to minimize load on external server.

OSM objects have 3 basic elements:

- Node – defines point in space, each node should have at least id number and two coordinates
- Way – an ordered set of nodes that is used to represent linear features and area boundaries
- Relation – multipurpose data structure that documents the relationship between two or more data elements i.e. explains how elements work together

Due to the fact that the number of features in OSM is high and large amount of them is irrelevant to the study (for ex. fire hydrant availability), has poor coverage (for ex. availability of street lamps) or repeats for different types (i.e. nodes may denote similar objects as ways) some

manual data selection is advised. This will as well help to slightly mitigate the "curse of dimensionality" by adding more relevant data.

Spatial information about population and workplaces is usually available from government organizations. However, as such data is usually aggregate, additional actions could be necessary to distribute it properly among administrative areas (see disaggregation algorithm below).

Algorithm: Data disaggregation
Data:
Areas of buildings within each administrative area, ar
People per square meter per place type, $psqr$
Number of people per administrative area, p
Result: Number of people per building
begin
For area **in** All administrative areas **do**
    For building **in** area **do**
    Get number of relevant nodes/ways, n
    Get sum of $psqr$ for all places (nodes/ways), $spsqr$
    Get building multiplier, $m = spsqr/n$
    end
    Sum all multipliers for an area, sm
    Get number of people per building in an area, $p \cdot ar_i \cdot m/sm$
end

Spatial variables should be computed based on venue location and chosen depending zone parameters i.e. number/length of attributes within or intersecting designated area.

In order to speed up calculations in large datasets, as spatial relation operations like intersection/within are computationally intensive, it is necessary to construct spatial index. R-tree index, that was introduced by (Guttman 1984), was used in this study. The main rationale behind its introduction was the fact that "classical one-dimensional database indexing is not appropriate for multidimensional data" as well as that "structures based on exact matching of values like hash tables are also not very useful because range search is required" and "structures using one dimensional ordering (like B-trees) do not work properly because of multidimensional search space." Although some algorithms developed over B-trees were suitable for two-

dimensional data, R-tree is more universal approach that is implemented in various software libraries.

R-tree is a hierarchical data structure that is used to dynamically organize set of n-dimensional objects and represent them by minimum bounding boxes. Each node may have several children objects in it. Leaves of the tree contain pointers to the database objects instead of children nodes (Manolopoulos, Nanopoulos et al. 2010). The structure is designed so that a spatial search requires visiting only a small number of nodes (Guttman 1984). It is also necessary to note that usage of bounding boxes may produce false positive results, therefore all candidates should be inspected.

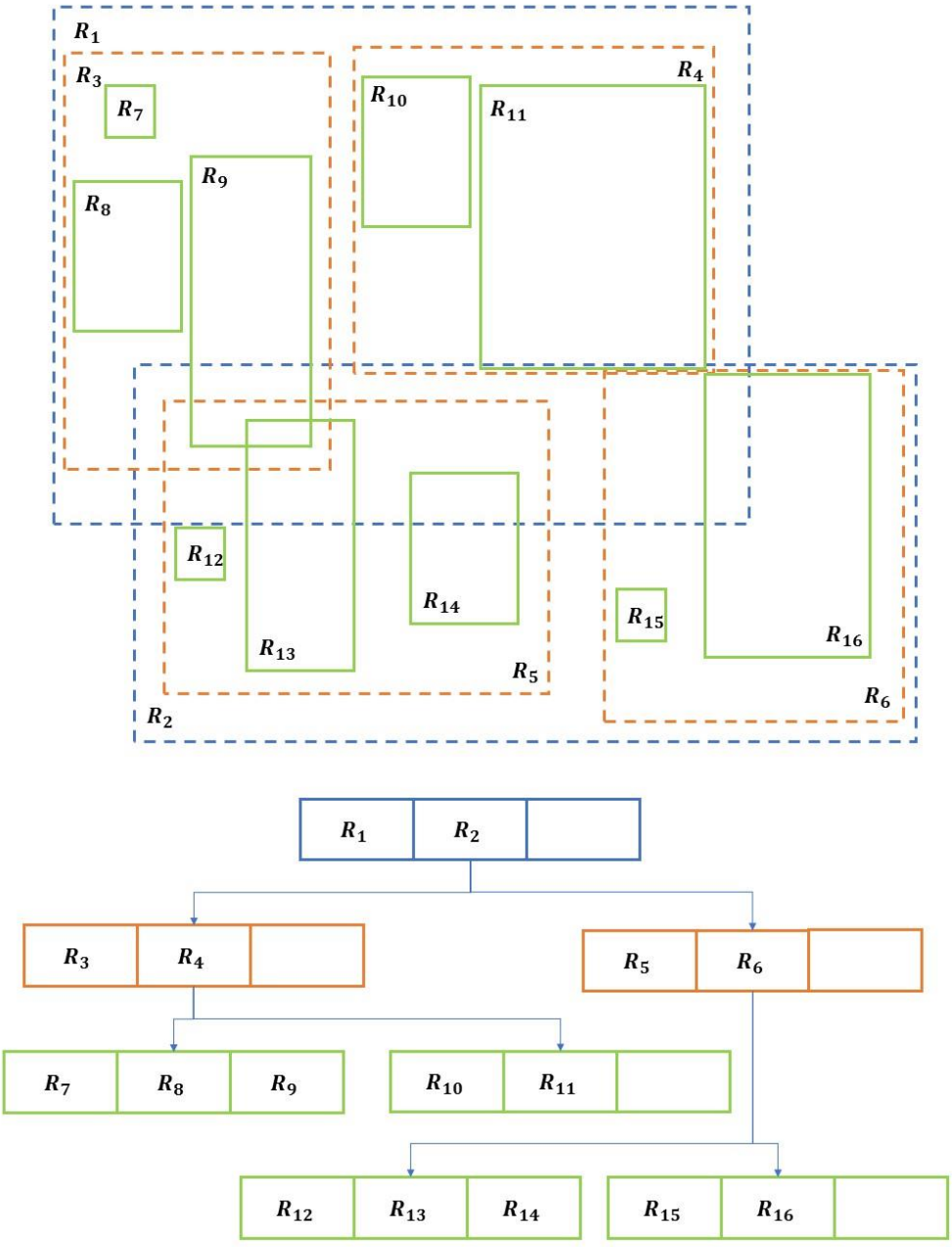

Figure 2: R-tree structure.

Depending zone

Different approaches are used in literature to define zone of influence. Among them Voronoi diagrams that divide planes according to the nearest-neighbor rule: Each point is associated with the region of the plane closest to it (Aurenhammer 1991).
Simple buffers based on Euclidean distance from a point are also quite commonly used in a literature, see review in (Rodas 2017).

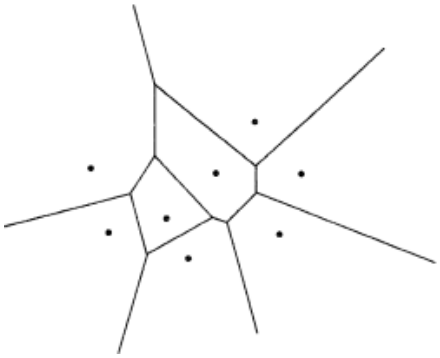

Figure 3: Example of Voronoi diagram (Aurenhammer 1991).

As spatial resolution in this study is relatively low, it was decided to use method similar to buffering, but with real walking distances. This could provide much better results for irregular shaped areas and for areas containing natural boundaries like rivers.
This is done with OSM road graph, with edge values equal to distances. Subset of edges is selected based on defined walking distance. Afterwards this subset is used to calculate shape of influence area (convex hull).

As some authors suggest that "definition of locational properties is usually an ill-structured problem" (Kisilevich, Keim et al. 2013), testing several influence distances is advised.

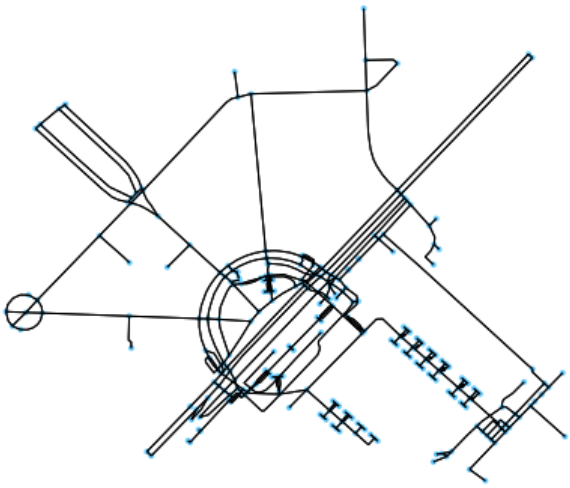

Figure 4: Part of road graph belonging to an area.

4.1.2. Data exploration

General data description

Description of the data with the help of basic statistics. Discussion of some qualitative parameters.

Clustering

In order to explore and understand the data it was decided to first group venues into clusters.

Clustering algorithm

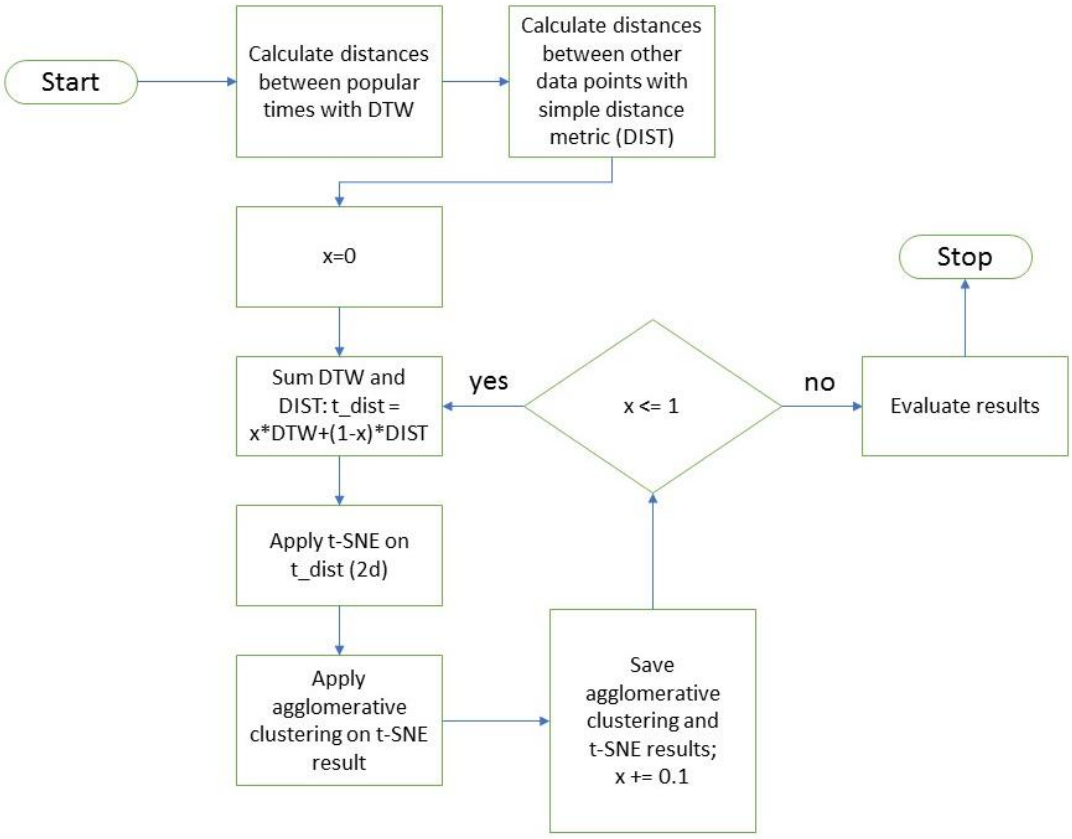

Figure 5: Clustering algorithm (flow-chart).

1. Calculate similarities between "Popular Times" sequences with Dynamic time warping (DTW) metric that is used in time series analysis to measure similarity between sequences that are mapped in a non-linear fashion. DTW is quite robust distance measure for time series, that allows to match similar shapes even if they are out of phase.

The objective of DTW is to compare two time-dependent sequences $X: (x_1, x_2, \dots, x_n)$ of length $N \in \boldsymbol{N}$ and $Y: (y_1, y_2, \dots, y_m)$ of length $M \in \boldsymbol{N}$. These should be sequences sampled at equidistant points in time.
Then matrix of size n by m is constructed where each point (i,j) is a distance between values x_i and y_j. Typically, if this distance is small, x and y are similar to each other and vice versa.
By calculating all distances cost matrix is formed. Then the goal is to find an alignment between X and Y with minimal overall cost.

$$DTW(X,Y) = min\sqrt{\sum_{k=1}^{K} w_k}$$

where w_k is a k-th element of alignment between X and Y i.e. $(i,j)_k$

A lot of research has been done on the topic of DTW performance.
Original DTW has $O(n^2)$ complexity or in fact $O(n \cdot m)$ if sequences have different lengths, however there are some techniques for ex. LB_Keogh (Keogh and Ratanamahatana 2005) and LB_Improved (Lemire 2009) that reduce complexity close to $O(n)$.

Several constraints may also be useful to limit DTW search complexity:

- Monotonicity – backward movements are not allowed
- Continuity - only one step at a time is possible
- Boundary conditions - first and last elements of X and Y are aligned to each other
- Warping windows - movement is restricted to a certain distance from diagonal. Apart from limiting complexity it takes into account some physical properties of real world. For example, event may be considered similar if it happens within two hours from current event.
- Slope constraint - path cannot be too steep or too shallow

2. Calculate distances between other data points with traditional Euclidean distance metric.

3. Use sum of the above distances as an input to clustering algorithm instead of original data.
4. Use t-distributed Stochastic Neighbor Embedding (t-SNE) algorithm (Maaten and Hinton 2008) to visualize the data and check for patterns.

There are several linear dimensionality reduction algorithms available for high-dimensional data representation (for example Principal component analysis (PCA)). However, they may miss some non-linear structure in the data. Several non-linear methods exist as well (for example Isomap, Local linear embedding (LLE), t-SNE). Unlike Isomap, LLE and its variants, t-SNE is sensitive to local structure of the data that could be useful to get distinct data clusters (Pedregosa, Grisel et al.). Main disadvantage of t-SNE is high computational intensity, though it is less important in current study due to moderate number of samples.

t-SNE gives each observation point a location in 2-dimensional or 3-dimensional space. This technique is a variation of stochastic neighbor embedding (SNE) that also reduces crowding of points in the center of the map comparing to SNE.

SNE converts high-dimensional Euclidean distances between data points into probabilities that represent similarities. The conditional probability that x_i would be x_j neighbor $p_{i|j}$ is calculated with the formula below

$$p_{i|j} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{i \neq j} exp(-||x_i - x_j||^2/2\sigma_i^2)}$$

where σ_i is the variance of the Gaussian that is centered on data point x_i. As we are only interested in modeling pairwise similarities, $p_{i|i} = 0$.
It is also possible to compute similar conditional probability for low-dimensional points which may be called x_i, x_j projections – y_i, y_j.

$$q_{i|j} = \frac{exp(-||y_i - y_j||^2)}{\sum_{i \neq j} exp(-||y_i - y_j||^2)}$$

where the Gaussian is $1/\sqrt{2}$ and $q_{i|i} = 0$.

If the y_i and y_j similarity to points x_i and x_j is correctly modeled the conditional probabilities $p_{i|j}$ and $q_{i|j}$ will be equal.

The SNE method tries to find a low-dimensional data projection with the difference between $p_{i|j}$ and $q_{i|j}$ is minimum. The fit of $q_{i|j}$ to $p_{i|j}$ is measured with the Kullback-Leibler divergence (Kullback and Leibler 1951). With the help of gradient descent method SNE minimizes the sum of Kullback-Leibler divergences over all data points. The cost function C is calculated with formula below

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{i|j} log(\frac{p_{i|j}}{q_{i|j}})$$

where P_i is the conditional probability distribution over all other data points for x_i value, and Q_i is the conditional probability distribution over all other projected data points for y_i.

Due to the fact that Kullback-Leibler divergence is not symmetric, different types of error in the pairwise distances in the low-dimensional projection are not weighted equally. That means that cost for closely located projected points to represent distant points is relatively low, comparing with high cost of representation of closely located data points with distant projected points.

To get σ_i value, that results in a P_i with perplexity specified by the user, binary search is performed by SNE. The perplexity is calculated with formula below

$$Perp(P_i) = 2^{H(P_i)}$$

where $H(P_i)$ is the Shannon entropy (Shannon 1948) of P_i measured in bits

$$H(P_i) = -\sum_j p_{j|i}\, log_2(p_{j|i})$$

The perplexity may be considered as a smooth measure of the effective number of neighbors. Larger datasets usually require larger perplexity. Generally, it is recommended to use values between 5 and 50.

To deal with a “crowding problem”, when even quite small attractive forces put together relatively dissimilar points in the center of the plot t-SNE method was introduced. It is

necessary to note that such problem may arise in SNE and other local techniques for example in Sammon mapping (Sammon 1969).

Instead of conditional probabilities $p_{i|j}$ and $q_{i|j}$ t-SNE uses joint probability distributions P and Q. It optimizes symmetric version of SNE with cost function

$$C = \sum_i KL(P||Q) = \sum_i \sum_j p_{ij}\, log\left(\frac{p_{ij}}{q_{ij}}\right)$$

with

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

and

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{i \neq j}(1 + ||y_i - y_j||^2)^{-1}}$$

gradient is equal

$$\frac{\partial C}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})\,(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}$$

Use of heavy-tailed student t-distribution with one degree of freedom allows modelling relatively moderate distances in multidimensional space with larger distances in projections. That helps to get rid of unwanted attraction forces between projection points representing relatively dissimilar data points.

Algorithm: Simple version of t-Distributed Stochastic Neighbor Embedding
Data: data set $X = \{x_1, x_2, \ldots, x_n\}$
cost function parameters: perplexity Perp,
optimization parameters: number of iterations T, learning rate η, momentum α(t).
Result: low-dimensional projection $Y^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
compute pairwise affinities $p_{j|i}$ with perplexity Perp.

Set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$

Generate initial solution $Y^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $N(0, 10^{-4}I)$

For t=1 **to** T **do**

- compute low-dimensional affinities $q_{ij} = \frac{(1+||y_i - y_j||^2)^{-1}}{\sum_{i \neq j}(1+||y_i - y_j||^2)^{-1}}$
- compute gradient $\frac{\partial C}{\partial y_i} = 4\sum_j (p_{ij} - q_{ij})\,(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1}$
- set $Y^{(t)} = Y^{(t-1)} + \eta\ \frac{\partial C}{\partial y_i} + \alpha(t)(Y^{(t-1)} - Y^{(t-2)})$

end

5. Apply agglomerative hierarchical clustering on t-SNE data points (two-dimensional array). Set connectivity parameters to k-neighbors graph result (weighted graph of k-neighbors for points in array).

   Several clustering techniques were checked on test example of two-dimensional array with agglomerative clustering providing best results due to the use of connectivity matrix, restricting assignment to a particular cluster. Moreover, as number of clusters is decided on visual analysis, techniques that do not define exact number are of little use.
   In fact, it is possible to perform this operation by hand as general procedure is not fully automated.

   Hierarchical clustering is a family of cluster analysis techniques that construct clusters based on two general approaches: agglomeration and division. In agglomerative clustering cluster is initialized for each observation that are merged successively. Divisive clustering on the other hand starts with one cluster and splits it recursively. The hierarchy of clusters is represented by tree or dendrogram, with one cluster in a root of a tree representing all others.

   In order to decide which clusters should be merged in agglomerative approach, or which clusters should be split in divisive clustering, it is necessary to provide some measure of similarity between sets of observations. This is usually achieved by an appropriate metric (distance measure between pairs of observations, for example Euclidean) and linkage criterion, which specifies similarity between sets of observations as a function of the pairwise distances between observations.

Metric choice may influence shape of clusters as distances between multidimensional points may vary depending on metric selected.

Some widely used linkage criteria include:

- Ward linkage – minimizes the sum of squared differences within all clusters. It is a variance-minimizing approach and in this sense, is similar to the k-means objective function but tackled with an agglomerative hierarchical approach.
- Maximum (complete) linkage – minimizes the maximum distance between observations of pairs of clusters.
- Average linkage – minimizes the average of the distances between all observations of pairs of clusters.
- Single linkage – minimizes the distance between the closest observations of pairs of clusters.

Quite interesting feature of agglomerative clustering is the possibility to add connectivity constrains with the help of connectivity matrix that defines neighbors for each observation (only close points should merge). This helps to avoid forming distributed clusters. Connectivity matrix may be constructed with some prior information (for example cluster venues depending on some rating threshold) or could be learned from data, for example with k-nearest neighbors algorithm (k-NN) (Dudani 1976).

Agglomerative clustering is quite hard algorithm in terms of computational complexity, however when used with connectivity matrix it may be scaled to relatively large problems.

Algorithm: Agglomerative clustering
Data: data set $X = \{x_1, x_2, \ldots, x_n\}$
Connectivity matrix $M_{n\,x\,n}$
Result: X with assigned clusters
begin
Assign each point x_i to separate cluster
Calculate distances (similarity) between every pair of observations (precomputed distance matrix)
While number of clusters **< N do**

- Group neighboring (according to connectivity matrix) clusters into a single cluster with the help of distances calculated in previous step.
- Remove the pair of clusters that were merged from matrix
- Calculate distances from new cluster to every other cluster and add them into matrix

end

6. Evaluation of clustering algorithm results with the best performing internal measurement metrics according to (Hassani and Seidl 2017) and/or by inspection of their graphical representation.

   There are two types of clustering evaluation metrics: external and internal.
   External metrics compares clustering results to the reference, therefore it is useful when ground truth about data is available, that is rarely the case in practice (Hassani and Seidl 2017).
   Internal metrics checks the structure of found clusters and their relations. It is based on assumed "goodness" of cluster structure, that is mostly defined by two measures (or their variations): compactness and separation. Compactness measures closeness of cluster elements. It is usually measured with variance (average distance to the cluster center). Separation measures the difference of clusters. Quite often it is measured with distances between cluster centers or between objects contained in different clusters.
   As it comes from measures definitions they may not perform well when it is difficult to get clusters that are perfectly separated from each other, therefore, in some cases visual inspection is advised.

4.1.3. Modelling

In order to get a model, certain operations need to be performed. As was recommended previously, testing of several depending zone values is included into modelling procedure.

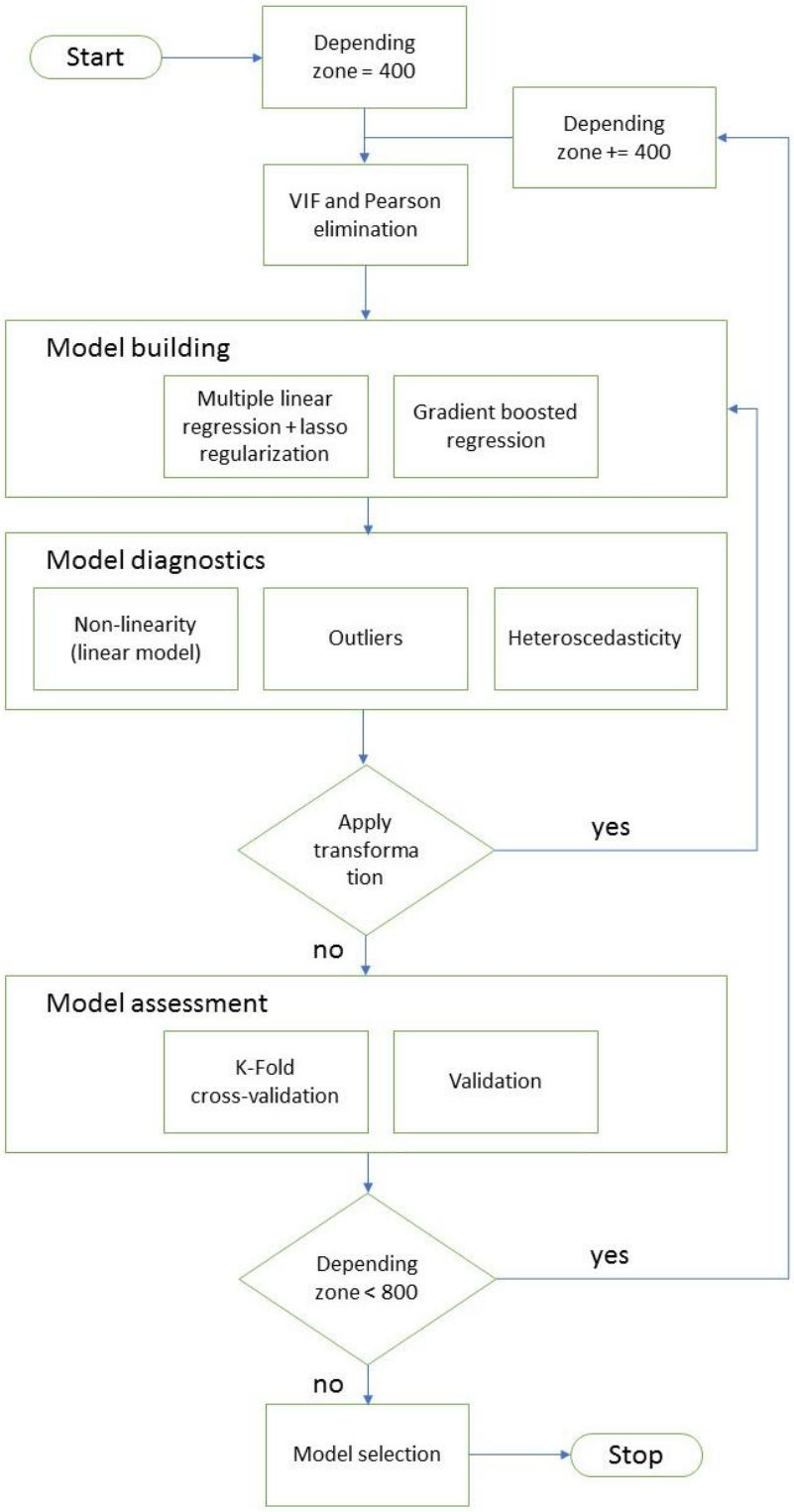

Figure 6: Modelling procedure (flow-chart).

Please note that “VIF and Pearson elimination” section is included into “Model diagnostics”.

Model building

In this study, two modelling approaches will be considered: multiple linear regression and decision tree method to check if non-linear predictor will perform better i.e. if behavior of model is non-linear. Both classes of models performed quite well in previous studies, for example see (Rodas 2017).

Multiple linear model

It is an approach to model relationship between independent variables X and response variable Y.

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n + \varepsilon$$

Where β_0 is called intercept, ε is an error term, $X_1 \ldots X_n$ are independent variables and Y is dependent variable. Same equation in vector form

$$Y = \beta X + \varepsilon$$

Where β is (n+1)*1 vector and X is k*(n+1) matrix.

Coefficients $\beta_0, \beta_1 \ldots \beta_n$ are estimated with ordinary least squares (OLS). OLS minimizes residual sum of squares.

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

This is one of the most common approaches used in literature approaches, see review in (Rodas 2017). As it has simple structure and is relatively easy to interpret.

However, the fact that the number of variables in this study is relatively high comparing to the number of samples, may result in model overfitting and poor predictions. It could therefore be necessary to reduce dimensionality and/or shrink coefficients, and as a result increase prediction accuracy, possibly improve model interpretability and mitigate multicollinearity. Among feasible approaches for this task are:

Subset selection:

Stepwise regression is most common here. It is relatively computationally intensive (comparing with least squares) and cannot guarantee that the best model is selected. Also, it may not perform well with highly correlated variables.

Moreover, as mentioned in book (Friedman, Hastie et al. 2001) "by retaining a subset of the predictors and discarding the rest, subset selection produces a model that is interpretable and has possibly lower prediction error than the full model. However, because it is a discrete process – variables are either retained or discarded – it often exhibits high variance, and so doesn't reduce the prediction error of the full model. Shrinkage methods are more continuous, and don't suffer as much from high variability".

Shrinkage (regularization):

Regression with ridge and/or lasso regularization. Such methods are computationally effective, almost identical to least squares estimation. Ridge regression selects all variables, and while it may not be a problem for model accuracy, it is harder to interpret it.

$$RSS + \alpha \sum_{j=1}^{p} {\beta_j}^2 ,$$

where α is a regularization parameter that should be chosen separately. Regularization parameter is small when coefficients β_j are close to zero, that effectively shrink them towards zero.

Lasso on the other hand may significantly reduce number of variables (depending on regularization parameter), although being more restrictive. And therefore, is much easier to interpret.

Lasso regression is an extension of OLS with and addition of slightly different to ridge regression regularization term to an optimization objective. So, if α is sufficiently large, some coefficients estimates become equal to zeroes.

$$RSS + \alpha \sum_{j=1}^{p} |\beta_j|$$

Regularization parameter controls the impact of regularization on model coefficients. It is necessary to note that it should not be applied to intercept.

Correct regularization parameter (α) may be chosen with cross-validation procedure. For each fold MSE is calculated for different α, final α is chosen as average optimal value for all folds (see model diagnostics section for cross-validation description).

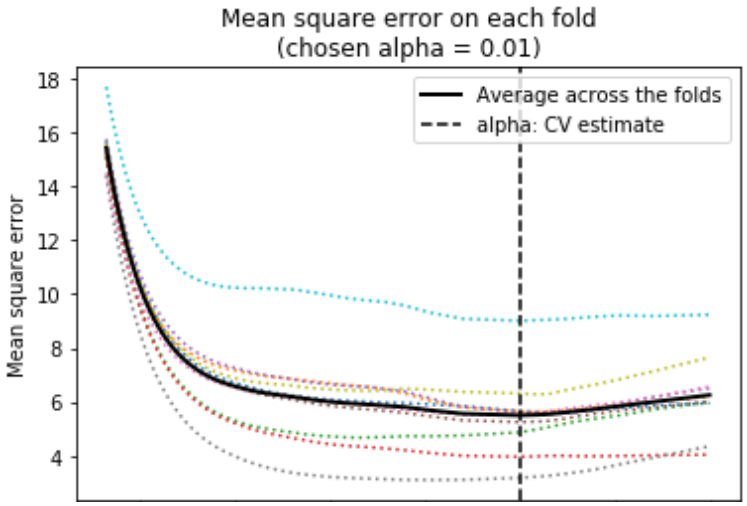

Figure 7: Regularization parameter (α) selection with cross-validation.

Dimensionality reduction:

PCR is quite popular approach to dimensionality reduction, closely related to ridge regression. One can even think of ridge regression as a continuous version of PCR (Friedman, Hastie et al. 2001).

Model building procedure:

1. Select model (Lasso).
2. Run model with cross-validation procedure that selects α for each output (total 168 alphas).

Decision trees

Set of splitting rules used to segment the predictor space into number of simple rectangular regions that can be summarized in a tree. Usually mean or mode of simple region is used to predict output.

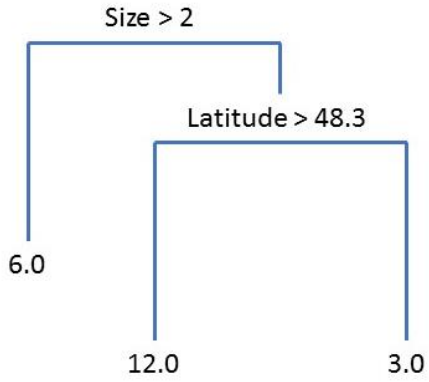

Figure 8: Example of a decision tree.

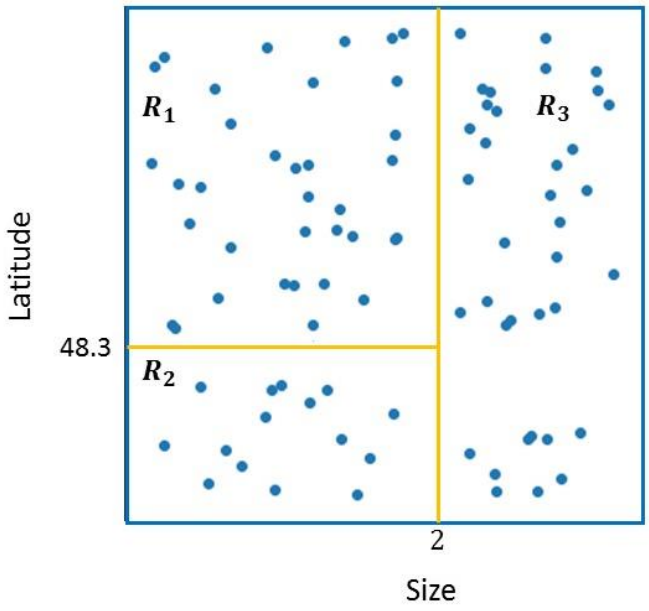

Figure 9: Example of a decision tree partition.

As a first step predictor space is divided into j distinct non-overlapping rectangles (boxes) $R_1 \dots R_j$ that minimize RSS

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})$$

where $\hat{y}_{R_j}$ is the mean of observations within j-th box. As in most cases it is computationally infeasible to consider all possible partitions, greedy top-down approach – recursive binary splitting is used. It starts from a single region and splits it into two at each iteration. The best split is made at a particular step without looking ahead.

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1}) + \sum_{i:x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})$$

The process continues until stopping criteria is reached.
In order to deal with complex tree shape and therefore overfitting tree pruning technique is used. First complex tree T_0 is built and then it is pruned back to obtain subtree. Quite efficient pruning method is cost complexity pruning or weakest link pruning, where sequence of trees indexed by non-negative α parameter is considered. For each value of α there is a subtree $T \subset T_0$ that minimizes equation below.

$$\sum_{m=1}^{|T|} \sum_{i:x_i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2 + \alpha|T|$$

Where $|T|$ is a number of terminal nodes of tree T, R_m is a box of m-th terminal node and $\hat{y}_{R_m}$ is a mean of training observations in R_m. So, as α increases the cost of having a lot of nodes will increase.

Trees are quite useful and simple for interpretation, however relatively inaccurate and non-robust. To overcome this drawback several techniques are used: bagging, random forests and boosting. Essentially all of them use multiple trees and combine them to make a single prediction.

Bagging (Breiman 1996): taking repeated random samples from training set (bootstrap) and building regression tree models for all of them without pruning, then averaging models results to reduce variance. Number of trees (B) is not critical and not lead to overfitting, however predictions are highly correlated that does not lead to substantial reduction of variance comparing with single tree.

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

where B is number of trees and $\hat{f}^{*b}(x)$ separate tree model.

Random forests (Barandiaran 1998): similar to bagging, however random subsets of features (usually sqrt of features) are used for each tree, which help to decorrelate the trees.
Boosting: combines multiple decision trees. It doesn't involve bootstrap. Given a model, decision tree is fitted to its residuals and is added to a decision function to update residuals. Each of this tree could be quite small, with just a few terminal nodes. Number of nodes is determined by a parameter d. Learning rate parameter α is used to shrink contribution of each tree. Can overfit data if the number of trees is too large comparing to previous methods, however still quite robust to outliers.

Algorithm: Boosting
Data:
independent variables, *X*
dependent variables, *Y*
Return:

$$\hat{f}(x) = \sum_{b=1}^{B} \alpha \hat{f}^b(x)$$

begin
$\hat{f}(x) = 0$; $r_i = y_i$
For b=1 **to** B **do**
Fit a tree $\hat{f}^b$ fit a tree with d splits (d+1 terminal nodes) to the training data (X)
Update $\hat{f}$ by adding shrunken version of a new tree

$$\hat{f}(x) \leftarrow \hat{f}(x) + \alpha \hat{f}^b(x)$$

Update residuals

$$r_i \leftarrow r_i - \alpha \hat{f}^b(x_i)$$

end

Some articles (Ogutu, Piepho et al. 2011, Ghimire, Rogan et al. 2012) indicate higher efficiency of boosting comparing to random forests although mentioning comparable accuracy. However boosting is less sensitive to data size and noise (Ghimire, Rogan et al. 2012). Particular model considered in this work is gradient boosting regression (Friedman 2001).

Model building procedure:

1. Select model (Gradient boosting regression).
2. Run model with cross-validation procedure that selects best number of trees B.

Model diagnostics

Several problems may occur when fitting regression to the data. Among most common are non-linearity, heteroscedasticity, outliers and collinearity.

Non-linearity: linear regression models assume linear dependence between independent and dependent variables, however this is not always the case. Residual plots could be useful to check for non-linearity. As a simple solution to this problem, several transformations could be applied (log, sqrt, polynomial).

Non-constant Variance of Error Terms (heteroscedasticity): linear regression models assume constant variance of error. However, in many cases variance is heteroscedastic i.e. variance of error term changes with a change of dependent variables. That may lead to bias in standard error coefficients and therefore unreliable hypothesis tests or t-statistics. OLS estimators may also be less efficient.

Bruesch-Pagan statistical test may be used to test the data on heteroscedasticity. It is one of the most common tests. It assumes that error term is a linear function of independent variables.

$$\varepsilon_i^2 = \alpha_0 + \alpha_1 X_{i1} + \ldots + \alpha_n X_{in} + u_i$$

However, in practice error estimate is used instead, $\hat{\varepsilon}_i^2$. Alternatively, it can be performed with predicted dependent variables $\hat{Y}$ instead of independent X.

$$\hat{\varepsilon}_i^2 = \delta_0 + \delta_1 \hat{Y}_i$$

Algorithm: Breusch-Pagan test

- Estimate linear regression model
- Use predicted values $\hat{Y}$ to estimate $\hat{\varepsilon}_i^2$
- Calculate F-statistic and chi-squared statistic

$$F = \frac{\frac{R^2_{\hat{\varepsilon}^2}}{1}}{\frac{1 - R^2_{\hat{\varepsilon}^2}}{n-2}}$$

$$\chi^2 = nR^2_{\hat{\varepsilon}^2}$$

In general, if any of this test statistics is significant, it could be considered as evidence of heteroscedasticity. As mentioned by (Greene 2002) this test exaggerates the significance of results in small or moderately large samples. In this case the F-statistic is preferable.

As a solution to heteroscedasticity problem, it is possible to transform dependent variable using functions like sqrt or Box-Cox.

$$y = \begin{cases} \frac{(y^\lambda - 1)}{\lambda}, for\ \lambda > 0 \\ log(y), for\ \lambda = 0 \end{cases}$$

It is necessary to note that for Box-Cox transformation all variables have to be strictly positive. It is possible to achieve this by introducing a shift i.e. add 1 to all values in case zeros are present.

Outliers: point for which true value is far more than predicted value, may be caused by different reasons including errors in data. Normalized residual plots could also be useful here.
Observations with normalized residuals greater than 3 could be considered as outliers.

Collinearity: when two or more independent variables are closely related.
However, this is rarely a case in practice. This might influence regression coefficients and/or produce larger standard errors. It could be difficult to separate effects of collinear variables on dependent variable.
Collinearity could be addressed by variable elimination or combination of collinear variables. Some point out that ridge regression is another way to deal with this issue (Grewal, Cote et al. 2004).
However in several articles researchers notice that multicollinearity may lead to desirable outcomes and it is also reasonable to consider highly correlated variables (Friedman and Wall 2005). In other article authors state that high R^2 score and large sample size may mitigate multicollinearity problems like inaccurate standard errors and coefficient estimates. So, in general multicollinearity "should be viewed in conjunction with other factors known to affect estimation accuracy" (Mason and Perreault Jr 1991).

It is also not a big problem for combinations of decision trees (see descriptions of them in Model selection).

Multicollinearity may be measured with Variance Inflation Factor (VIF).

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}},$$

where $R^2_{X_j|X_{-j}}$ is the R^2 from a regression X_j onto other variables.

As a rule of thumb VIF equal or higher than 10 should be considered as high degree of multicollinearity (Hair), although same article mentions that it is researchers duty to determine acceptable level.

It was decided to use 10 threshold for VIF and 0.7 for Pearson correlation as mentioned in (Zhao, Deng et al. 2014).

Model assessment

In order to evaluate a model, it is necessary to quantify how close is predicted value for a given observation to the true value for that observation. Certain methods are used for regression models. Most common among them are calculation of mean squared error (MSE) and R^2 [6].

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1}(y_i - \bar{y})^2}$$

Where $\hat{y}_i$ is predicted value, y_i is true value, $\bar{y}$ is the mean of observations and n is number of samples.

However, they are not suitable for selecting models with different number of variables as they tend to improve with each additional variable.

[6] R^2– coefficient of determination, provides a measure of how well future samples are likely to be predicted by the model

So, to estimate possible prediction errors there are two options either using adjustment methods ($AIC, BIC, adjusted\ R^2$) that account for number of variables used in model apart from errors or direct estimation (validation or cross-validation).

Indirect (adjustment) model metrics (where n is the number of observations, and k – number of parameters used):

Adjusted R^2 (R^2_{adj}) is quite common in comparing similar models with different number of parameters:

$$R^2_{adj}(y, \hat{y}) = 1 - \frac{n - 1}{n - k - 1}(1 - R^2)$$

Another popular measure for comparing models is **Akaike Information Criterion (AIC)**

$$AIC = n \cdot ln(MSE) + 2k$$

AIC tends to improve with larger number of parameters, therefore it could be prone to overfitting. To mitigate this problem, it is possible to use **Bayesian Information Criterion (BIC)**

$$BIC = n \cdot ln(MSE) + k \cdot ln(n)$$

BIC tends to select simpler models if reasonable number of samples is available i.e. more than 100.

Direct (estimation) model metrics:

As the name suggest this type of metrics involve direct computation. One of the main benefits of such approach is that it uses available data efficiently. It could be quite helpful in preventing overfitting, however may as well be relatively computationally expensive in some cases, and as a result it was not very popular until recently.

With **validation**, it is necessary to divide data onto train and test sets. Train set is used to fit the model. And test set is used to check for results reproducibility. Then process continues with new split. The problem of this approach is that results may vary greatly depending on sampling and it doesn't suit well for small datasets.

Cross-validation on the other hand overcomes validation drawbacks. It will be used in this work, as this technique is more essential for comparing models from different domains, as for example it is necessary to estimate number of variables in multiple decision trees for adjustment metric indirectly. The cross-validation approach considered in this work is k-Fold cross-validation.

Algorithm: k-fold cross-validation
Data:
independent variables, X
dependent variables, Y
Result: the average of all folds evaluation results
begin
Divide dataset into k folds i.e. subsets
For test_fold **in** Folds **do**
    Train model using $Folds$ minus $test\_fold$
    Evaluate results with $test\_fold$ that was not used in computation
end

Quite common is to use from 5 to 10 folds.

4.1.4. Evaluation

Analyze modelling results and select model with the help of R^2 metric from cross-validation and validation procedures. Select one that best fits the data.

4.2. Venue popularity measuring

After selection of a model for popular times it is possible to proceed with the second part of a research i.e. definition of number of venue visitors. The final goal of this activity is to approximate number of visitors for a given venue without need in additional sources like Google Popular Times and special hardware installed. However here, author presents only measuring setup and evaluates its performance against Google Popular Times data.

Defining number of venue visitors

As it was mentioned previously it is quite common to use technologies like WIFI in presence detection systems. Moreover, current progress in microelectronics enables us to build quite efficient and cost-effective solutions.

How are WIFI devices tracked?

Each device has unique MAC[7] identifier that is broadcasted with probe requests[8] within a certain operational range and certain time periods (device dependent). So, listening for such requests helps to identify unique devices in the area, as well as duration of their stay.
Some companies randomize MAC addresses in certain cases, that brings up additional noise to the measurements. However according to (Martin, Mayberry et al. 2017) randomization adoption is extremely low, especially in Android devices. So it is still possible to use passive sensing technology in presence detection systems as mentioned in article (Nunes, Ribeiro et al. 2017).

[7] MAC – media access control, is a part of data link layer (layer 2) of Open Systems Interconnection (OSI) model of computer networking that describes data transfer between system nodes (for details refer to ISO/IEC 7498-1 standard)

[8] Probe request – special frame (information block) that is sent by a client (mobile) station to discover networks in proximity. It requests information about access points parameters and, normally, all access points in the area respond to it (for details refer to IEEE 802.11 standard).

4.2.1. Data collection and cleaning

System setup

There are many solutions for WIFI detection either fully commercial systems or opened like Raspberry PI. The latter is quite small and cheap with a price starting at 10 Euros for Zero W model, as well as relatively energy efficient. Therefore, it was decided to use it as potentially scalable and highly portable solution, together with an external power bank (battery) as a primary energy source.

As this board doesn't have internal storage it is necessary to equip it with SD memory card. Linux operating system with Python support is installed onto memory card in order to facilitate fast prototyping. It is necessary to note, that writing code in C language could potentially increase battery operation time of the board.

WIFI driver was patched with the help of Nexmon framework (Schulz 2017) to enable monitor mode and thus check for network packets like probe requests in current study.

Python[9] script (see flow-chart below) was written to interact with WIFI card via sockets[10] object. It receives raw packets and checks if their radiotap headers[11] contain probe request frames. If yes, the data is added into a temporary object with MAC address being transformed with hash function for privacy reasons. Finally, if time period exceeds certain threshold, temporary object is written into file.

[9] Python – an interpreted high-level programming language for general-purpose programming (for details refer to https://www.python.org/)
[10] Sockets – programming interface for inter-process communication (IPC)
[11] Radiotap header – a mechanism to supply additional information about frames, from the WIFI card driver to user space applications and from a user space application to the driver for transmission. Designed initially for NetBSD systems by David Young (for details refer to http://www.radiotap.org/).

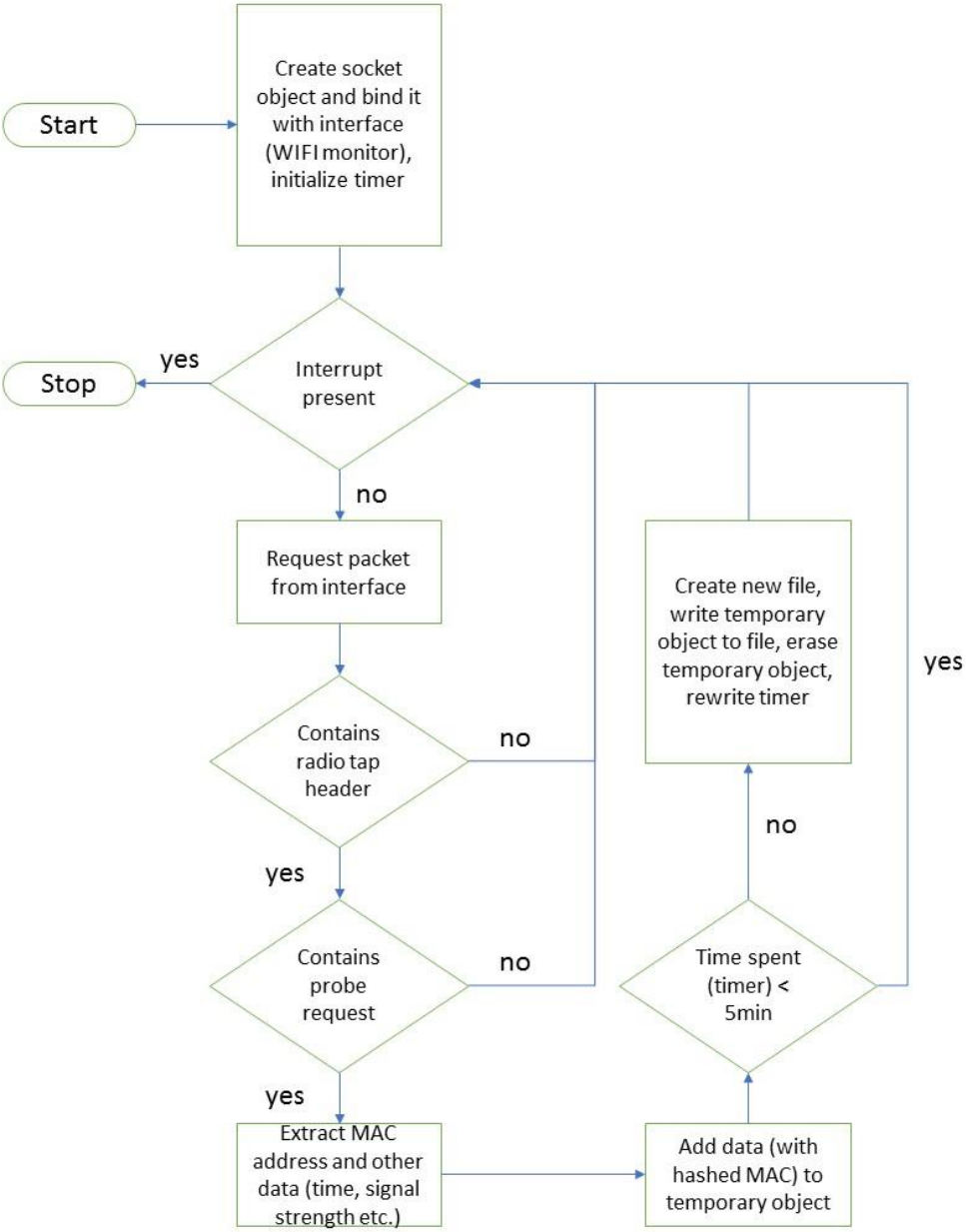

Figure 10: WIFI data acquisition (flow-chart).

As mentioned in (Nunes, Ribeiro et al. 2017) it is enough to monitor only one recommended non-overlapping channel[12] as it won't affect capture, due to the fact that device cycles through all channels when sending requests. Therefore, same 11th channel was used in this study.

[12] WIFI channel is a frequency used in WIFI network. Main non-overlapping channels are 1, 6 and 11 with central frequencies 2412, 2437 and 2462 MHz for 802.11 b/g networks.

Test setup

Assembled device is installed onto mobile platform that should be parked near venue of interest. Due to limitation of battery and offline nature of the device setup it is necessary to reinstall test setups at least once in two days.

4.2.2. Data exploration

As a first step files with data are parsed by Python script and each measurement is assigned to a period according to desired frequency (5 minutes by default), based on stored packet time.

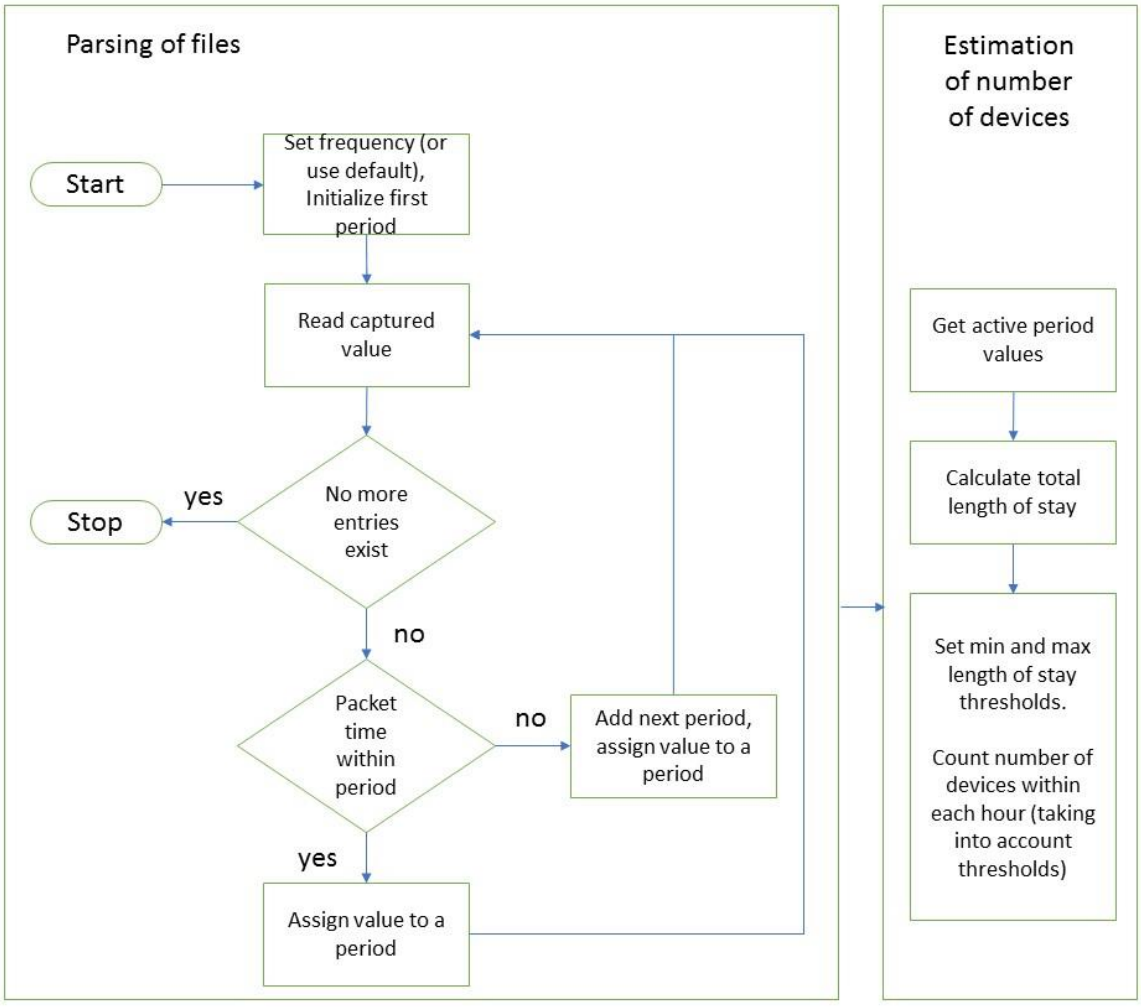

Figure 11: Defining number of devices available at each hour (flow-chart).

Then for all unique hashed MAC addresses, active periods (i.e. periods with captured frames) are identified. Difference of minimum and maximum values of these periods correspond to an approximate length of stay of a person (see example histogram below).

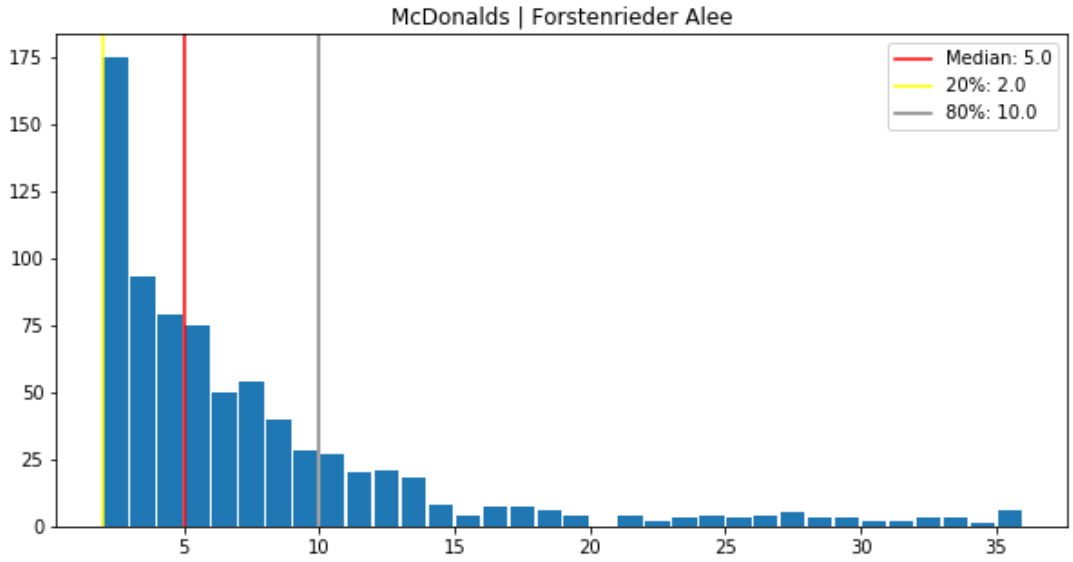

Figure 12: Example of length of stay in a particular venue.

Minimum and maximum limits on length of stay should be introduced to deal with passersby and workers/residents respectively. It could also be possible to calibrate values within these limits against Google Popular Times values. Although, this means that we believe that Google data is correct and has minimum bias.

As real occupancy numbers are unavailable, additional research is necessary to estimate share of detected devices in total venue visits.

5. Case study

5.1. Research area

City of Munich was selected for the case study. It is the capital and the most populated city in the German state of Bavaria, with a population of around 1.5 million[13], and third largest city in Germany.

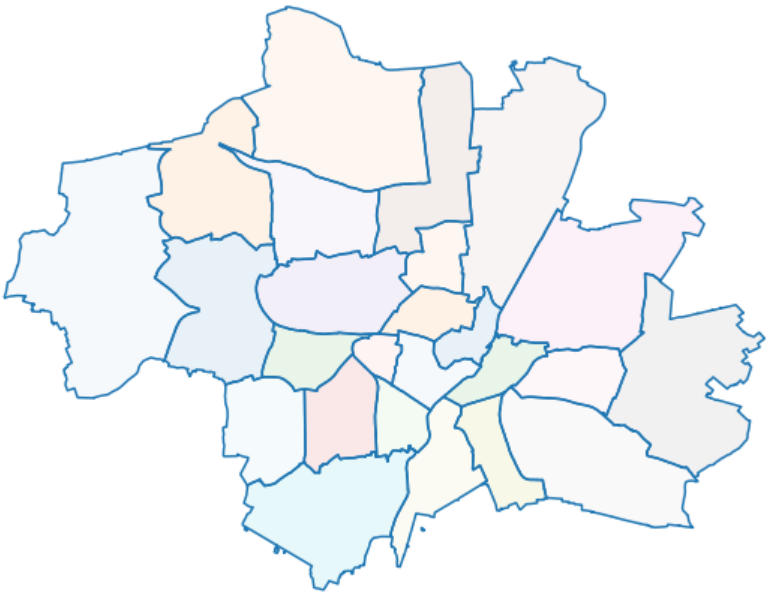

Figure 13: Munich districts.

5.2. Data sources

Table 1: List of primary data sources.

Yelp	https://www.yelp.com
Google Maps	https://www.google.com/maps
Google location API	https://developers.google.com/maps/documentation/geolocation/intro
Overpass API	https://wiki.openstreetmap.org/wiki/Overpass_API
OSM dump	https://www.geofabrik.de (pbf file)
Population	https://www.zensus2011.de (German Nationwide census, 2011)
Workplaces	https://www.muenchen.de (Munich, 2016)

[13] https://www.muenchen.de/rathaus/Stadtinfos/Statistik/Bev-lkerung.html

5.3. Software used in the project

All work for this project was done with the help of Python programming language and several Python libraries listed below, as well as others included into Anaconda[14] distribution. The only exception was the clustering metric (S-DBW), that was available in R's clv package.

Table 2: List of Python libraries.

Selenium	Emulation of user activity in browser
Beautiful Soup	Parsing of HTML and XML documents
Pandas	High performance and easy to use data structures and data analysis tools
Geopandas	Extension of pandas library for work with spatial data
Osmread	Reading of OpenStreetMap XML and PBF data files
Osmnx	Retrieving, constructing, analyzing and visualizing street networks
Scikit-learn	Tools for data mining and data analysis
Tslearn	Tools for data mining and data analysis of time series
Matplotlib	Data visualization
StatsModels	Estimation and evaluation of statistical models

[14] https://www.anaconda.com/download/

5.4. Venue popularity modelling

5.4.1. Data collection and cleaning

Yelp

As a first step of data collection it was necessary to collect basic data on available venues and Yelp web site is quite handy for this task. Here it is possible to extract venue name, price level, rating, number of reviews, venue tags and address. Extraction is made based on venue type (for example restaurants).

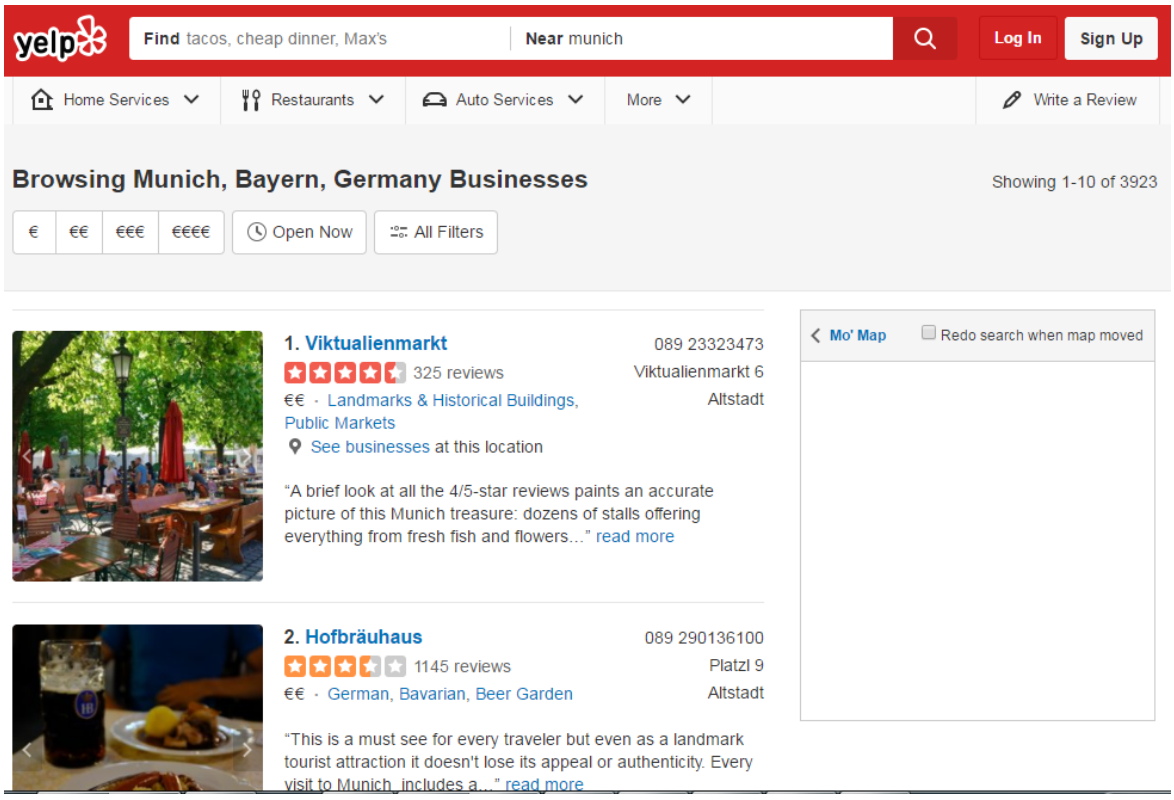

Figure 14: Screenshot of Yelp (yelp.com).

As an initial step, some manual research was made to get proper query form. Then Python script that interacts with a web page using Selenium library was started. It scanned through all venues within certain group, for example restaurants, parsed data with Beautiful Soup library and wrote all necessary data into file.

It is necessary to note that parsing process relied heavily on page markup and therefore any change to html tag names would require modification of parsing part of the script.

As another caveat web servers usually block maximum amount of venues returned to the client, however after examining all areas, number of missed places was negligible.

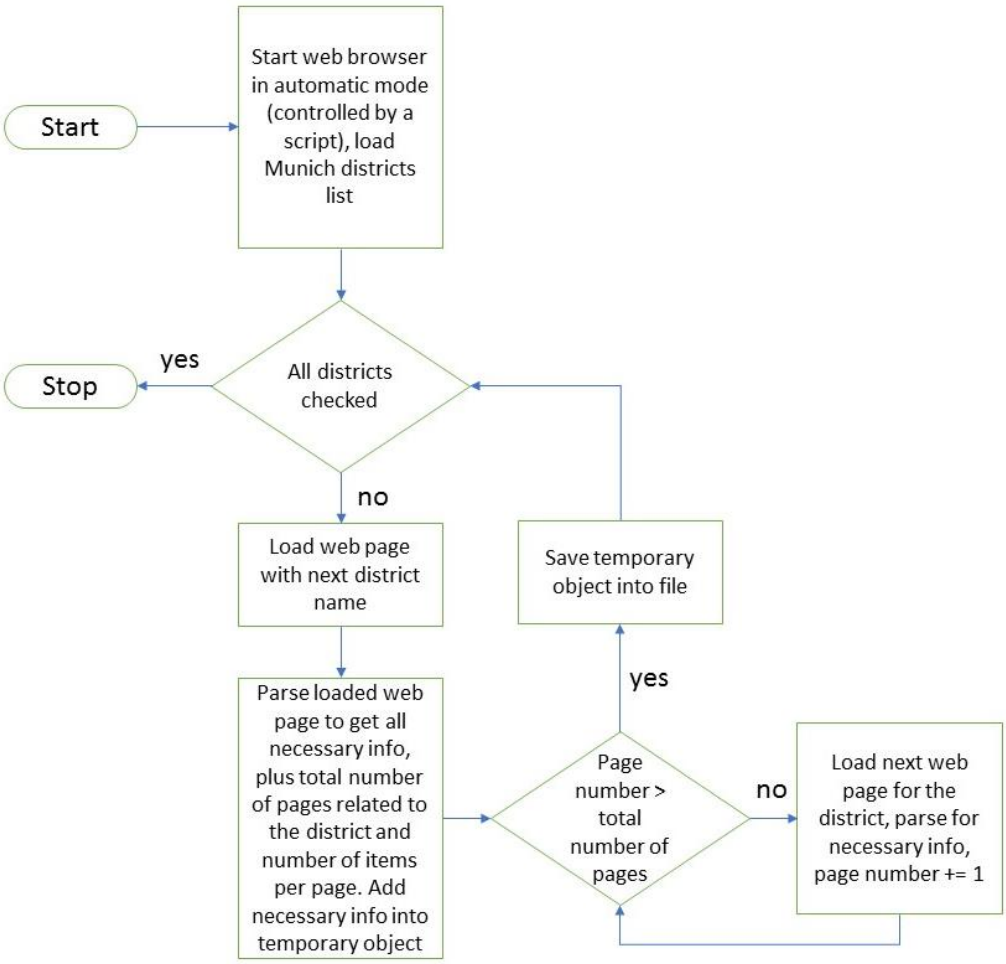

Figure 15: Yelp scraper (flow-chart).

Google Maps

Based on name and address from yelp.com additional information was extracted from Google Maps i.e. price level (available at few venues), rating, number of reviews, “Popular Times” and opening hours.

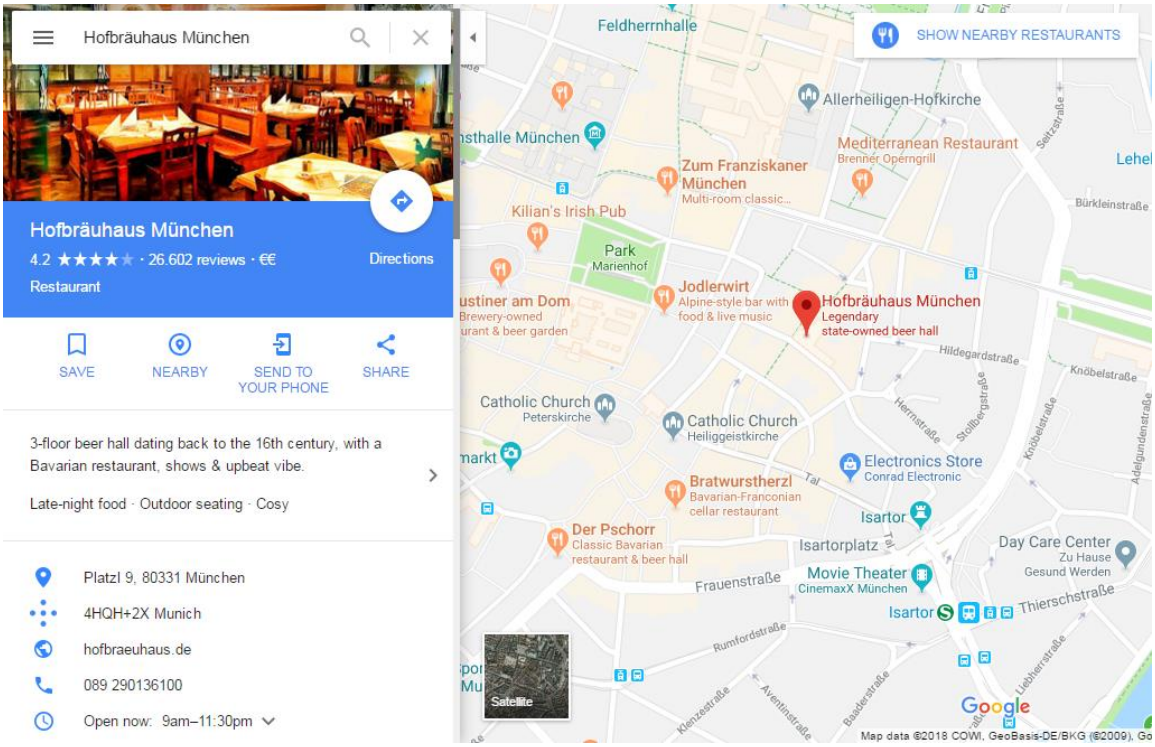

Figure 16: Screenshot of Google Maps (google.com/maps).

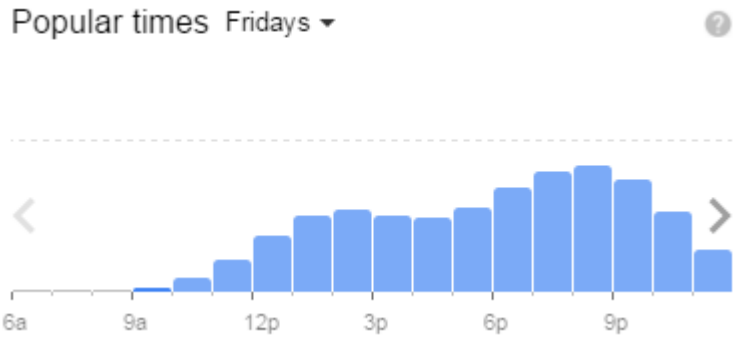

Figure 17: Screenshot of Google Popular Times section (google.com/maps).

Some preliminary work was done to check proper request form for Google script as well. Then slightly modified script, using same libraries as in Yelp case, was written to get data from Google Maps. It used iterator object to get last venue and saved it into file in order to get value in case of script crash.

Main problem of interacting with Google with this script was the fact that after certain number of requests, server stopped responding. Another challenge was the delay in loading of venue page i.e. it may take approximately 10 seconds to get the information.

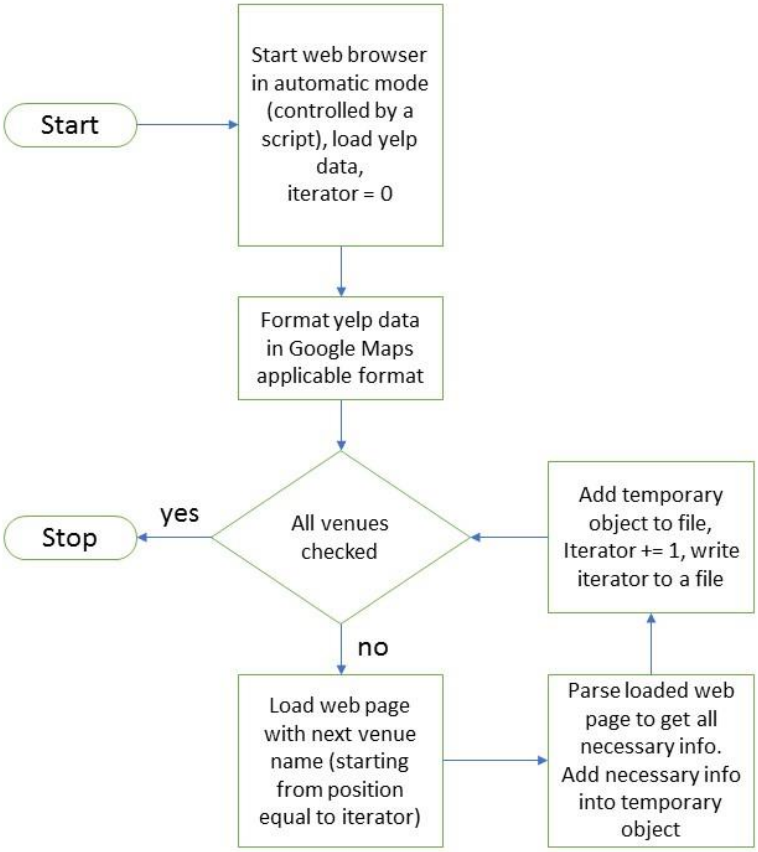

Figure 18: Google scraper (flow-chart).

Google location API

Geocoding (getting latitude and longitude) of venues based on name and address (this is also possible with OSM, however sometimes results are relatively inaccurate) was necessary for the next project step (referencing of objects).
Geocoding is done with a simple request to Google server. It is also necessary to get free developer API key to make these requests.
The major drawback of Google location API is limitation of requests for a free account.

Referencing of objects

OSM (pbf file with OSM objects from www.geofabrik.de)
German Nationwide census 2011 (100 x 100 m grid with population data within designated area from https://www.zensus2011.de)
Die sozialversicherungspflichtig Beschäftigten in München nach dem Wohnort im Dezember 2016 (workplaces in Munich per administrative region from https://www.muenchen.de)

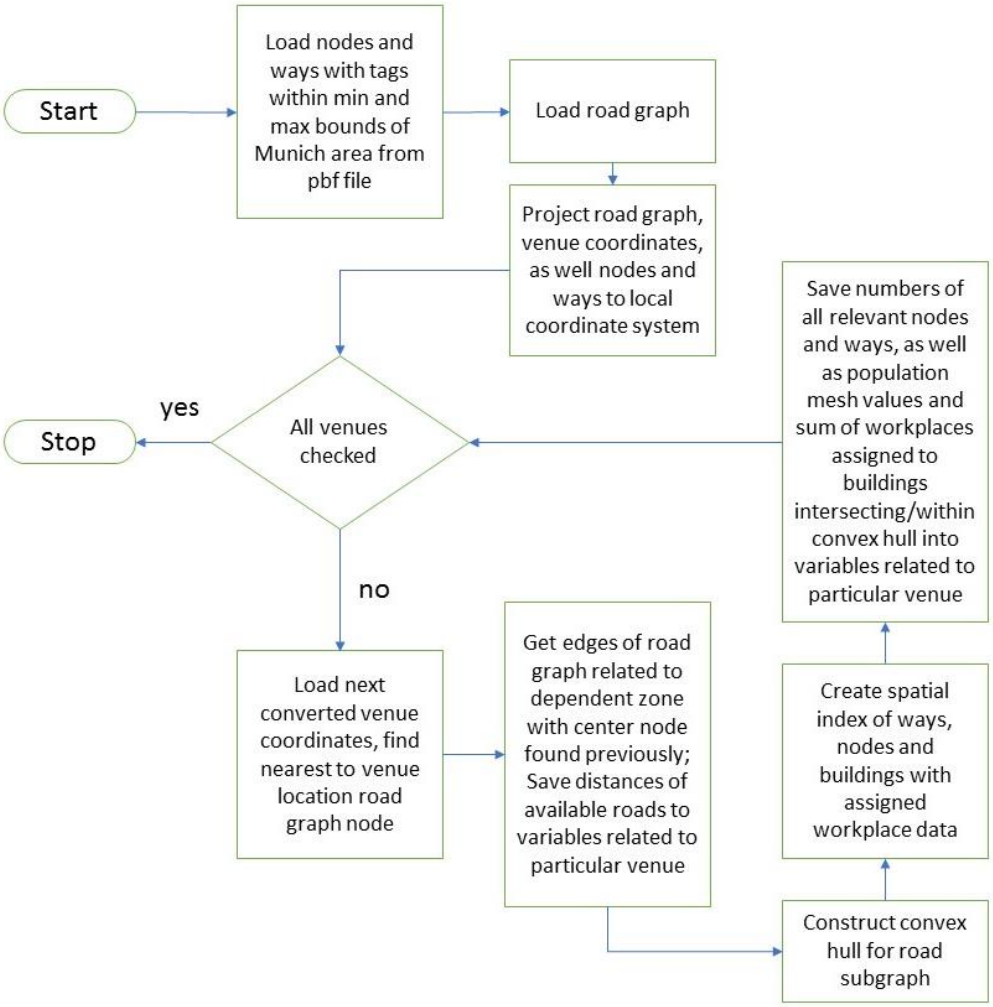

Figure 19: Assignment of variables related to venue dependent area (flow-chart).

To assign spatial information to the venues collected in previous steps, several procedures were implemented.

OSM data was loaded with osmread library to get only ways and nodes related to Munich area. Then road graph was loaded with osmnx library (via OSM API) and projected on a map. Venue coordinates, loaded in previous step, were used to get central points. Then road graph was used to get all roads within venue surrounding area. Two distances of influence were tested: 400 and 800 m, with former being quite common among literature, see review in (Rodas 2017). Road endpoints were used to construct convex hulls. All nodes, ways, population grid cells and sum of workplace values from buildings (see disaggregation algorithm to get number of workplaces per building below) intersecting/within convex hulls were added to appropriate variables. In order to speed up these calculations, built in spatial index function sindex of geopandas library was used.

Disaggregation algorithm

As population data was available in the form of grid cells with adequate spatial resolution no disaggregation steps were necessary. However, workplace data was relatively aggregate, therefore disaggregation algorithm was used to distribute workplaces among Munich administrative areas.

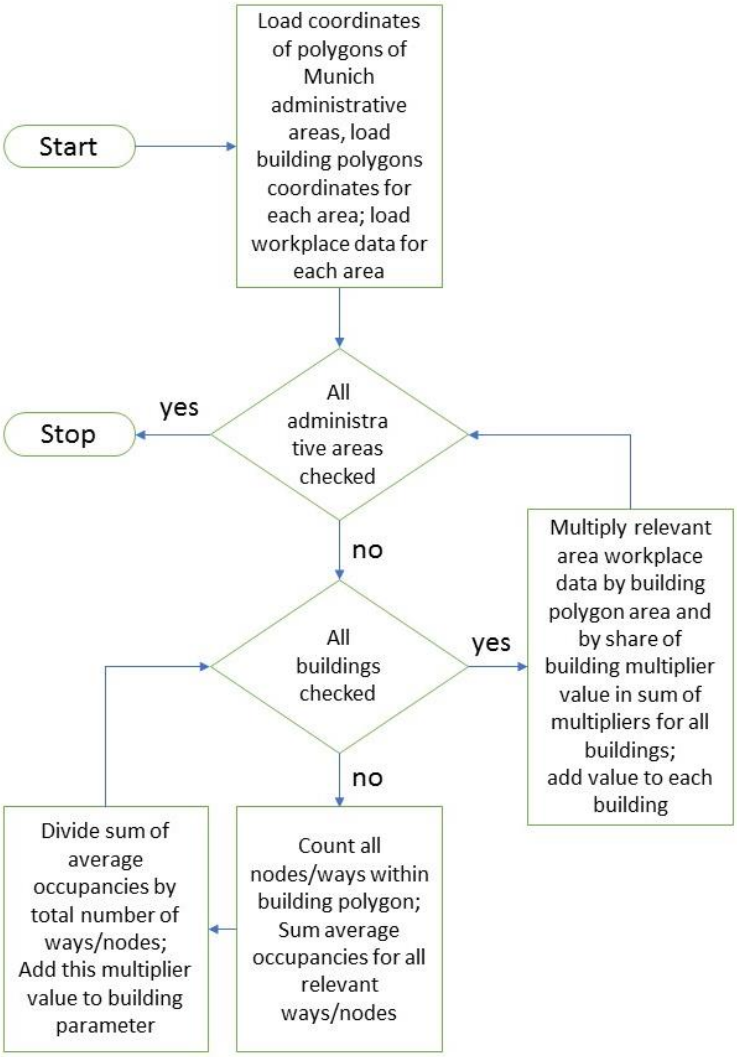

Figure 20: Disaggregation algorithm (flow-chart).

Coordinates of Munich administrative regions with building polygons were loaded using osmnx library. Then all nodes and ways within building polygons were counted.
Based on their average occupancy type (values for all nodes/ways are summed up and then divided by total number of nodes ways) multiplier was added to a building.

Only two different occupancy types were defined based on norms of workplaces per square meter in office (US[15]) and in educational institution (Europe[16]), as not so much data is available on this matter. Therefore, data was fitted into 2 groups: related to offices/retail/food, public buildings (non-educational) and universities, healthcare and educational organizations. For buildings without such nodes and tags minimum value was added.

Finally, data on workplaces within each administrative area is divided onto all buildings based on their area in square meters and multiplier.

It is necessary to note that this approach is relatively inaccurate, as for ex. doesn't account for building's number of floors and space occupied by a company (mostly because of lack of such data) and uses general data on job density per sqm.

Data structure

Variables were defined based on tag names from Yelp, classes and class groups of OSM, venue type. Some variables like rating and reviews were combined, others like latitude and longitude were converted to proper projections. For working hours, only current hour and two hours in each direction were used in order to limit collinearity. All Google Popular Times values were assigned to dependent variables.

Table 3: Variables description.

Variable name	Description
-	Index
Name	Name of venue
lat_conv	Latitude
lon_conv	Longitude
Price_index	Price level from Yelp
compound_rating	Weighted sum of ratings obtained from Yelp and Google Maps
total_reviews	Sum of reviews at Yelp and Google Maps
*	Type of amenity (for ex cafe_fastfood)
*	Tags attached (for ex. Caribbean)

[15] https://www.eia.gov/consumption/commercial/
[16] http://www.oecd.org/education/school/48483436.pdf

roads_* nodes_* ways_*	OSM data on length of different classes of roads and number of venues within prespecified area
workplaces	Workplaces data within prespecified area
population	Population data within prespecified area
*	Working hours (-2 hours, -1 hour, current hour, +1 hour, +2 hours)
*	Venue popularity data 24 hour/7 days (for ex. ('sun', 1))

5.4.2. Exploratory data analysis

A lot of studies consider variable “distance to city center”, however the term city center is not explicitly defined (Kisilevich, Keim et al. 2013), whether it should be geographical center (within current city borders), historical center or center of activities.

Although this variable is not used explicitly in the study, it could be useful to investigate this question. It is possible to say that due to the fact that the geographical center or centroid (according to OSM city borders) is shifted to the west and is located near Inner Ring road, it could not represent real center. Therefore, better option is to use filtered out (i.e. only venues with popular times) data from Google.
From the distribution of coordinates of these venues, it is possible to assume that Munich is relatively monocentric city, slightly stretched in north-south direction.

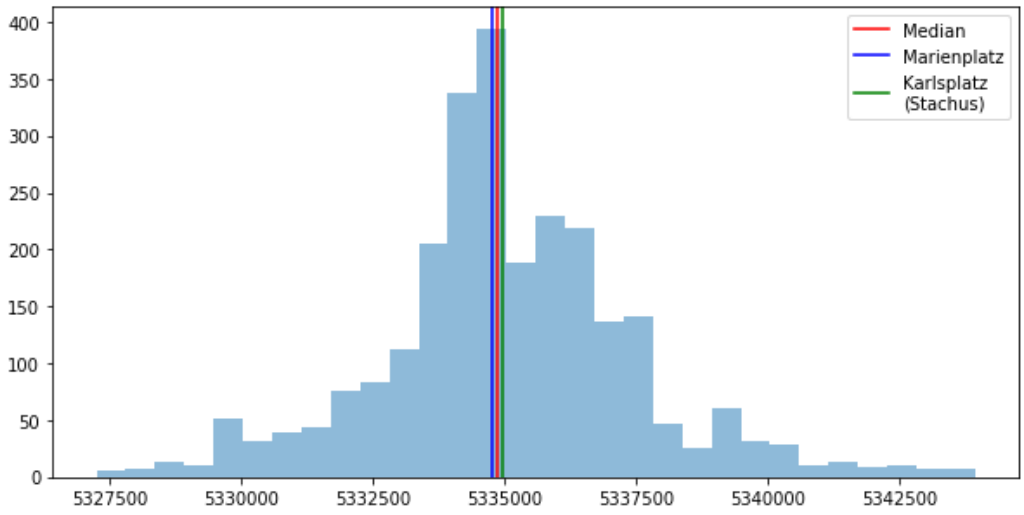

Figure 21: Latitude distribution

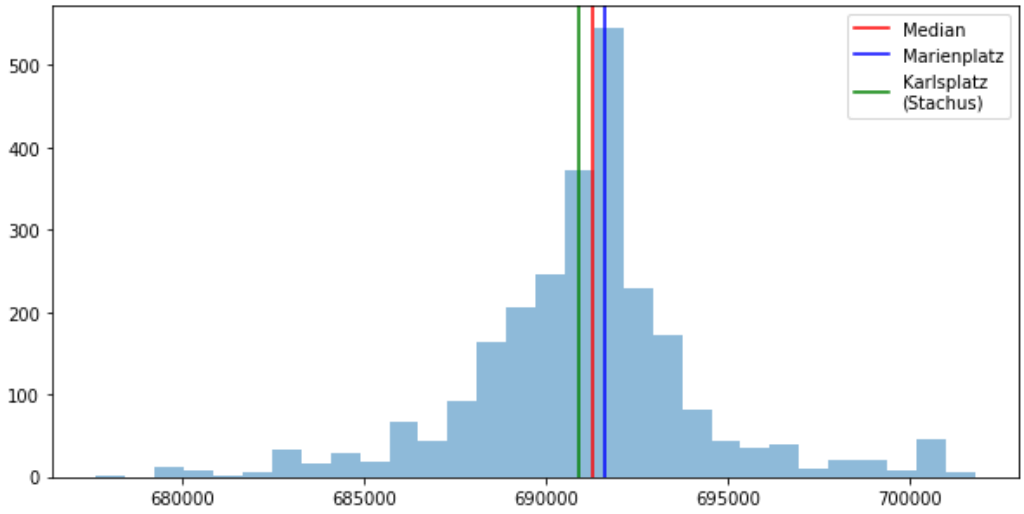

Figure 22: Longitude distribution

Moreover, based on this info, we may conclude that the central point of activities (with majority of them related to food) is located quite close to Marienplatz, Karlsplatz (Stachus) and Hauptbahnhof the region that is de facto city center and historical center as well.
Other common variables include total number of reviews from Google and Yelp, compound rating i.e. weighted sum of ratings from Google and Yelp, price index, population and workplaces.

Table 4: Common variables.

Name	Min	Max	Median	Mean	Standard Deviation
total_reviews	0.00	11423.00	111.00	174.20	275.05
compound_rating	0.00	5.00	4.24	4.19	0.42
Price_Index	0.00	4.00	2.00	1.57	1.02
population	0.00	11753.00	3923.00	4176.45	2814.16
workplaces	0.00	6495.00	1720.00	1904.15	1239.45

It is necessary to mention that share of items without Price_Index is close to 24% and more than 50% belong to Price_Index 2. Quite often it is misclassified, probably because of different price perception as well as low number of contributors (see example below).

Table 5: Comparison of some user defined price indexes.

	McDonald's	Burger King
Price_Index	Number of entries	Number of entries
0 (undefined)	2	1
1	8	7
2	11	3

Clustering

To understand available data structure, clustering procedures were implemented.
After calculating distances with DTW (global constraint – "Sakoe-Chiba", window size – 2) for "Popular Times" variables with tslearn library and Euclidean for others with scipi.spatial.distance matrix, data was visualized using t-SNE (learning rate – 200, number of iterations – 1000 and precomputed metric) method of sklearn library with certain varying parameters for sensitivity analysis i.e. perplexity (4 to 52 with step 2) and share of DTW (0 to 1 with step 0.1) in distance metric.

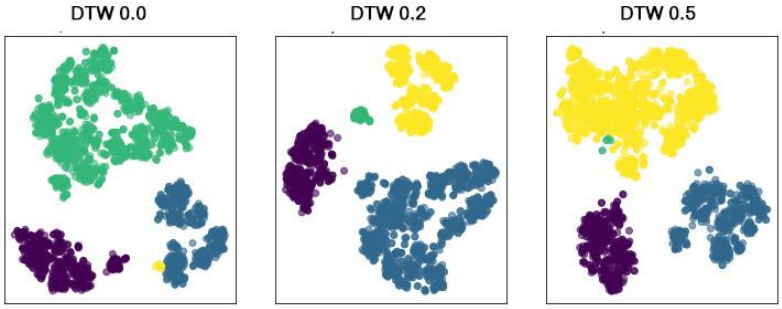

Figure 23: Example of t-SNE method visualization with 4 clusters.

From the analysis of visualization, it is evident, that, up to certain point, clustering results are dominated by location parameters like length of roads, facilities near the venue etc.

Moreover, as it was mentioned in methodology it could be really hard to define optimal clustering with internal validation measures.

To consider general metric quality it was decided to check some examples in detail. Let it be an option with DTW only distance and perplexity equal to 36. Best metrics from (Hassani and Seidl 2017) i.e. Calinski and Harabaz (CH) (Caliński and Harabasz 1974) and S-DBW (Halkidi and Vazirgiannis 2001) were used to evaluate clustering result. In the example below, both metrics improve up to 9 clusters, then CH metric slightly degrades, while S-DBW slightly improves. From 10 clusters both metrics proceed with improvement again, but from 12th CH start degrading. Similar ambiguous results were achieved with other example without DTW distance and perplexity 46. Moreover, CH kept improving after 13 clusters, however S-DBW was producing indefinite results after 10 clusters.

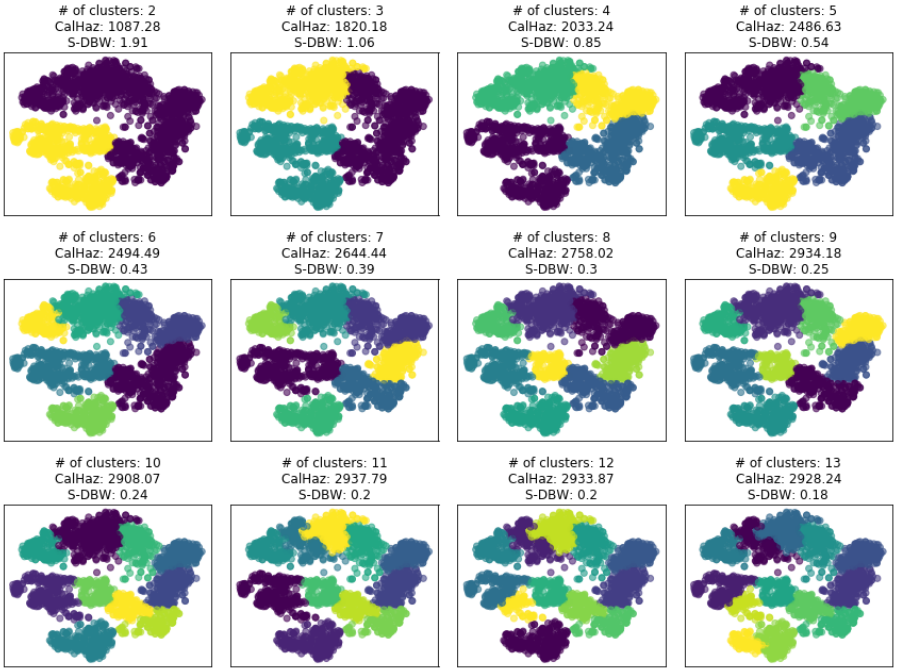

Figure 24: Clustering example (DTW 100%, Perplexity 36).

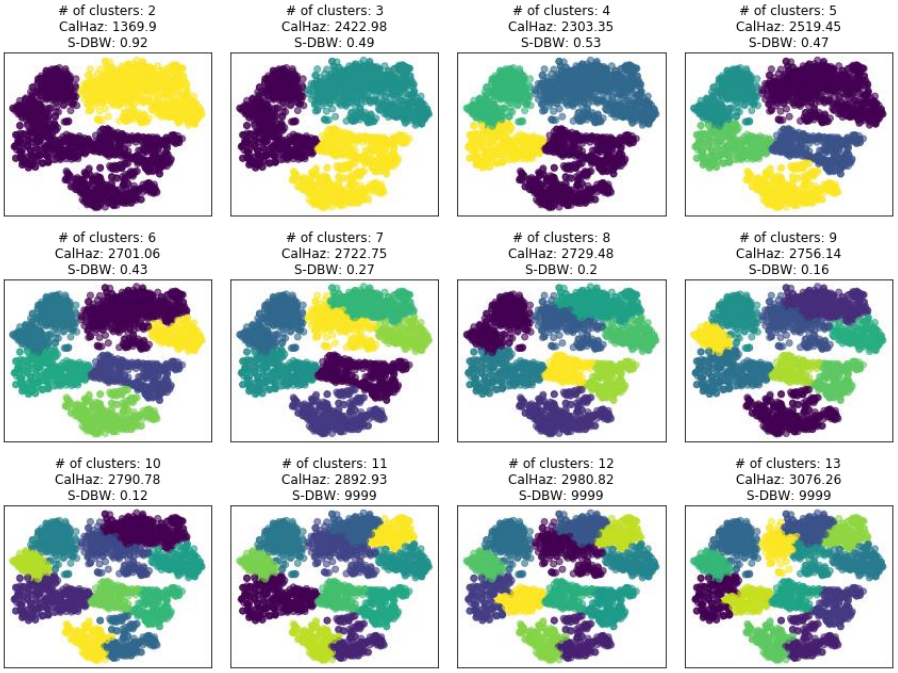

Figure 25: Clustering example (DTW 0%, Perplexity 46).

Given the above, it is quite evident that metrics results are not very reliable in this case. Therefore, it was decided to check t-SNE results with different perplexities and decide about clustering visually.

It was also found out that “complete” linkage performs badly with poorly defined clusters. Therefore “ward” linkage was used instead.

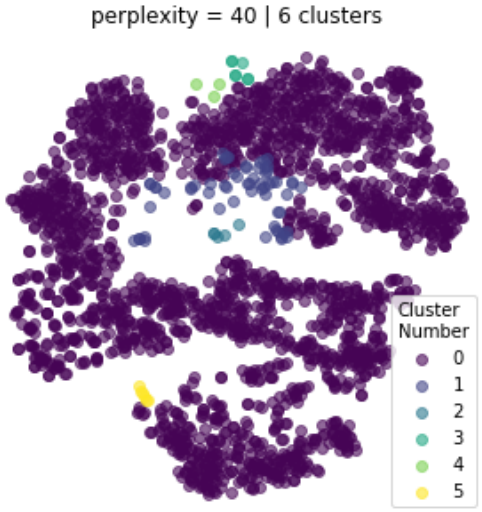

Figure 26: Clustering partition with "complete" linkage.

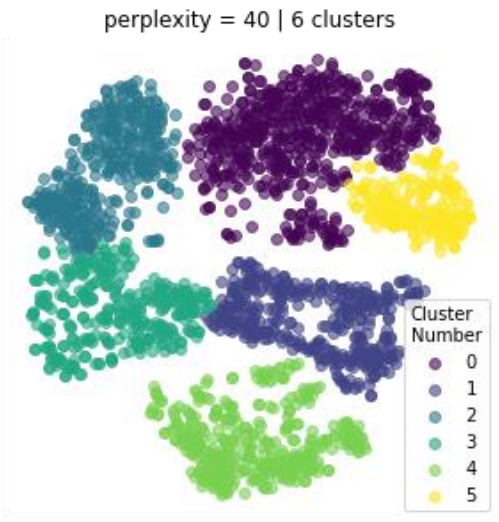

Figure 27: Chosen clustering partition with "ward" linkage.

Clusters description

After defining clusters, it is necessary to describe some distinct features of each.
Clusters 0 and 1 are quite similar to each other in geographical terms. Both have quite good transport connections. Cluster 0 has slightly lover population density comparing to 1. Population density in both groups is relatively low, but number of workplaces is quite high, and once again slightly higher in 1. Cluster 0 also has significant number of footways and possibly slightly more places for rest comparing to 1. Total number of reviews for cluster 0 is the highest among all clusters, however rating is the lowest (that is quite usual for famous and crowded places); for cluster 1 situation is opposite. Clusters 0 and 1 have more fast-food venues comparing to other clusters with majority belonging to 0.
Cluster 2 is generally well populated and has a lot of workplaces. It has slightly lower quality of transport connections, as well as shops, services and footways comparing to clusters 0 and 1. On the other hand this group has plenty of residential roads.
Cluster 3 could be associated with nightlife, as its popular time pattern suggests. Moreover, venues belonging to this group are generally located in central places with high population and workplace density, as well as high subway availability. There are also a lot of trees around. Here one may find a lot of shops, services. Quite significant part of this cluster belongs to bars.
Cluster 4 may be considered as a suburban location with lowest population and workplace density of 6 clusters. There is practically no hotels and other temporary stay places. This group also has minimum amount of organized bicycle parking, facilities for example restaurants and services.

Cluster 5 may be considered as relatively old residential area with relatively low number of workplaces, but significant population, though lower than in the city center. It also has limited access to subway and tramway. However, number of bus stops is generally similar to other quarters (except cluster 4). Number of stores and services is moderate, but in some cases slightly lower than in other clusters (except 4).

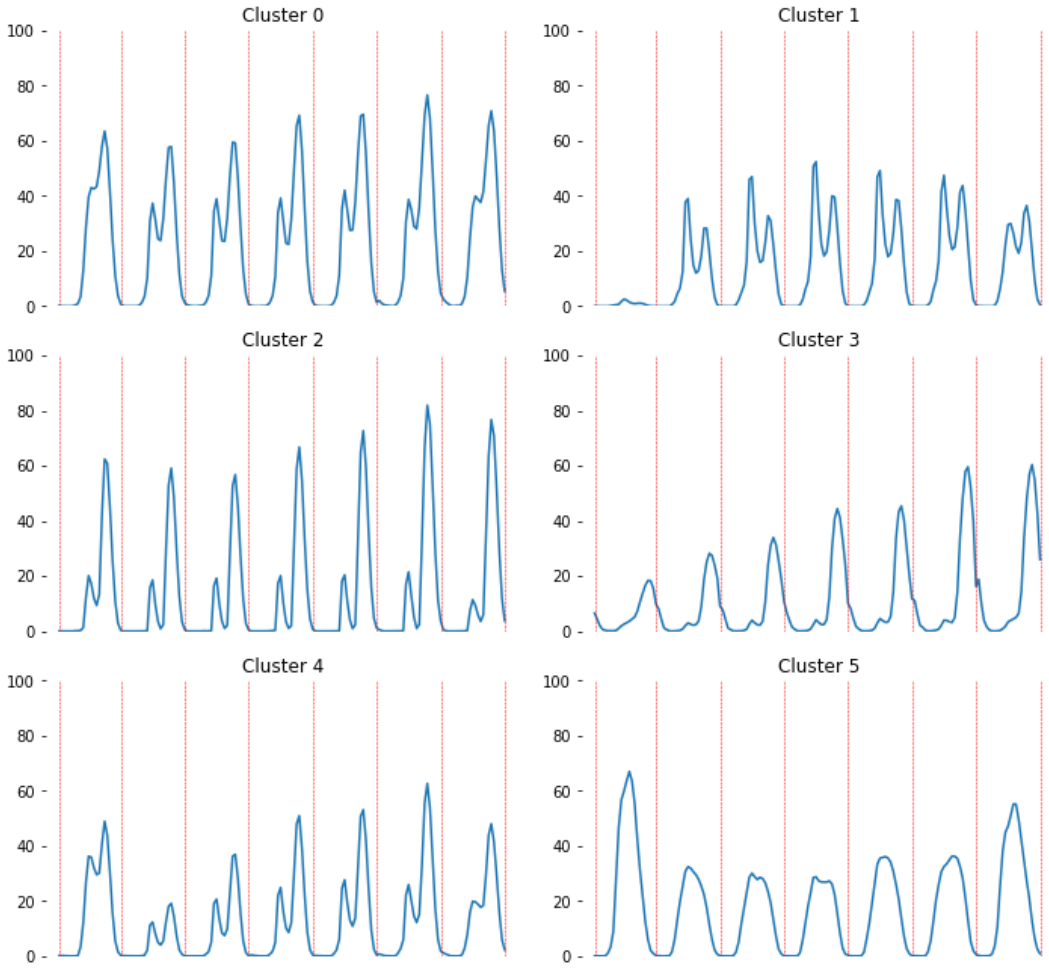

Figure 28: Mean popularity values per cluster.

5.4.3. Modelling

Linear regression with lasso regularization

Cross-validation method with 10 folds was used for each output (for regularization parameter selection and separately for cross-validation). After running a linear regression model, residuals were tested for several problems.

From figures of predicted versus true values, as well as residuals vs predicted values it was quite evident that outliers are present. As for heteroscedasticity, it could be detected visually in some cases. On the other hand, it may look like a result of natural boundaries of the data (0 – 100) for some models. Breusch-Pagan test was used to confirm visual hypothesis of heteroscedasticity, and majority of values were significant meaning that heteroscedasticity may be present.
It is also hard to tell something definite about data non-linearity, for now it will be assumed that data is linear and later on, non-linear method results (gradient boosting) will be inspected and compared to linear regression.
Moreover, it is possible to notice that the data does not always follow 45(degree) line and tends to have higher slope.
Logarithmic (Box-Cox with λ=0) and Box-Cox transformations were used to deal with these problems as these are quite common in literature and perform quite well.
Afterwards, performance of models before and after transformation was evaluated.

Several outliers, with residuals values more than 3 were removed that gave improvement in performance on training set, however, practically didn't influence test set or even resulted in slightly lower score (if test set is used as is).
In some cases, outliers may be a result of the fact that some venues could have occupational patterns more similar to for example clubs, however classified as bars, therefore their predicted occupation was much lower than true one, especially after midnight (see figure below). Similar behavior may be observed in the morning. So, in general this may be a result of some missing predictor or simply available data limitation.

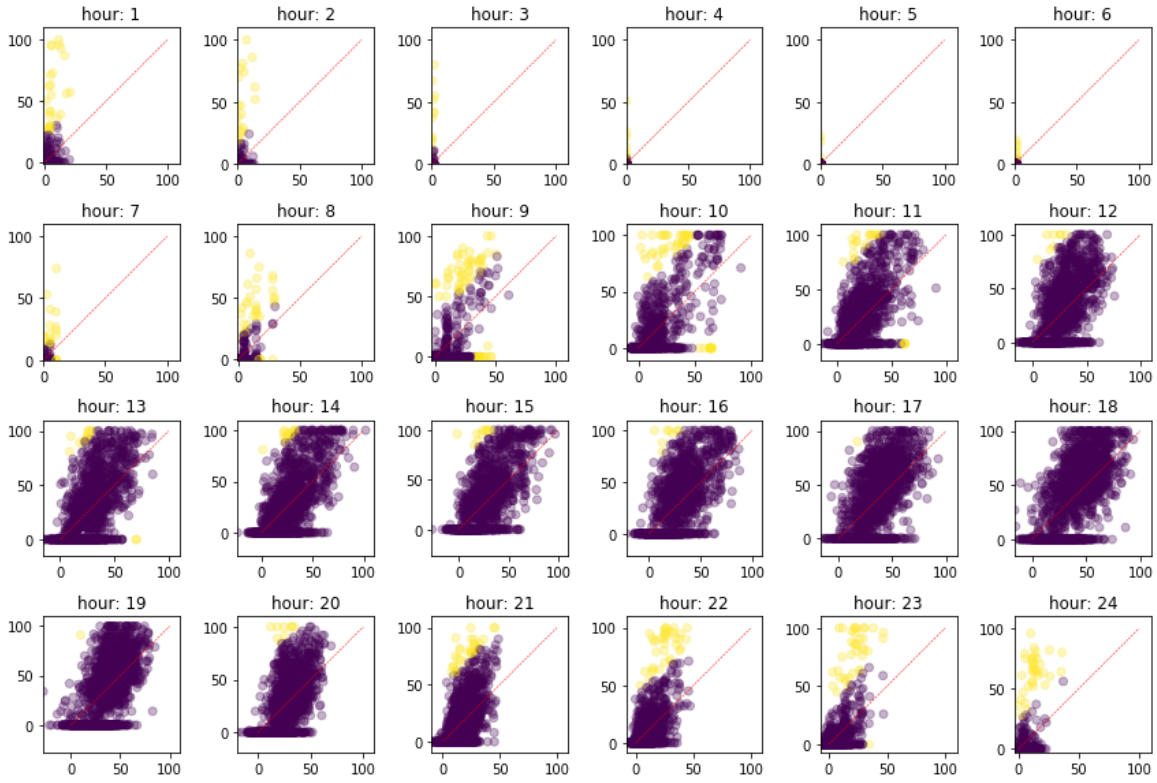

Figure 29: Predicted vs true values multiple linear regression example (outliers highlighted with yellow color). Horizontal axis – predicted y, vertical – true y.

It is also necessary to note when analyzing residual plots that even though absolute errors are not very high for some models, for example during early morning hours (approximately 3-6 am), prediction scores are quite low or even worse than constant as several occupations vary significantly from values close to zero.

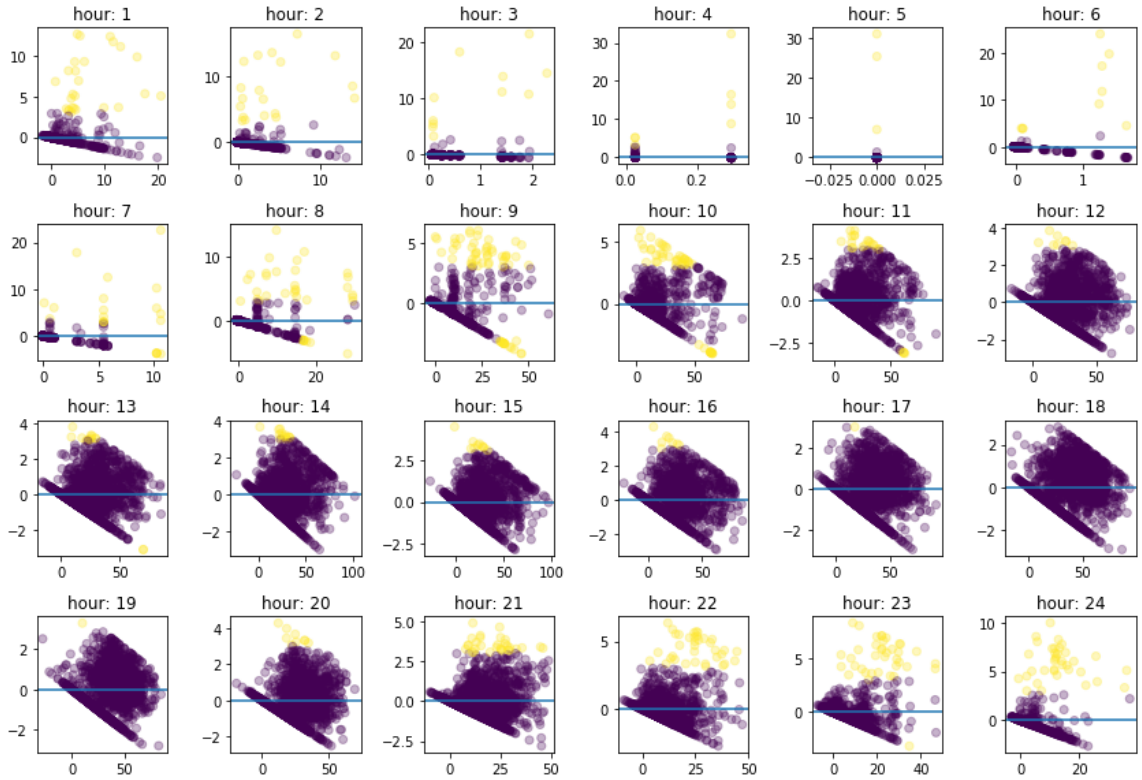

Figure 30: Multiple linear regression residuals example (outliers highlighted with yellow color). Horizontal axis – predicted y, vertical – normalized residuals.

Table 6: Multiple linear regression with lasso results (400 m dependent zone; median values).

	No transformation	Box-Cox (λ=0)	Box-Cox (λ=-0.2)
MSE	141.80	0.72	0.34
R^2	0.42	0.46	0.46
MSE (CV)	153.89	0.78	0.39
R^2 (CV)	0.34	0.43	0.43
MSE (test set)	161.83	0.75	0.38
R^2 (test set)	0.32	0.45	0.45
MSE (w/o outliers)	98.68	0.53	0.23
R^2 (w/o outliers)	0.49	0.57	0.58
MSE (w/o outliers, CV)	107.72	0.56	0.24
R^2 (w/o outliers, CV)	0.42	0.54	0.55
MSE (test set)	162.42	0.73	0.35
R^2 test set	0.31	0.44	0.44

As we can see from the table above, logarithm transformation resulted in significant improvement of model fit of 0.13 for test set. Box-Cox transformation parameter was selected as

a parameter with best score of an average of training and test set. However, its results were not much different from logarithm transformation. On the other hand, number of variables was lower comparing to logarithmic transformation and model without transformation.

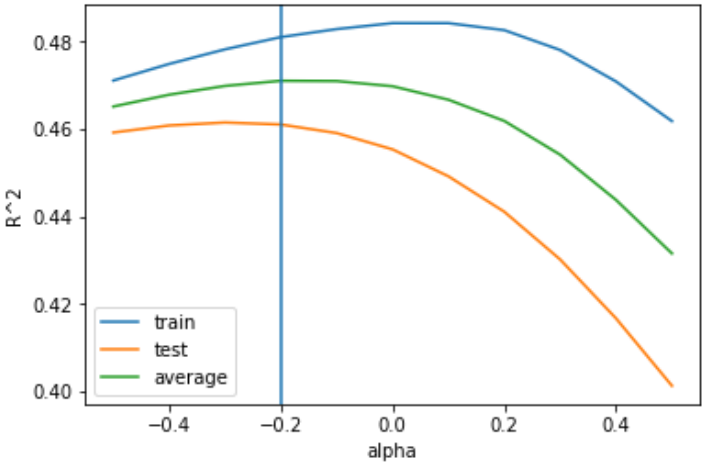

Figure 31: Box-Cox parameter selection (Multiple linear regression, 400 m dependent zone).

Table 7: Multiple linear regression with lasso results (800 m dependent zone; median values).

	No transformation	Box-Cox (λ=0)	Box-Cox (λ=-0.1)
MSE	143.09	0.72	0.48
R^2	0.41	0.47	0.47
MSE (CV)	155.08	0.78	0.54
R^2 (CV)	0.34	0.43	0.43
MSE (test)	161	0.75	0.52
R2 (test)	0.33	0.45	0.45
MSE (w/o outliers)	99.63	0.53	0.35
R^2 (w/o outliers)	0.49	0.56	0.57
MSE (w/o outliers, CV)	109.94	0.55	0.37
R^2 (w/o outliers, CV)	0.42	0.54	0.54
MSE (test)	162.15	0.738	0.49
R^2 (test)	0.33	0.44	0.44

As it is possible to notice from table with 800 m models results, these models performed practically the same, comparing to 400 m counterparts. Possibly, the fact, that 800 m models produced more multicollinear variables, that were eliminated according to VIF and Pearson thresholds, may also have influenced this. Number of variables used by corresponding models

also varies, with median values slightly higher for 800 m model non-transformed version, a little bit lower in logarithm transformation and higher for Box-Cox.
Finally, it was decided to use only 400 m models.

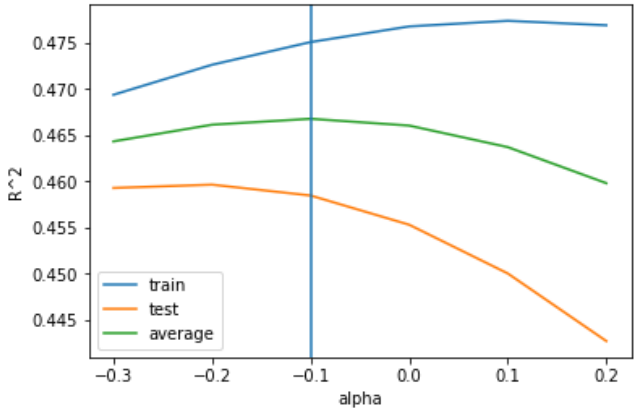

Figure 32: Box-Cox parameter selection (Multiple linear regression, 800 m dependent zone).

Gradient boosted regression

Cross-validation method with 10 folds was used for each output (for number of trees selection and separately for cross-validation). To reduce computational complexity relatively high learning rate 0.01 was used. After running a gradient boosted regression model, residuals were tested for several problems.
As GBR models are quite robust to outliers, and due to the fact, that removing of outliers has not influenced linear model test results, it was decided to skip testing models without outliers.

Quite similar pattern to linear regression could be observed with GBR models. Dependent variables transformation was used here as well. Performance of models with and without transformation was evaluated.

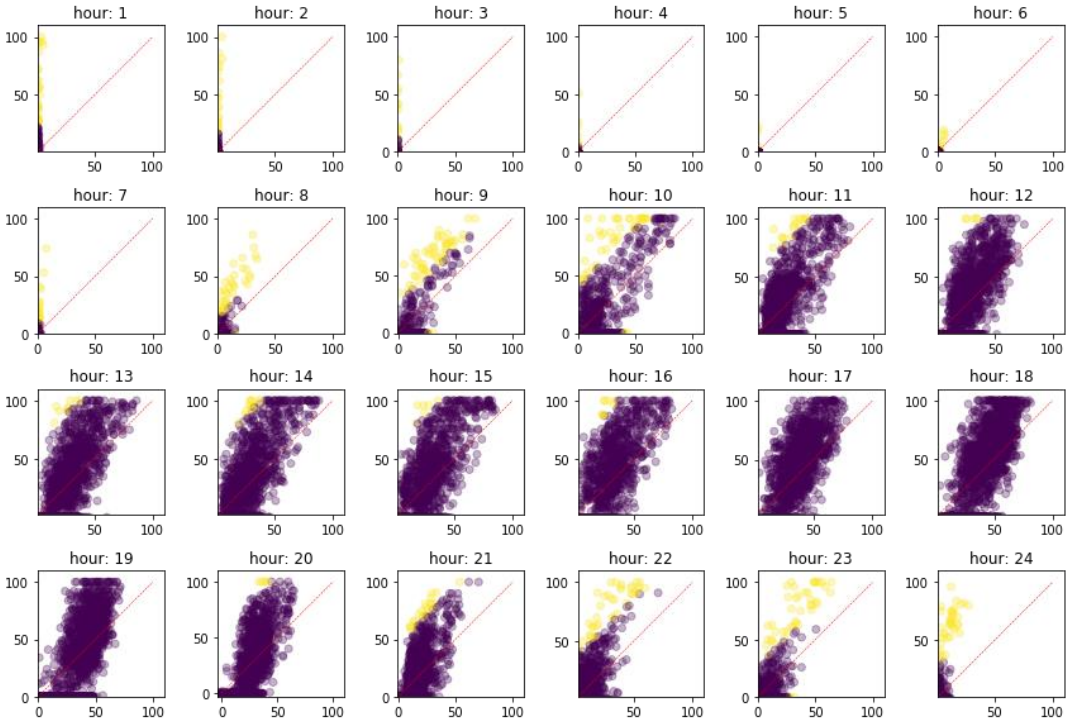

Figure 33: Predicted vs true values GBR example (outliers highlighted with yellow color). Horizontal axis – predicted y, vertical – true y.

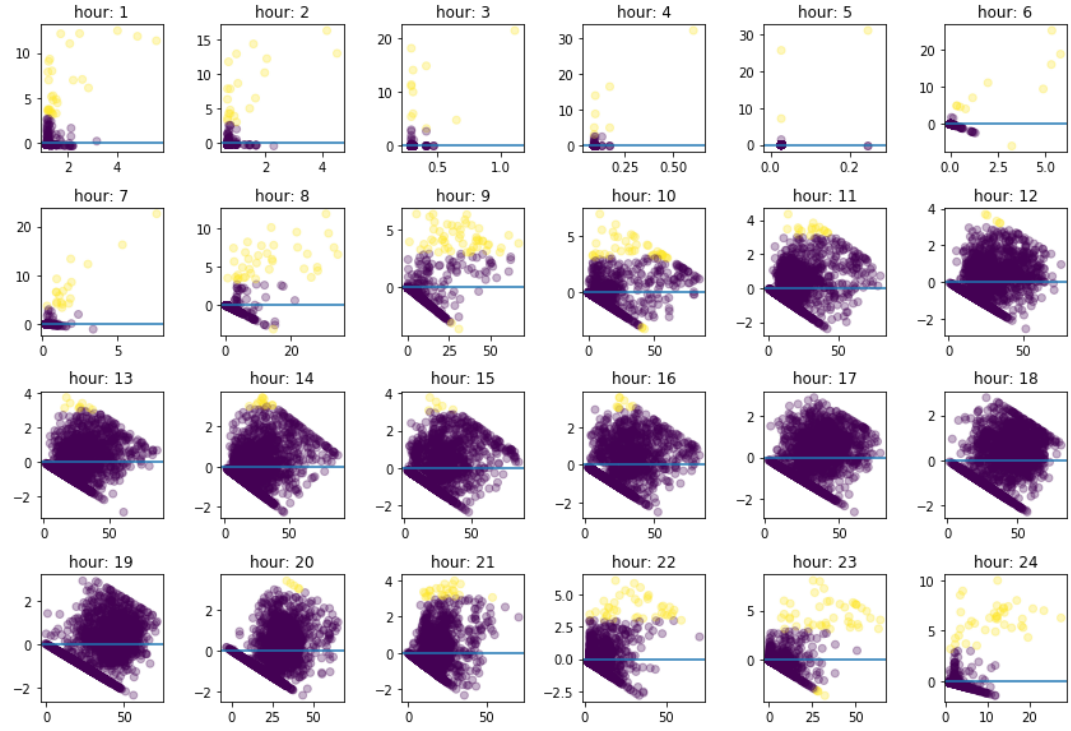

Figure 34: GBR residuals example (outliers highlighted with yellow color). Horizontal axis – predicted y, vertical – normalized residuals.

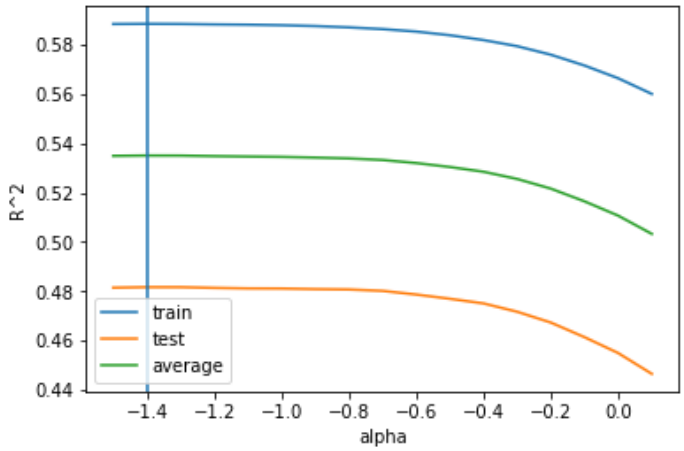

Figure 35: Box-Cox parameter selection (GBR, 400 m dependent zone).

As it is possible to see in the table below GBR provide significantly better fit for training set, comparing to linear regression. However, cross-validated and test results are quite similar to it.

Table 8: Gradient boosted regression (400 m dependent zone; median values).

	No transformation	Box-Cox (λ=0)	Box-Cox (λ=-1.4)
MSE	119.29	0.59	0.02
R^2	0.50	0.59	0.61
MSE (CV)	154.16	0.76	0.03
R^2 (CV)	0.34	0.45	0.47
MSE (test set)	162.34	0.70	0.02
R^2 (test set)	0.33	0.47	0.49

As in multiple linear regression, 800 m models do not provide significantly better fit. Therefore, it was decided to use only models with 400 m depending zone for GBR regression as well.

Table 9: Gradient boosted regression (800 m dependent zone; median values).

	No transformation	Box-Cox (λ=0)	Box-Cox (λ=-1.0)
MSE	119.64	0.58	0.02
R^2	0.50	0.61	0.61
MSE	162.15	0.70	0.02
R^2 (test set)	0.33	0.48	0.50

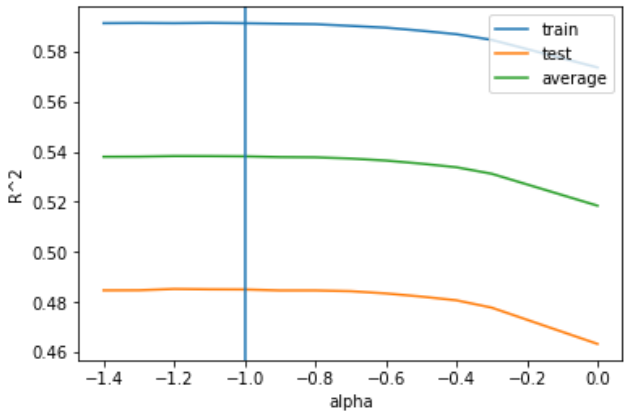

Figure 36: Box-Cox parameter selection (GBR, 800 m dependent zone).

As for data linearity, the performance of multiple linear models and GBR models is comparable in most cases. This means that the data in general may be modelled with linear regression.

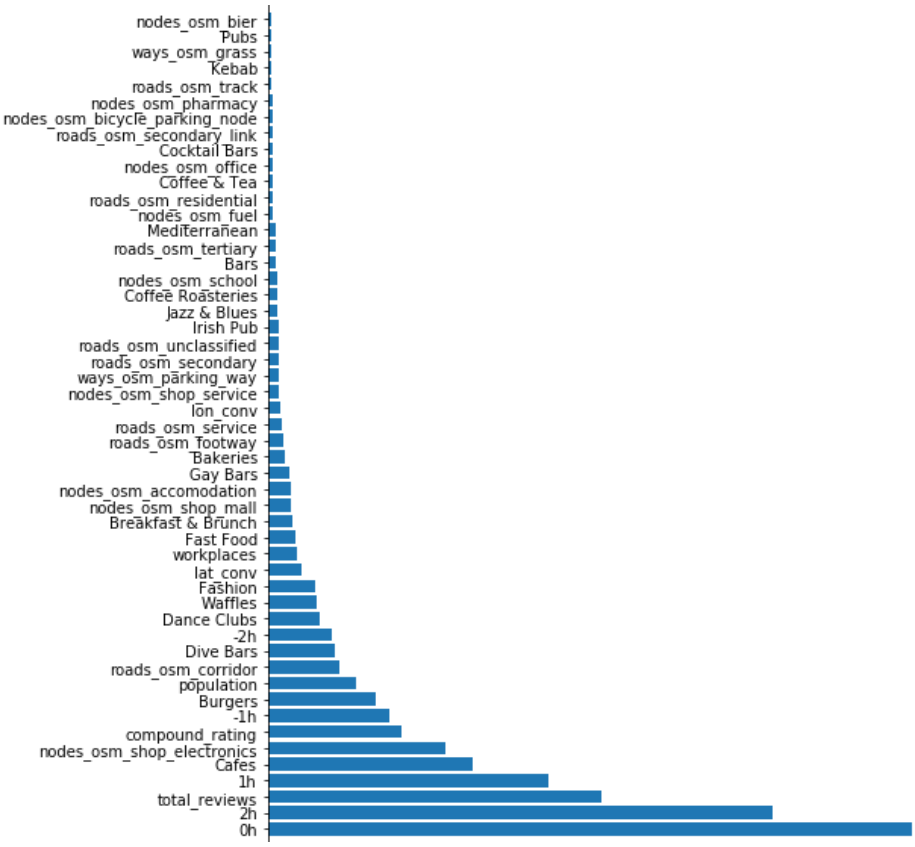

Figure 37: Most important variables within all GBR models (w/o transformation).

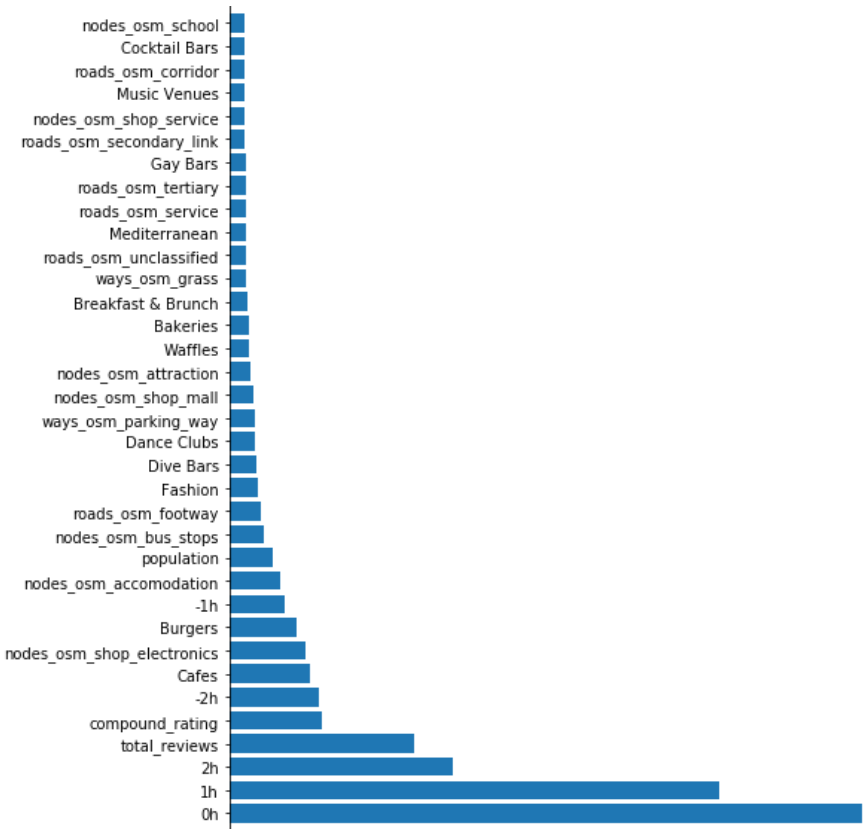

Figure 38: Most important variables within all GBR models (log transformation).

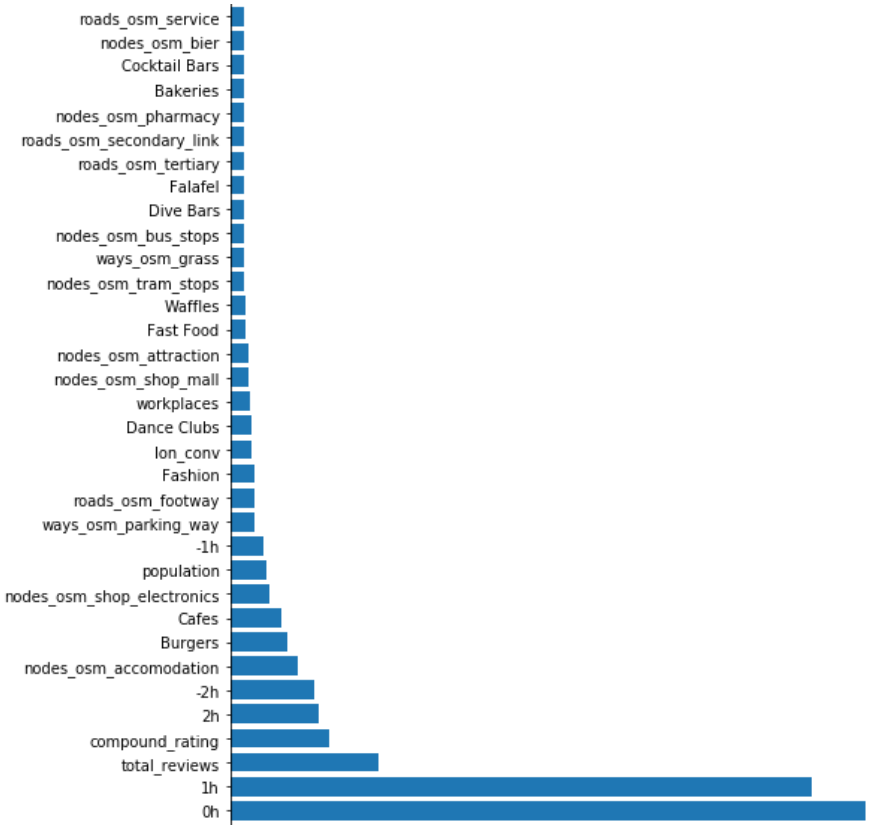

Figure 39: Most important variables within all GBR models (Box-Cox transformation).

As it is possible to see from figures above, showing most important features for all models, the number of features with sum of importances higher than threshold (0.6 used here to limit their number to a manageable level) is decreasing for transformed models. It is reduced to 35 from 51 with logarithm transformation and to 34 from 51 with Box-Cox transformation.

Selection of most important variables is quite logical as well, with current hour (i.e. that defines whether venue is opened or closed) being most important one. Although number of reviews and total rating were not thought to be good predictors, as certain venues had not enough of such data, both have significant importances. Therefore, it may be useful to collect review scores and their numbers from other sources in future.

Some venue features like “Burgers” also got significant importance, especially at certain hours early in the morning. It may have quite logical explanation, as an activity transition from clubs or bars to fast-food venues that may serve burgers and may be opened at this time.

Relatively small significance of spatial features may arise from the fact that a lot of venues with available popularity values are located close to each other. Although “nodes_osm_accomodation” – the variable that includes hotels, hostels and short term rented apartments is quite significant.

5.4.4. Evaluation

As a final step of modelling procedure, it is necessary to compare different model results.
In total 2016 models were built, however it was found out that 800 m models behaved generally similar. Therefore, it was decided to proceed with only 400 m ones, leaving 1008 models for multiple linear regression with lasso regularization and gradient boosting regression.
Removing outliers, although improving training results, did not provide better model fit, and even resulted in lower results for test set.
It was also found out that, in general, models with transformations fitted data better. As it is possible to see from figure of difference of transformed linear models with models without transformation below, the performance of the former in most cases is better by a significant margin.
Number of variables used, was not considered as a selection criterion. However, it is useful to note that median number of variables in transformed models is lower.

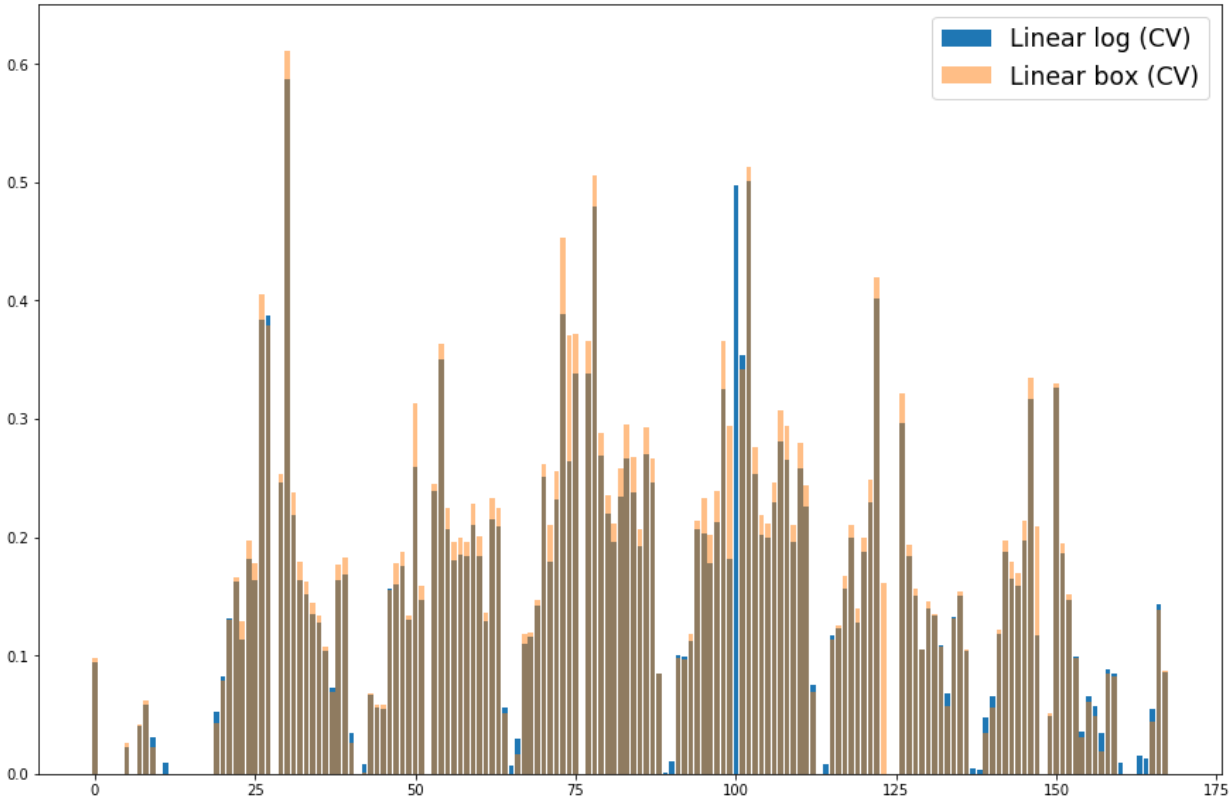

Figure 40: Difference between transformed models and models without transformation (linear regression).

Similar results were achieved with gradient boosted regression, with Box-Cox behaving slightly better than logarithm transformation, however for certain hours in the end of the week GBR with logarithm transformation achieved better results.

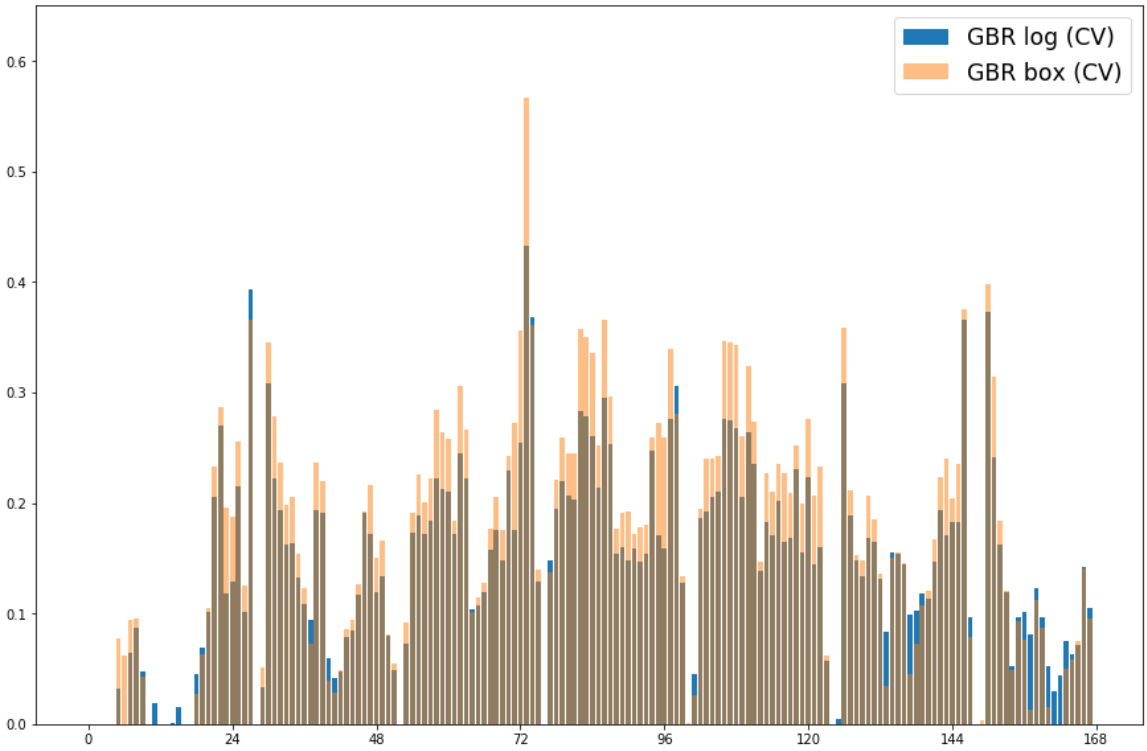

Figure 41: Difference between transformed models and models without transformation (GBR).

Finally, it is useful to compare two best performing groups of models i.e. linear and GBR with Box-Cox transformation. As it is possible to see from the figure below, in some cases GBR outperforms linear models by significant margin. Therefore, it may be concluded that GBR method with Box-Cox transformation presented the best performance among reviewed models

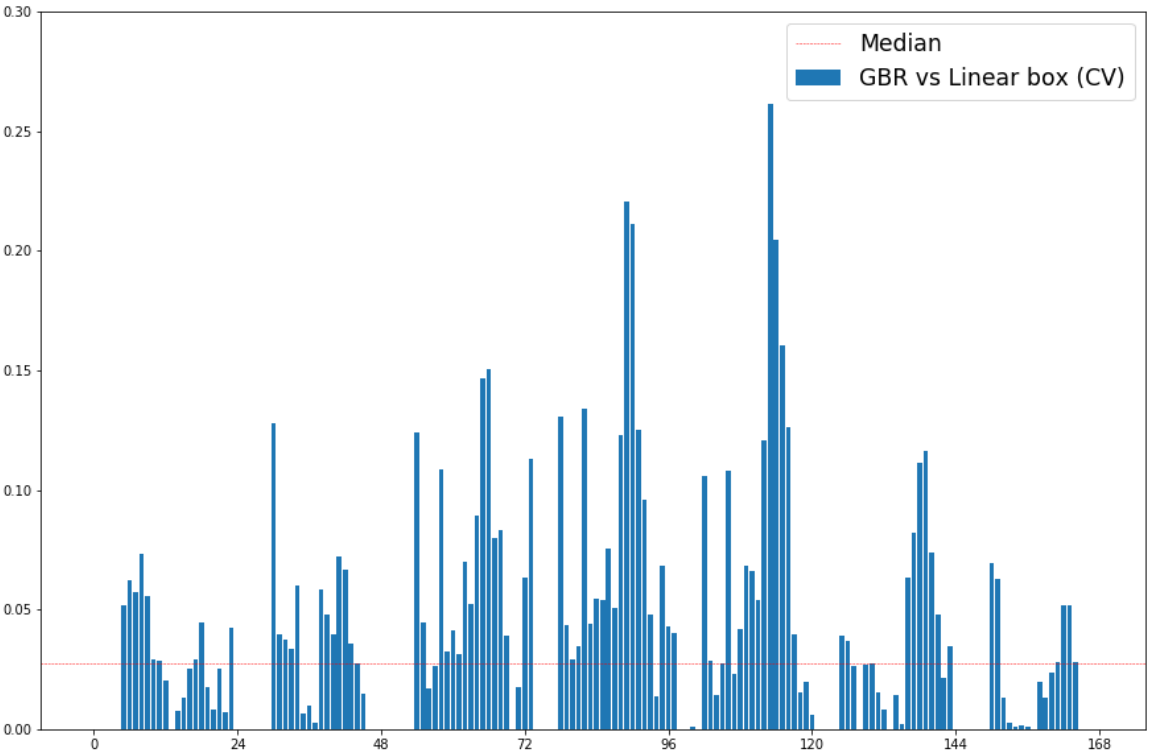

Figure 42: Difference between GRB and linear models with Box-Cox transformation.

5.5. Venue popularity measuring

5.5.1. Data collection and cleaning

As a part of test setup Raspberry Pi Zero W with SD memory card was obtained. Linux operating system Raspbian was installed on it with all necessary applications and libraries. Finally, Python script was loaded into memory.
It was noted that Raspberry PI did not have power independent clock, that sometimes may result in incorrect behavior of scripts that use system timers.
Several points within city limits were inspected with this test setup.
As a part of data cleaning procedure all signals with appearance in only one period were removed from the data.

5.5.2. Data exploration

After cleaning the data, correlation with Google popular times was checked. Correlation with several minimum and maximum thresholds was tested and the pair with maximum value was plotted for visual check. In order to deal with noise in measurements, minimum and maximum thresholds were set for length of stay in each organization. The minimum was set to 15 minutes and maximum to 180 minutes or 3 hours. It is evident that 15 minutes is relatively low bound, however certain frames in the beginning and end of a visit may be lost, so this may be good initial approximation. Maximum value is based on author's experience.

In order to get comparable subplots representing correlation, all data points were normalized.

It is necessary to note that Google data is the result of averaging over 2 weeks, therefore all spikes and drops are less visible. Moreover, only devices with enabled location history are used by Google as sources for "Popular Times" feature.

First tested point was Japanese restaurant “Takumi”. Result of WIFI data collection was quite similar to Google “Popular Times”, althoug small drop in the beginning of the operation is visible. Apart from other things, this drop might be a result of fluctuation of schedules of nearby organizations, for example as this restaurant is close Technical University of Munich (TUM), change of student activities may influence attendance. It is also possible to see that number of visitors in this restaurant is quite high for this venue type. It might be a result of influence of other facilities located nearby. Although it could be close to reality, additional WIFI monitoring devices in the area may help to clear up this question.

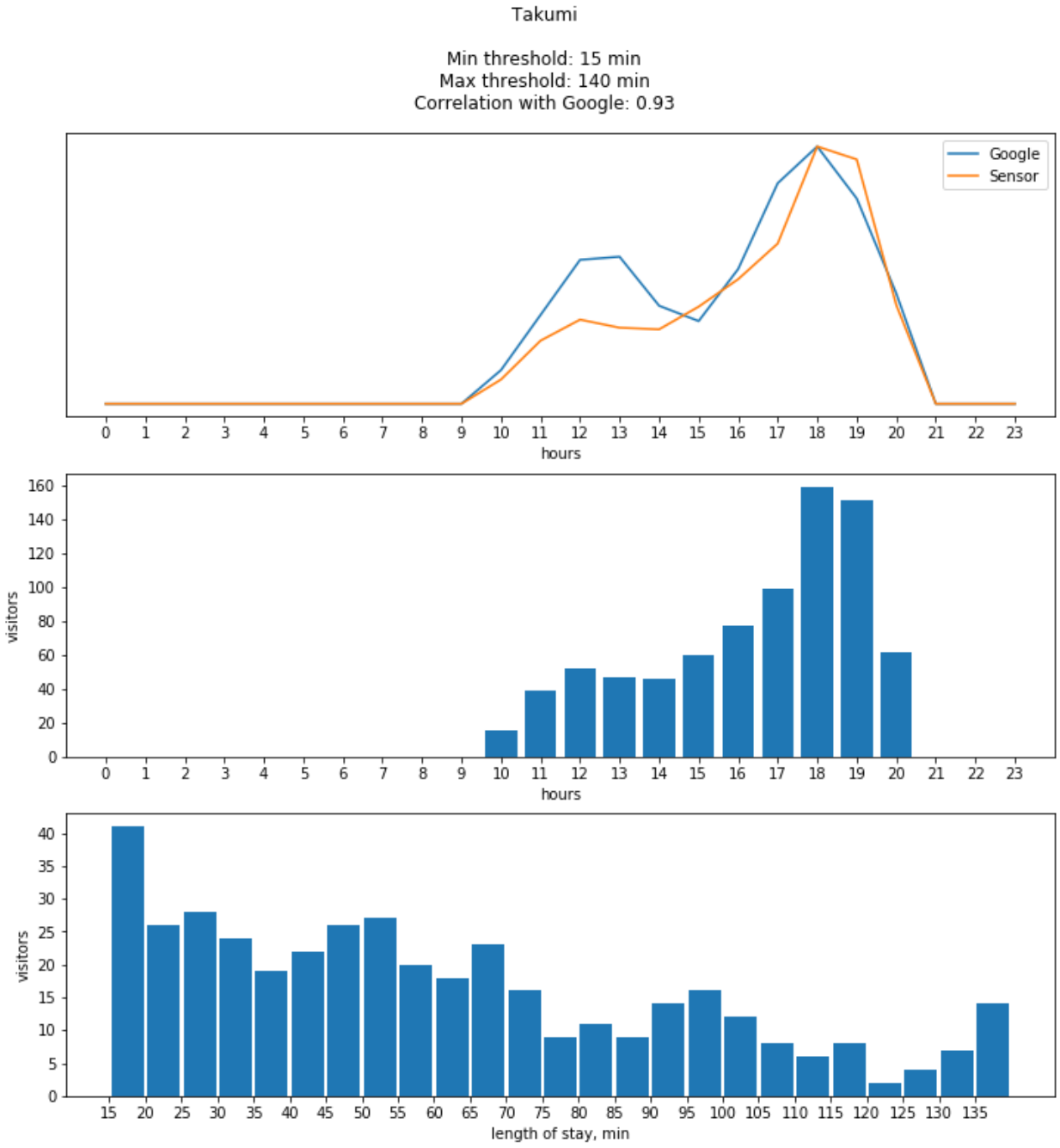

Figure 43: Venue attendance ("Takumi").

Other tested point was McDonald's restaurant near Forstenrieder Alee. As it is mentioned earlier, spikes and drops are not visible on Googles data on this venue. Significant drop is also present in sensor data comparing to Google from 17 to 19 o'clock. Several explanations of this is possible. It could be either the influence of surrounding organizations, detection problems due to building configuration and the fact that this venue has also drive-through option i.e. certain visitors may be filtered as passersby.

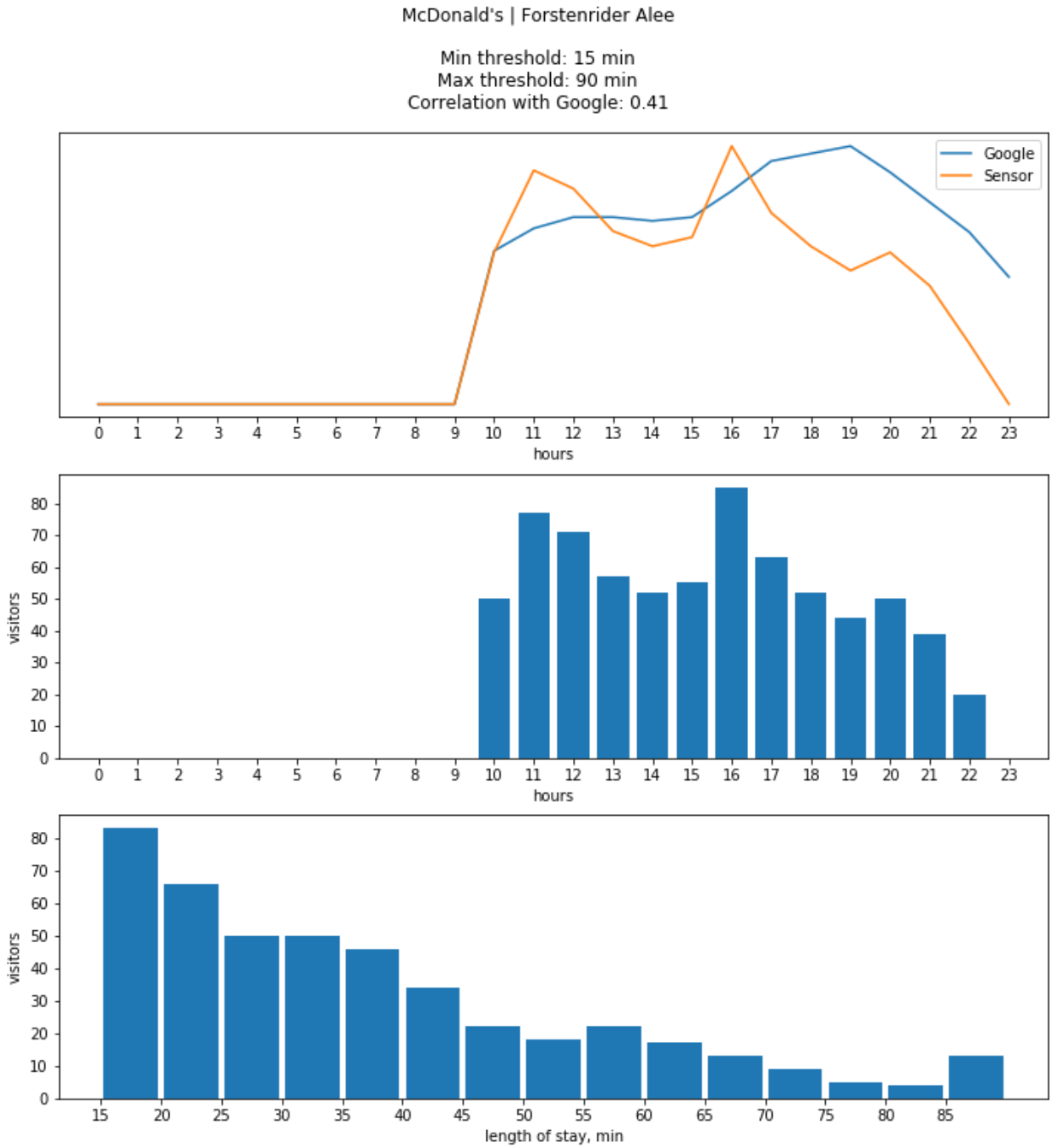

Figure 44: Venue attendance ("McDonald's" Forsetrieder Alee).

Next venue of similar type – “Burger King” near Holzapfelkirchen. It has similar spike near 12 o’clock and drop near 15, although next hours look like shifted version of Google. It might be possible, that the reason behind two similar patterns in these fast-food chains is the result of drive-through option, that is available in both venues.

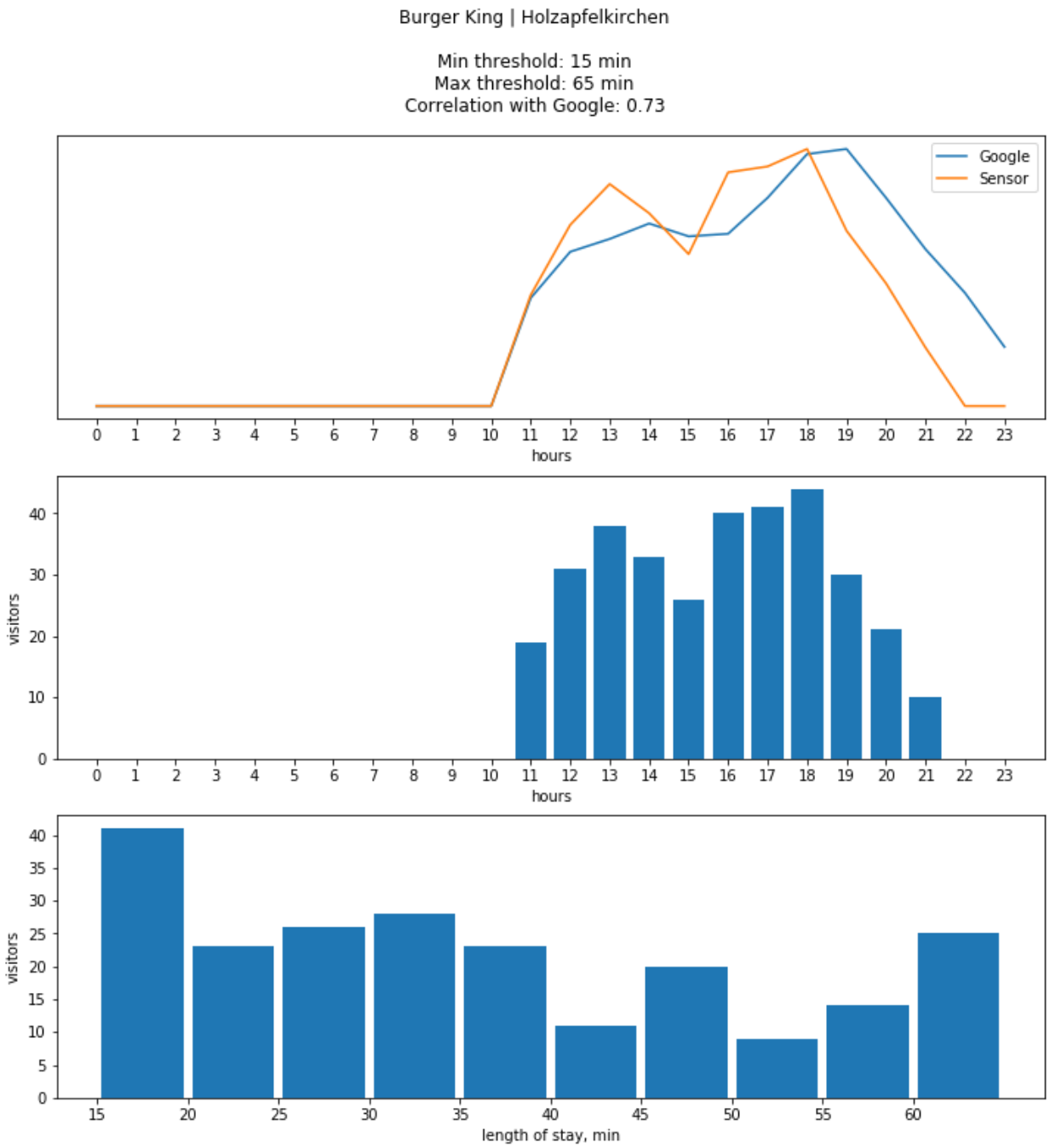

Figure 45: Venue attendance ("Burger King" Holzapfelkirchen).

Quite interesting example of nuances in Google data is shown with Loewenbraukeller restaurant. In the figure below it is possible to note that sensor line differs significantly from Google. However, for this venue several hours of real time data was available (rarely available for large venues; constant website monitoring is required). This data shows much more similar pattern to the sensor. Therefore, it is quite evident, that collection of data over two weeks could mitigate this problem.

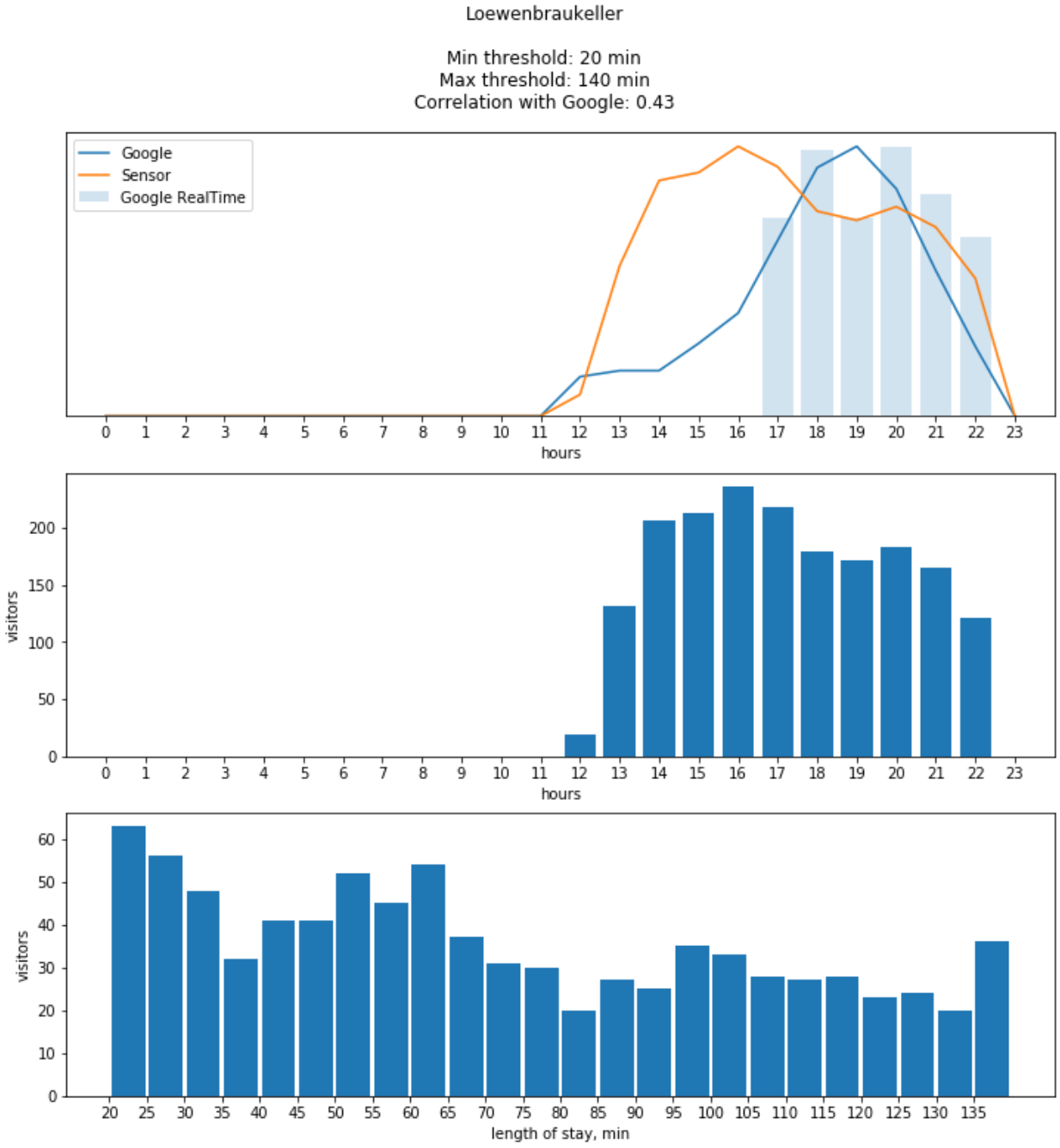

Figure 46: Venue attendance ("Loewenbraukeller").

Pattern of the cafeteria “Cardamom” below is also quite interesting. Peak in the middle of the day in sensor data, comparing to Google “Popular Times” may be a result of overlap of signals from several venues that are located quite close to each other. As a solution, it may be advised to install sensor near each venue and to define visitors of exact one by additionally analyzing received signal strength.

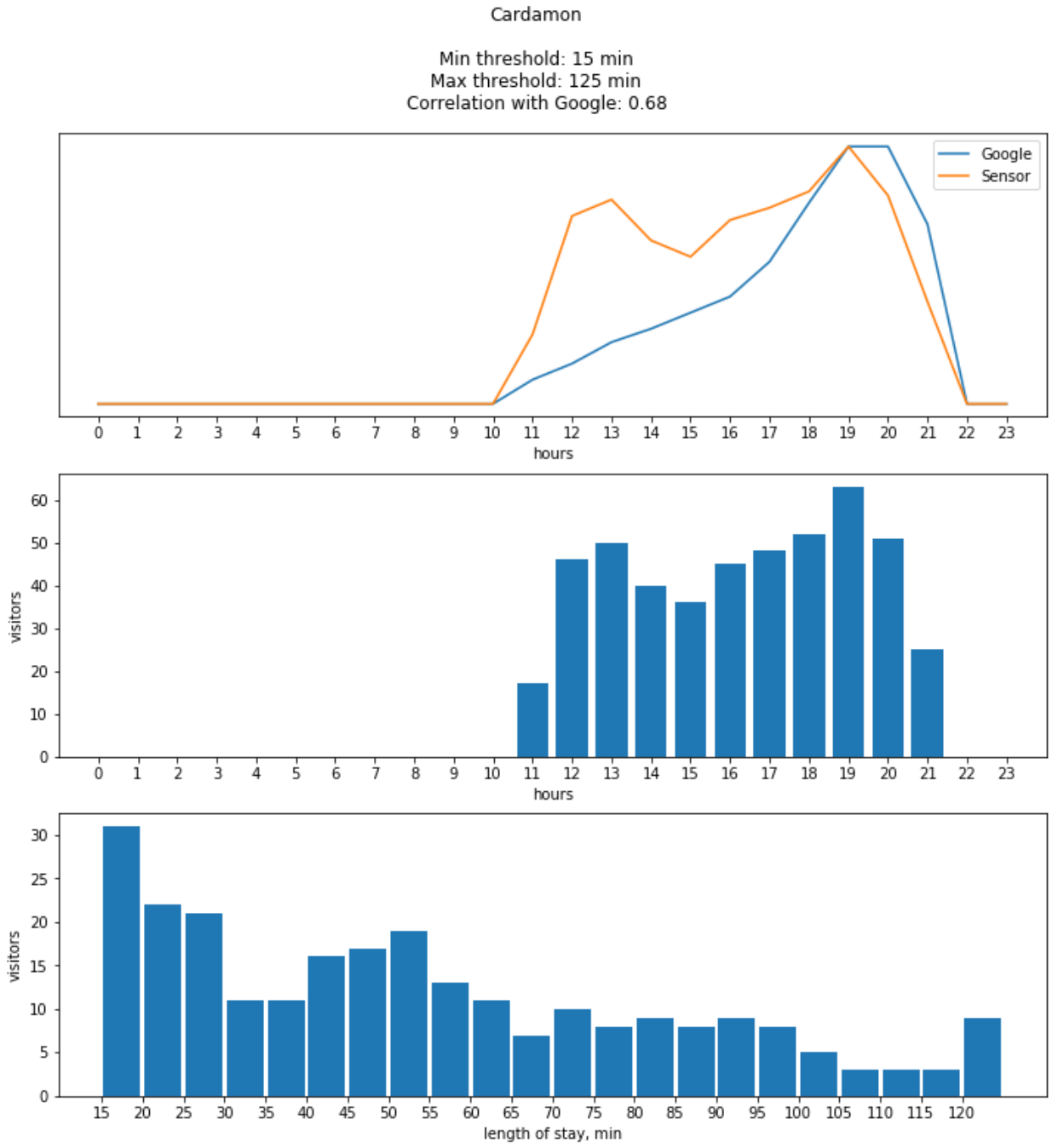

Figure 47: Venue attendance ("Cardamom").

Another venue – “Lo Studente” shows practically ideal correlation with Google. This may be a result of various factors. First of all, relatively open architecture with several tables outside main building that guarantees good signal reception by WIFI monitor. Secondly, no big overlapping facilities nearby.

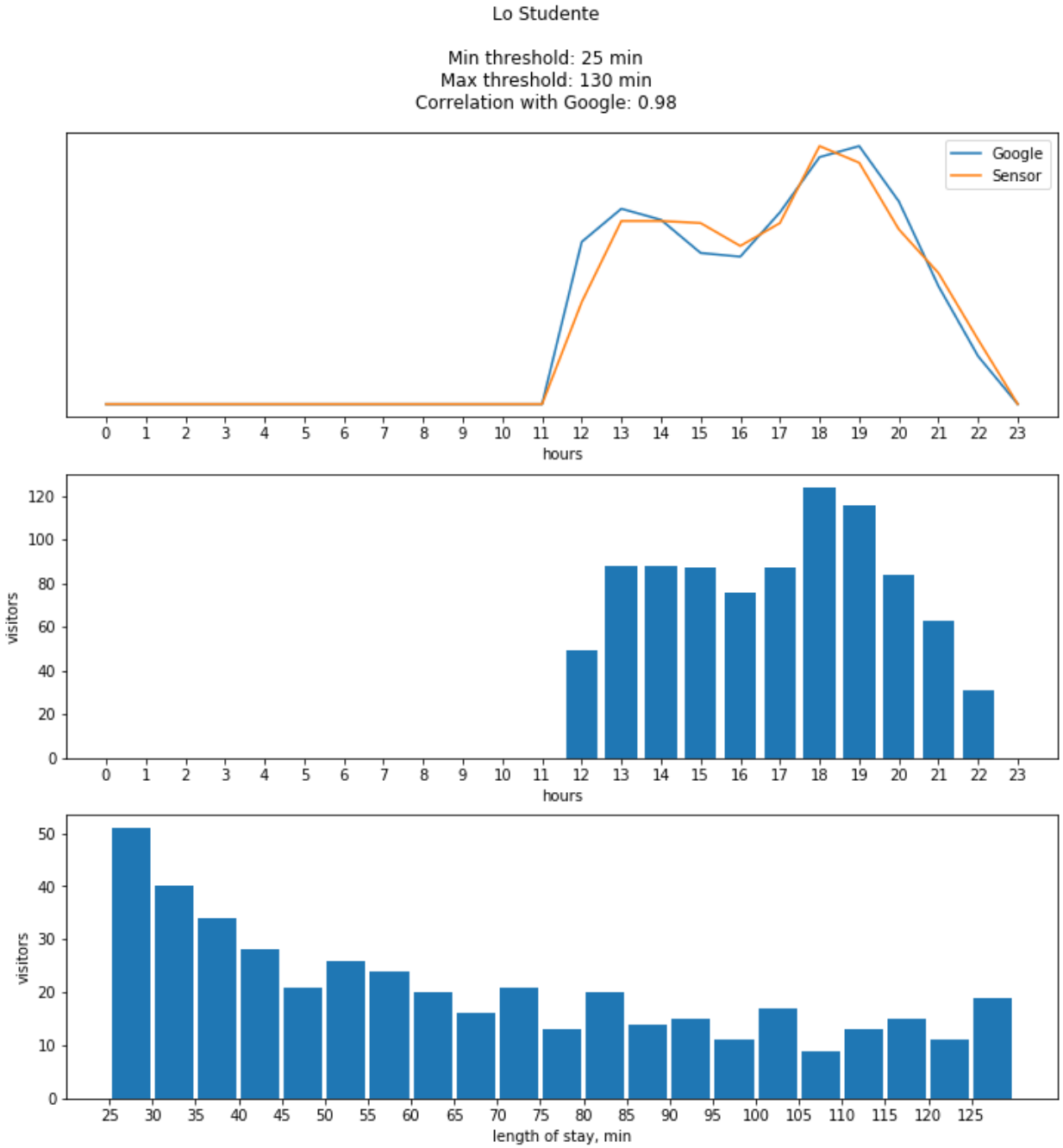

Figure 48: Venue attendance ("Lo Studente").

The “Iunu” cafeteria pattern is also quite similar to Google. Bursts in the beginning and end of a working hours may be explained by the fact that it is located near a small park with several benches, so sensor may have captured some people resting on them.

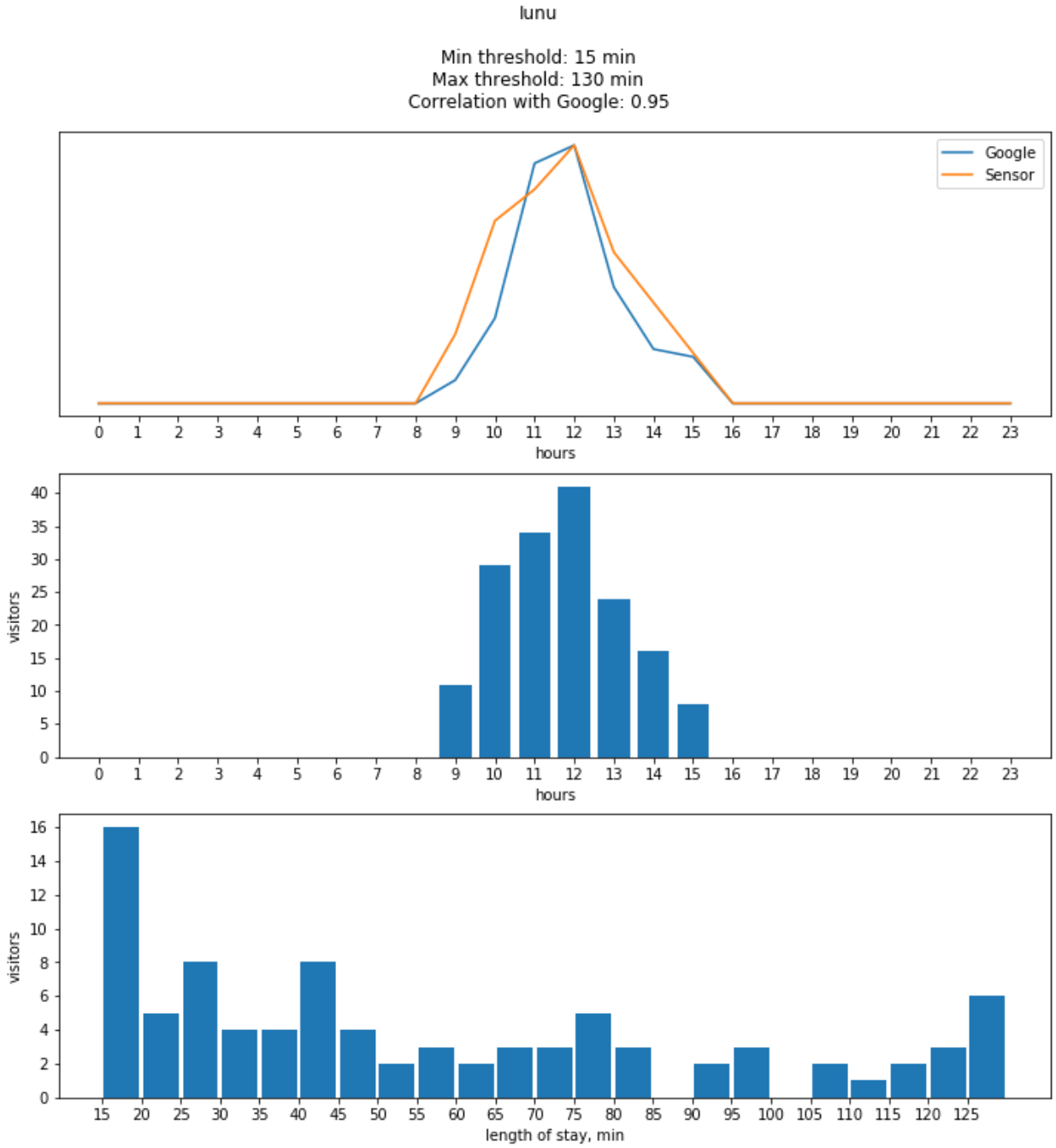

Figure 49: Venue attendance ("Iunu").

Another example of generally good correlation is “Nasca” restaurant. Several passersby may have been detected by the sensor at this place, especially in the beginning of the period, due to the fact that this venue is located on a busy street in between of TUM and Theresienstrasse subway. In the late hours the sensor data is simply unavailable.

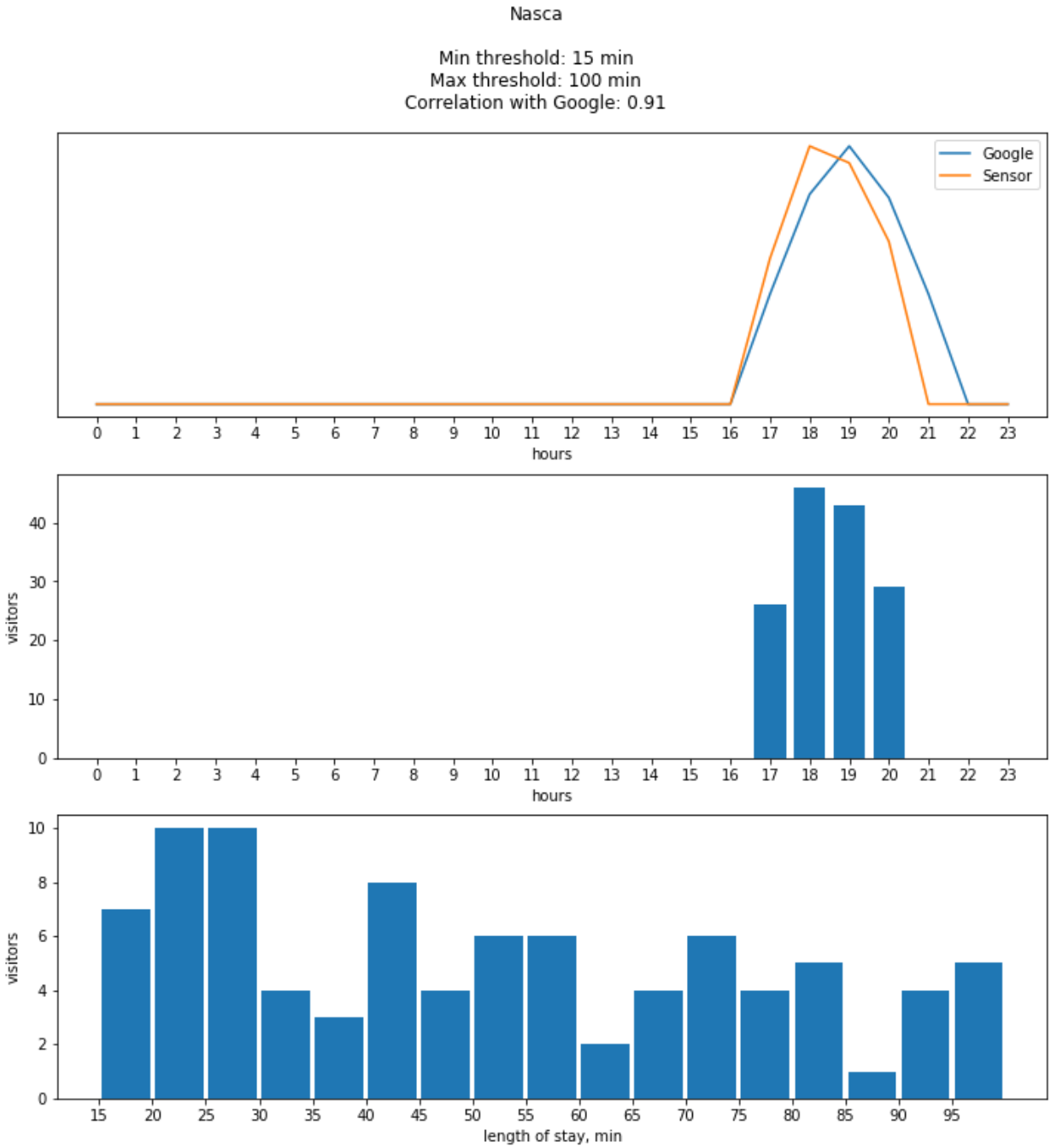

Figure 50: Venue attendance ("Nasca").

In the “Pizzeria da Antonio” below the pattern also looks relatively the same. Small drop in the beginning is a result of late start of data capture. Peak in the middle of the day may be a result of some overlapping venues, although it is hard to say this exactly with available data and due to the fact that number of visitors is relatively low. That means that even small group of people may influence the result.

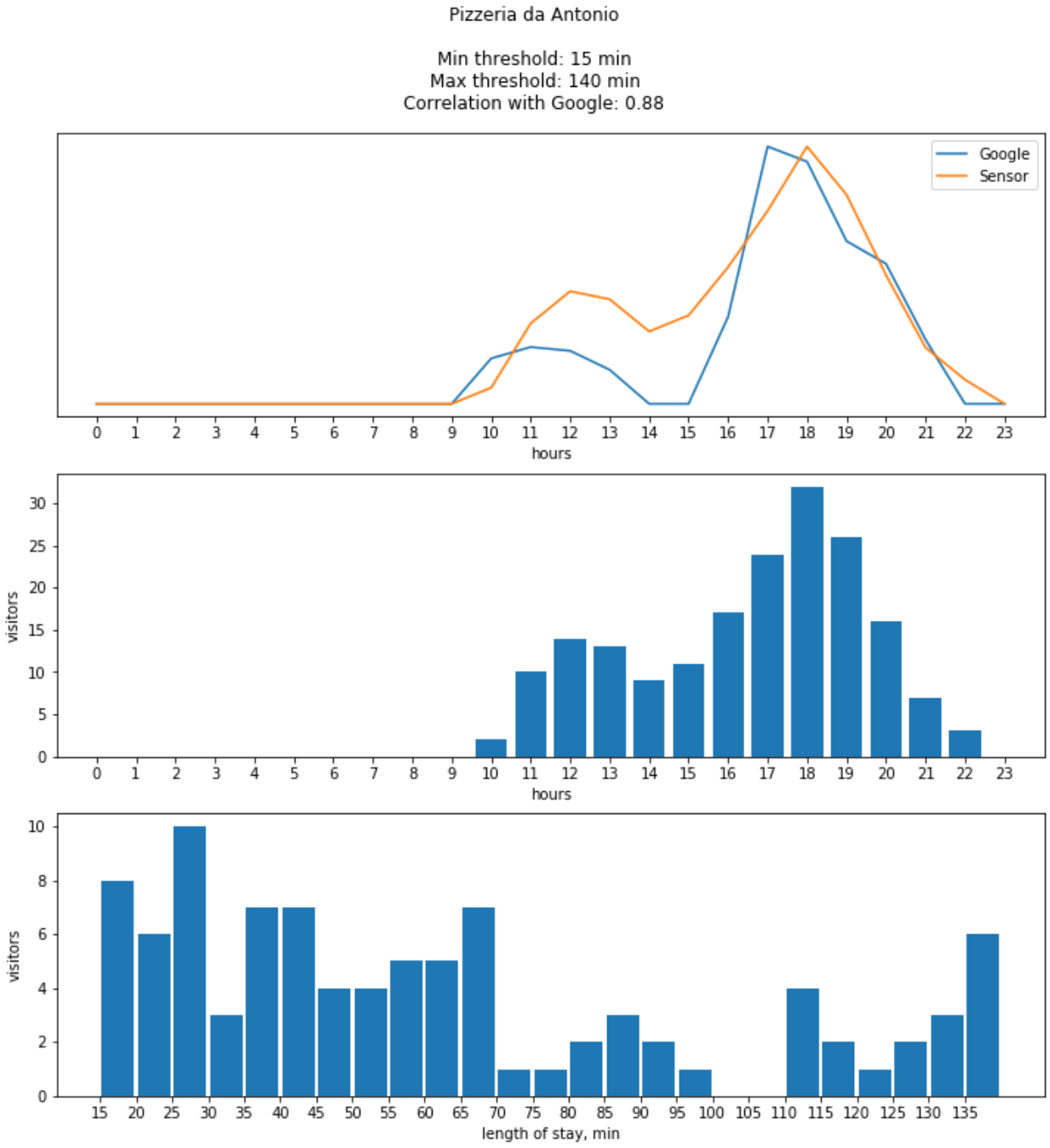

Figure 51: Venue attendance ("Pizzeria da Antonio").

In “Joon” cafeteria below slightly more people were detected comparing with Google data. It is hard to explain the reasons behind these. Therefore, it may be connected with demand fluctuations. After 20 sensor data is unavailable.

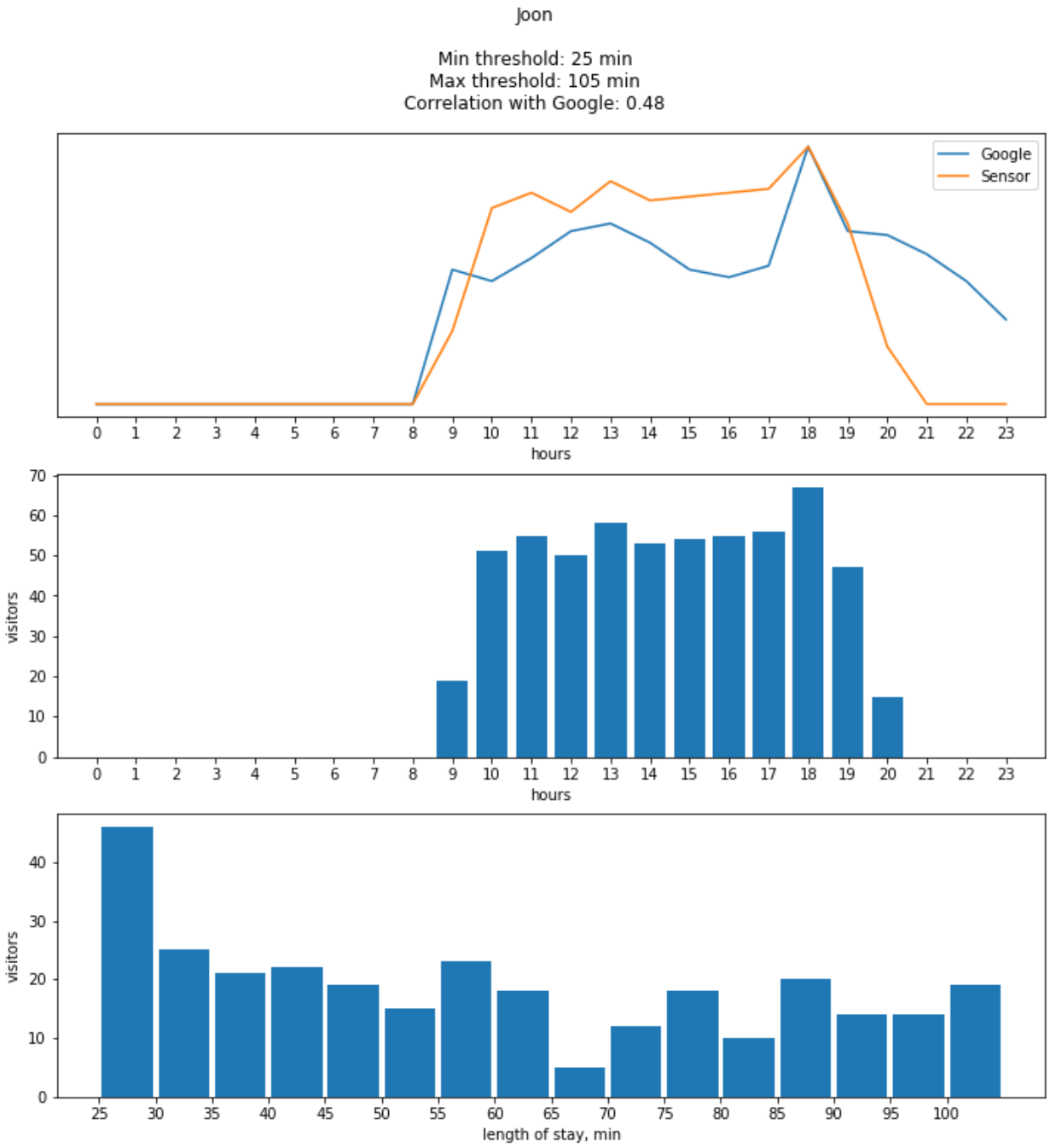

Figure 52: Venue attendance ("Joon")).

Data capturing for “KFC” cafeteria in Tal began quite close to 12 and stopped at 21, therefore it is possible to notice big dissimilarity in the beginning and end of the first subgraph. Here it is also quite clear that additional data is necessary to provide more reliable estimates.

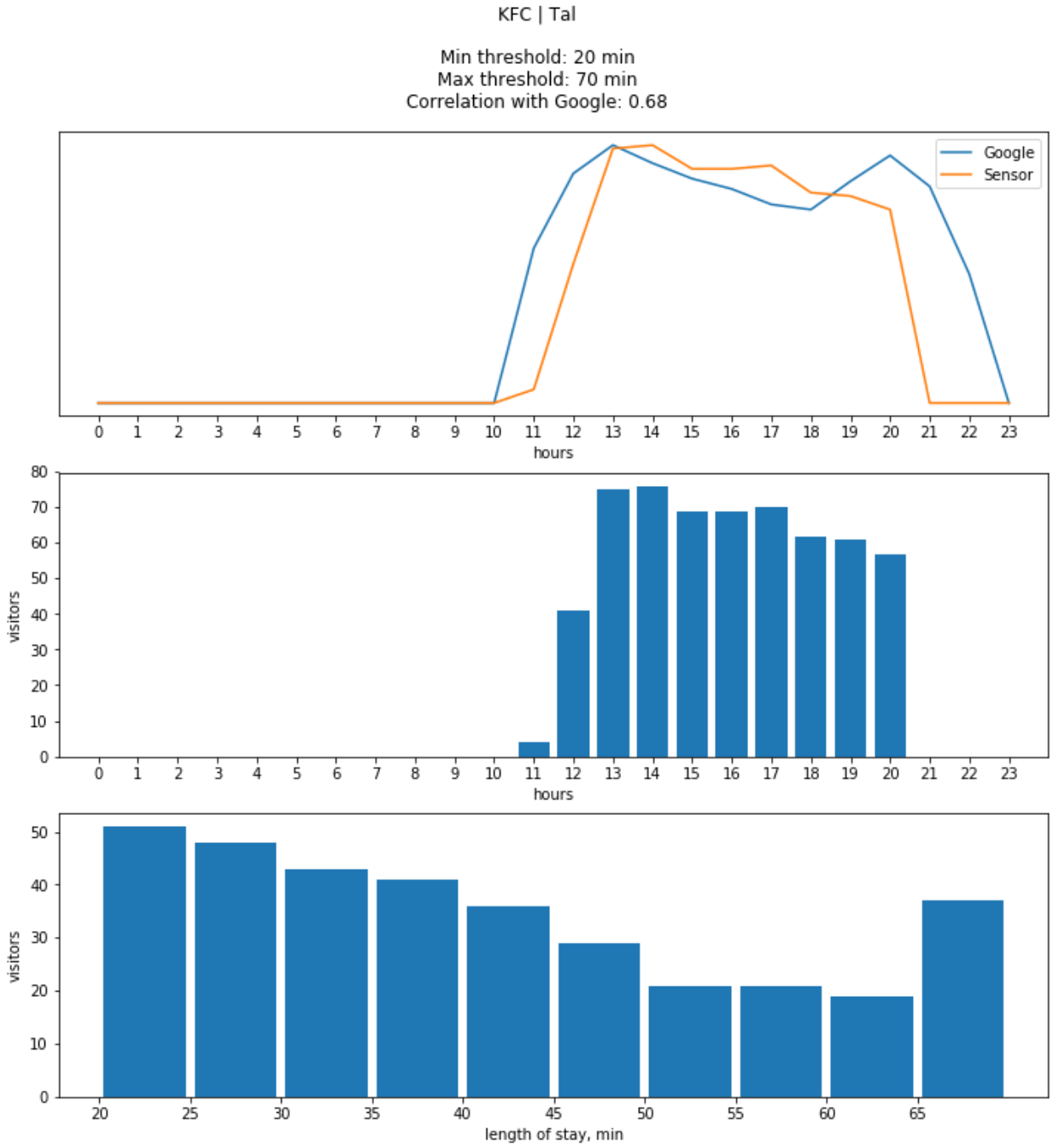

Figure 53: Venue attendance ("KFC" Tal).

The last venue – “Pavillon” also shows pattern that is quite similar to Google data. Although, number of visitors according to sensor in the middle of the day was higher. There are practically no other businesses nearby, therefore this may be a result of the fact that certain people visit nearby park for several hours. In this case additional constraint may be introduced, to count only signals with more than 2 appearances within certain range of time.

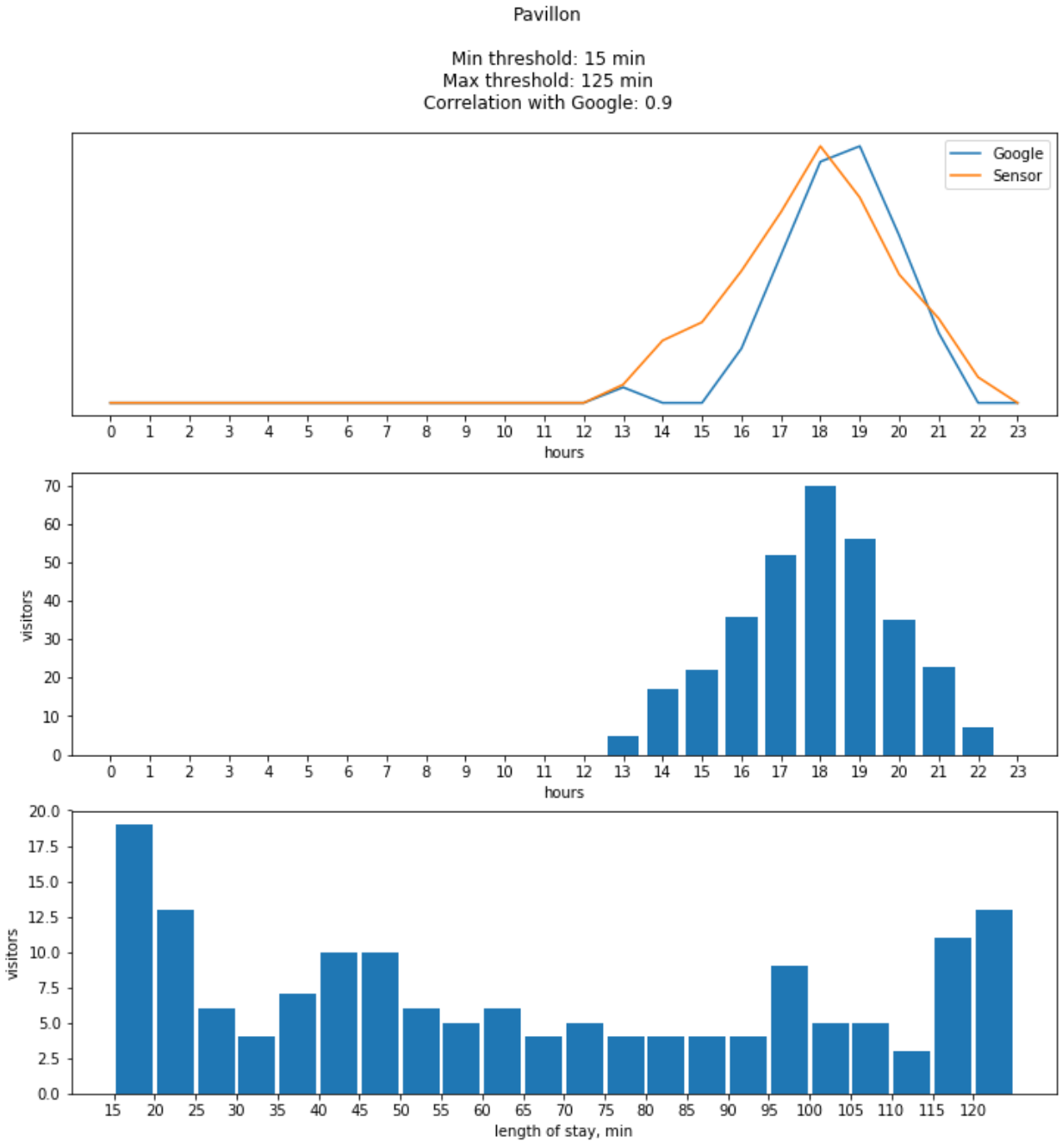

Figure 54: Venue attendance ("Pavillon" Solln).

Based on the above, it is possible to say that proposed method works quite well in obtaining occupancy data. However, continuous observations are necessary to provide reliable results and better occupancy estimates.

Certain adjustments may be useful in order to improve detection rate, although some of them are subject to legislation constraints, like active scanning instead of using only monitor mode of WIFI card. Other option may require installing several detectors and collection and analysis of network data. It may remove significant amount of noise from measurements and, depending on network scale, provide additional insight on users' behavior, for example trip chaining.

6. Conclusion

Research of mobility patters is without doubt very important field of studies. It becomes more and more important in today's world. And especially in transportation domain, where paradigm shift from personal transport ownership to use of services like car sharing, as well as public transport is quite evident.

One of the factors that made it possible is technological progress and recent progress in information and communication technologies (ICT) in particular.

Modern ICT in the form of web services (like maps, social networks etc.) generate enormous amounts of data that may be used in transportation planning practice and could potentially significantly improve planning procedures and reduce associated costs.

With current project, author tried to show the possibility of using services like Google Popular times and venue catalogues like Yelp, as well as OSM data for estimating of venue attendance. Although such data may not be very detailed, especially in Germany, due to its strong privacy protection policies, it was shown that it is possible to make forecasts with reasonable accuracy for at least busiest hours, when proper planning is of most importance.

As of modelling, it was shown that simple linear models may be used on par with much more demanding algorithms like GBR, for which cross-validation time was more than ten times higher. Transformation of dependent variable allowed to improve prediction power of models even further. It was also evident that for certain hours (for example late evening and early morning hours) forecasting potential leaves much to be desired, therefore looking for sources of additional information is advised.

It was also shown that simple and low-cost device may effectively monitor venue attendance, and with proper tuning, may provide important information on people travel behavior and changing patterns, due to certain developments. Collection of this information during long term periods may further improve mobility planning potential, as well as forecasting in other relevant fields. Of course, suggested setup has quite evident drawback – the system is the using autonomous power supply that limits information collection capabilities. It also has no wireless communication means to send information to a server, though it could be installed quite easily, with slight increase in price.

In addition to the above, certain techniques were developed for data collection from Yelp and Google, that may reduce time necessary for collection of information for further research. Information processing framework was also introduced to combine above data with OSM features, population/workplace numbers and potentially all spatial related features available.

Algorithm developed for WIFI monitoring is simple and easily portably to other languages, as it has minimum Python specific dependences, and therefore may be used on other hardware platforms, given proper WIFI library/driver availability.

6.1. Further research

Certain topics mentioned in this study require further research.

Collection of additional data

- Research of factors and sources that were not considered in current study, for example Foursquare and Facebook data, along with detailed analysis of contents of venue reviews, possibly with the help of neural networks.
- Large scale WIFI data collection with a network of devices.

Improvements in data analysis

- Improvement of clustering of venues with additional data collected.
- Research of other regression techniques with different parameters that may provide better model fit.
- Reliable estimation of true number of visitors and development of a model that will reliably forecast venue attendance numbers.

Case study approach expansion

- Research of the possibility of using similar approach in other locations, especially those that may have more data for analysis

Other

- Research of the legal possibilities of active (as opposite to passive monitoring) WIFI data collection.

7. Literature

Abrishami, S., P. Kumar and W. Nienaber (2017). Smart Stores: A scalable foot traffic collection and prediction system. Industrial Conference on Data Mining, Springer.

Aurenhammer, F. (1991). "Voronoi diagrams—a survey of a fundamental geometric data structure." ACM Computing Surveys (CSUR) **23**(3): 345-405.

Barandiaran, I. (1998). "The random subspace method for constructing decision forests." IEEE transactions on pattern analysis and machine intelligence **20**(8).

Breiman, L. (1996). "Bagging predictors." Machine learning **24**(2): 123-140.

Caliński, T. and J. Harabasz (1974). "A dendrite method for cluster analysis." Communications in Statistics-theory and Methods **3**(1): 1-27.

Chaniotakis, E., C. Antoniou, J. M. S. Grau and L. Dimitriou (2016). Can Social Media data augment travel demand survey data? Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on, IEEE.

Cortez, P., L. M. Matos, P. J. Pereira, N. Santos and D. Duque (2016). Forecasting store foot traffic using facial recognition, time series and support vector machines. International Joint Conference SOCO'16-CISIS'16-ICEUTE'16, Springer.

Deveaud, R., M.-D. Albakour, C. Macdonald and I. Ounis (2015). Experiments with a venue-centric model for personalisedand time-aware venue suggestion. Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM.

Dudani, S. A. (1976). "The distance-weighted k-nearest-neighbor rule." IEEE Transactions on Systems, Man, and Cybernetics(4): 325-327.

Friedman, J., T. Hastie and R. Tibshirani (2001). The elements of statistical learning, Springer series in statistics New York, NY, USA:.

Friedman, J. H. (2001). "Greedy function approximation: a gradient boosting machine." Annals of statistics: 1189-1232.

Friedman, L. and M. Wall (2005). "Graphical views of suppression and multicollinearity in multiple linear regression." The American Statistician **59**(2): 127-136.

Garber, L. (2013). "Analytics goes on location with new approaches." Computer(4): 14-17.

Ghimire, B., J. Rogan, V. R. Galiano, P. Panday and N. Neeti (2012). "An evaluation of bagging, boosting, and random forests for land-cover classification in Cape Cod, Massachusetts, USA." GIScience & Remote Sensing **49**(5): 623-643.

Greene, W. H. (2002). Econometric Analysis (5th Edition). Upper Saddle River, New Jersey, Prentice Hall.

Grewal, R., J. A. Cote and H. Baumgartner (2004). "Multicollinearity and measurement error in structural equation models: Implications for theory testing." Marketing science **23**(4): 519-529.

Gu, Y., Z. S. Qian and F. Chen (2016). "From Twitter to detector: Real-time traffic incident detection using social media data." Transportation research part C: emerging technologies **67**: 321-342.

Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching, ACM.

Hair, B. Babin, & Anderson (2009) Hair, JF, Black, WC, Babin, BJ, & Anderson, RE (2009). Multivariate data analysis, Upper Saddle River, NJ: Prentice Hall.[Google Scholar].

Halkidi, M. and M. Vazirgiannis (2001). Clustering validity assessment: Finding the optimal partitioning of a data set. Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, IEEE.

Hassani, M. and T. Seidl (2017). "Using internal evaluation measures to validate the quality of diverse stream clustering algorithms." Vietnam Journal of Computer Science **4**(3): 171-183.

Keogh, E. and C. A. Ratanamahatana (2005). "Exact indexing of dynamic time warping." Knowledge and information systems **7**(3): 358-386.

Kisilevich, S., D. Keim and L. Rokach (2013). "A GIS-based decision support system for hotel room rate estimation and temporal price prediction: The hotel brokers' context." Decision Support Systems **54**(2): 1119-1133.

Kullback, S. and R. A. Leibler (1951). "On information and sufficiency." The annals of mathematical statistics **22**(1): 79-86.

Lemire, D. (2009). "Faster retrieval with a two-pass dynamic-time-warping lower bound." Pattern recognition **42**(9): 2169-2180.
Li, Y., M. Steiner, L. Wang, Z.-L. Zhang and J. Bao (2013). Exploring venue popularity in foursquare. INFOCOM, 2013 Proceedings IEEE, IEEE.
Libelium. "Smartphone Detection." Retrieved 06.08, 2018, from http://www.libelium.com/products/meshlium/smartphone-detection/.
Maaten, L. v. d. and G. Hinton (2008). "Visualizing data using t-SNE." Journal of machine learning research **9**(Nov): 2579-2605.
Manolopoulos, Y., A. Nanopoulos, A. N. Papadopoulos and Y. Theodoridis (2010). R-trees: Theory and Applications, Springer Science & Business Media.
Manotumruksa, J., C. Macdonald and I. Ounis (2016). Predicting contextually appropriate venues in location-based social networks. International Conference of the Cross-Language Evaluation Forum for European Languages, Springer.
Martin, J., T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye and D. Brown (2017). "A study of MAC address randomization in mobile devices and when it fails." Proceedings on Privacy Enhancing Technologies **2017**(4): 365-383.
Mason, C. H. and W. D. Perreault Jr (1991). "Collinearity, power, and interpretation of multiple regression analysis." Journal of marketing research: 268-280.
Meeks, W. L. and S. Dasgupta (2004). "Geospatial information utility: an estimation of the relevance of geospatial information to users." Decision Support Systems **38**(1): 47-63.
Noulas, A., S. Scellato, N. Lathia and C. Mascolo (2012). Mining user mobility features for next place prediction in location-based services. Data mining (ICDM), 2012 IEEE 12th international conference on, IEEE.
Nunes, N., M. Ribeiro, C. Prandi and V. Nisi (2017). Beanstalk: a community based passive wi-fi tracking system for analysing tourism dynamics. Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, ACM.
Ogutu, J. O., H.-P. Piepho and T. Schulz-Streeck (2011). A comparison of random forests, boosting and support vector machines for genomic selection. BMC proceedings, BioMed Central.
Pedregosa, F., O. Grisel, M. Blondel and G. Varoquaux. "Manifold learning." Retrieved 02.09, 2018, from https://scikit-learn.org/stable/modules/manifold.html.
Rodas, D. D. (2017). "Identification of spatio-temporal factors affecting arrivals and departures of shared vehicles."
Sammon, J. W. (1969). "A nonlinear mapping for data structure analysis." IEEE Transactions on computers **100**(5): 401-409.
Schulz, M. a. W., Daniel and Hollick, Matthias. (2017). "Nexmon: The C-based Firmware Patching Framework." Retrieved 01.06, 2018, from https://nexmon.org.
Shannon, C. E. (1948). "A mathematical theory of communication." Bell system technical journal **27**(3): 379-423.
Tafidis, P., J. Teixeira, B. Bahmankhah, E. Macedo, C. Guarnaccia, M. C. Coelho and J. M. Bandeira (2018). Can Google Maps Popular Times Be an Alternative Source of Information to Estimate Traffic-Related Impacts?
Wang, L., R. Gopal, R. Shankar and J. Pancras (2015). "On the brink: Predicting business failure with mobile location-based checkins." Decision Support Systems **76**: 3-13.
Willing, C., K. Klemmer, T. Brandt and D. Neumann (2017). "Moving in time and space–Location intelligence for carsharing decision support." Decision Support Systems **99**: 75-85.
Yoshimura, Y., A. Krebs and C. Ratti (2016). "An analysis of visitors' length of stay through noninvasive Bluetooth monitoring in the Louvre Museum." arXiv preprint arXiv:1605.00108.
Zhao, J., W. Deng and Y. Song (2014). "Ridership and effectiveness of bikesharing: The effects of urban features and system characteristics on daily use and turnover rate of public bikes in China." Transport Policy **35**: 253-264.

8. Appendix

8.1. Fitted versus True values

Multiple linear regression with lasso regularization (no transformation)

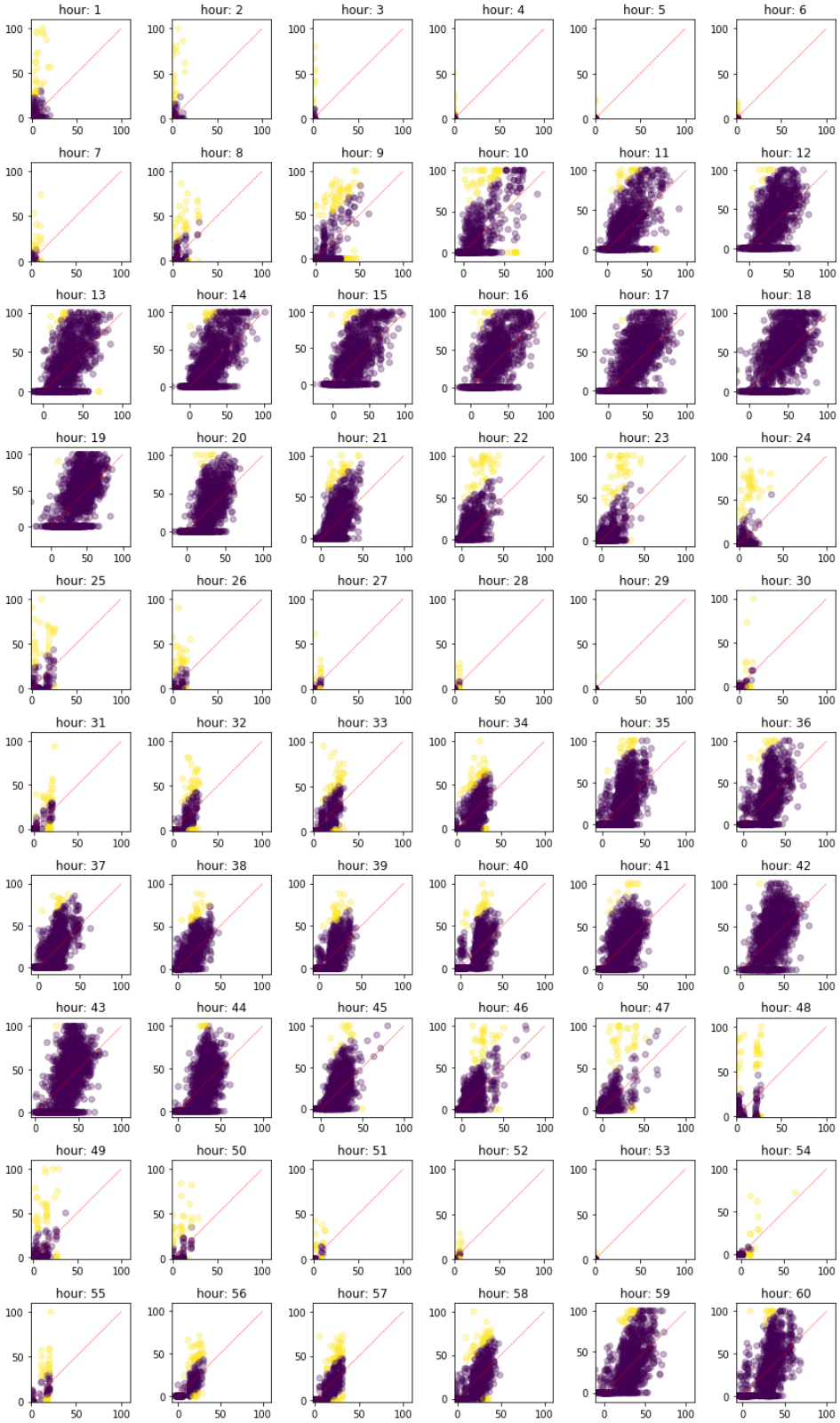

Figure 55: Multiple linear regression with lasso, fitted vs true values (part 1).

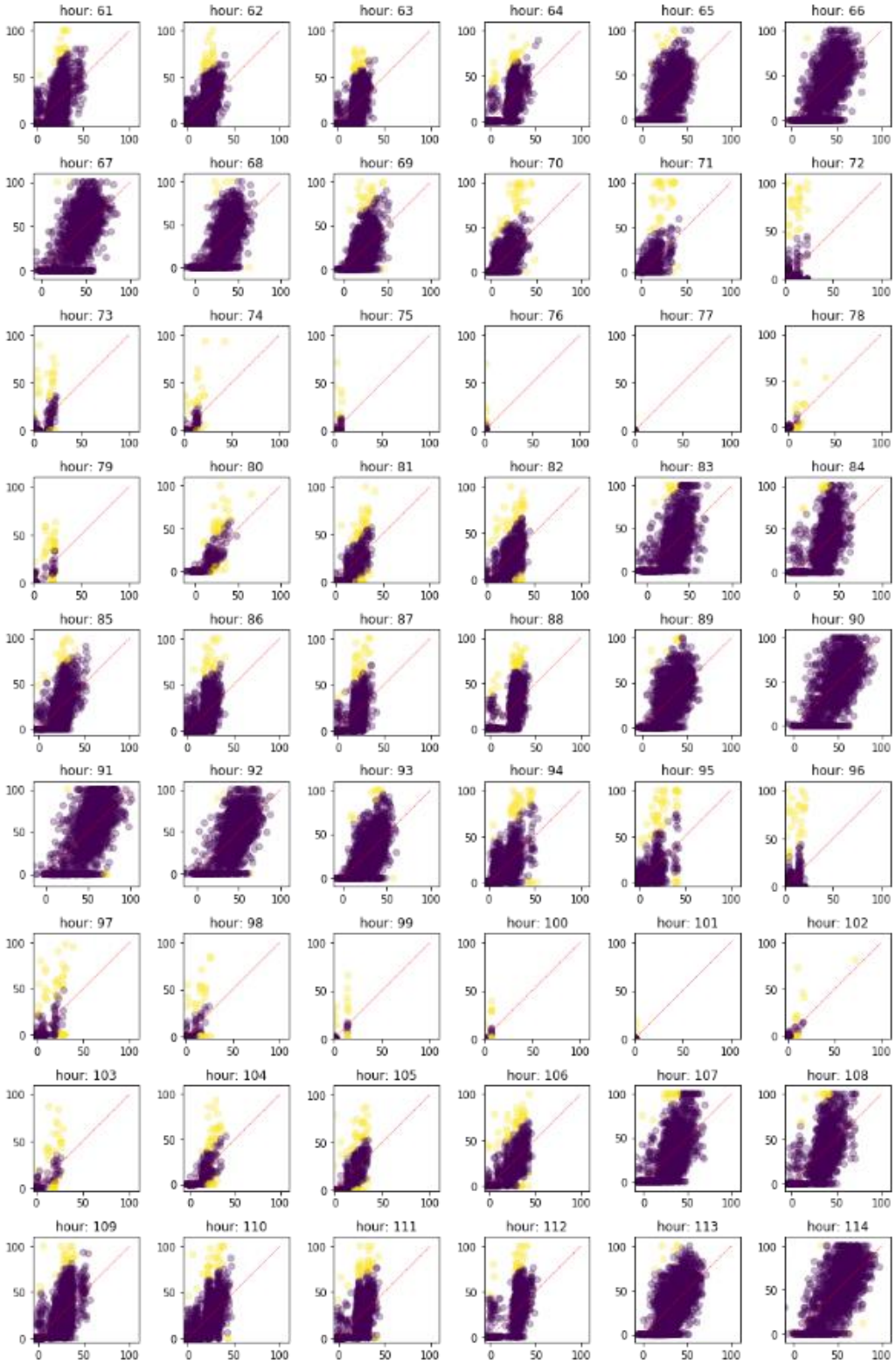

Figure 56: Multiple linear regression with lasso, fitted vs true values (part 2).

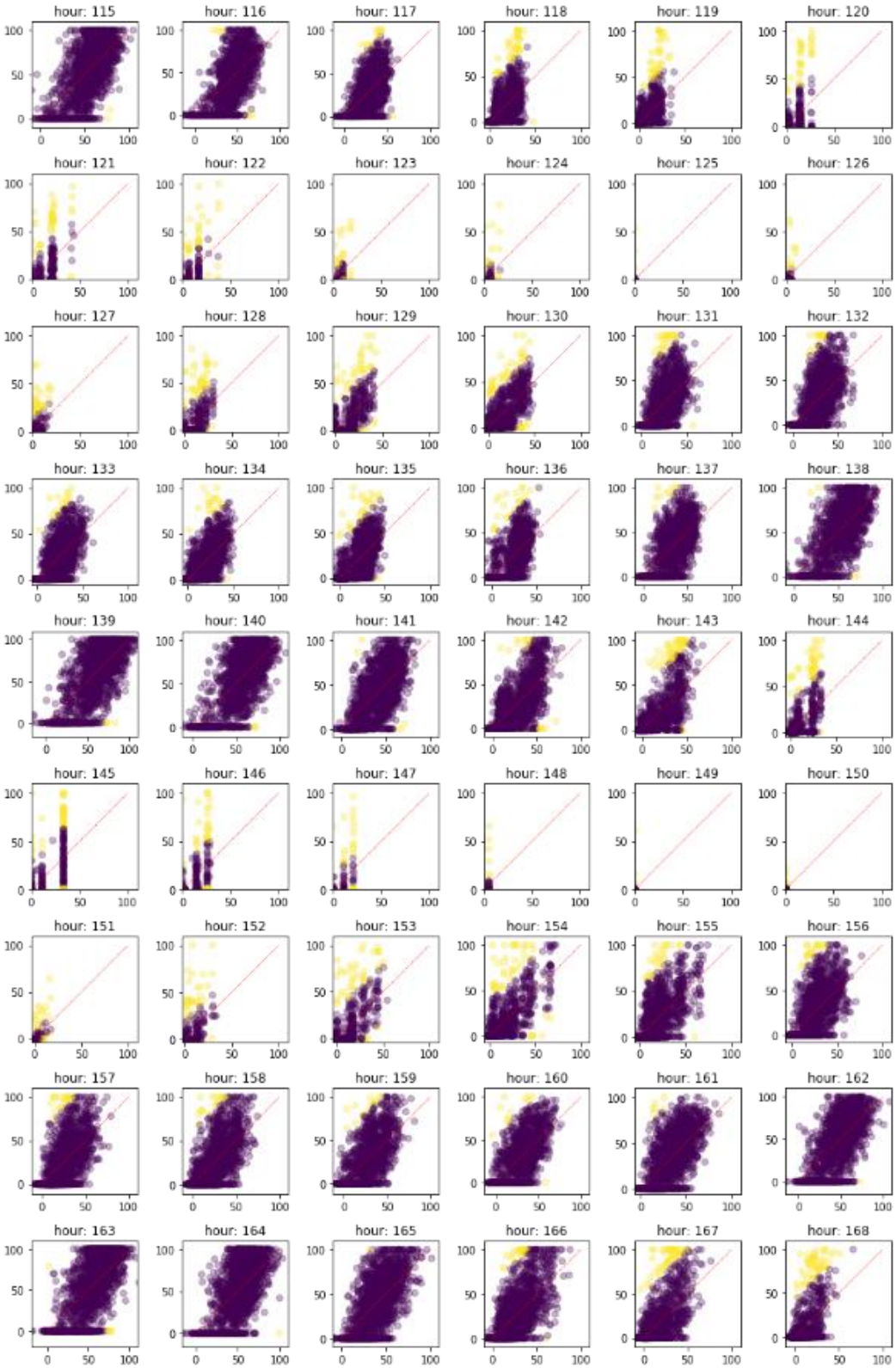

Figure 57: Multiple linear regression with lasso, fitted vs true values (part 3).

Multiple linear regression with lasso regularization (logarithm transformation)

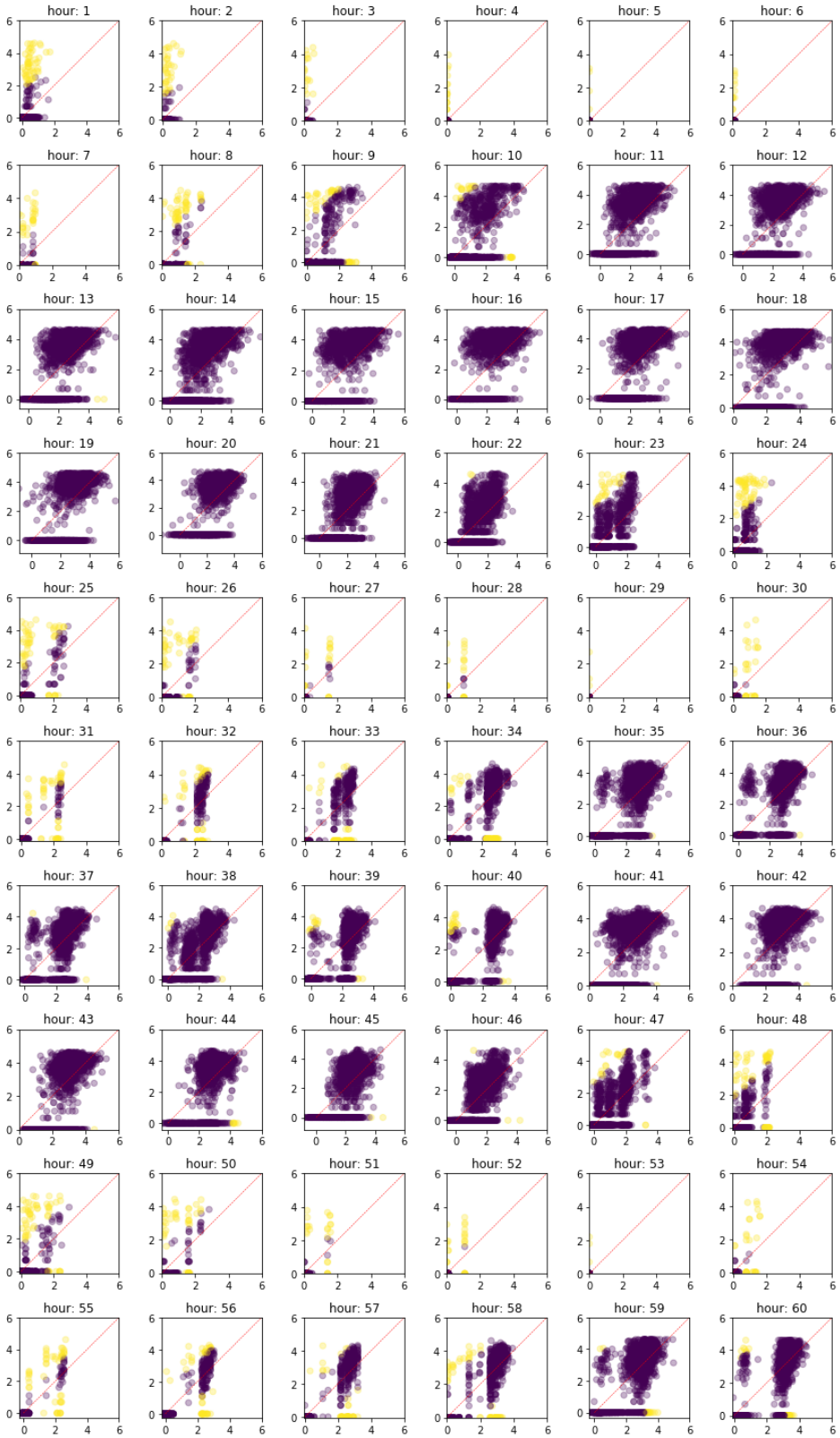

Figure 58: Multiple linear regression with lasso, logarithm transformation, fitted vs true values (part 1).

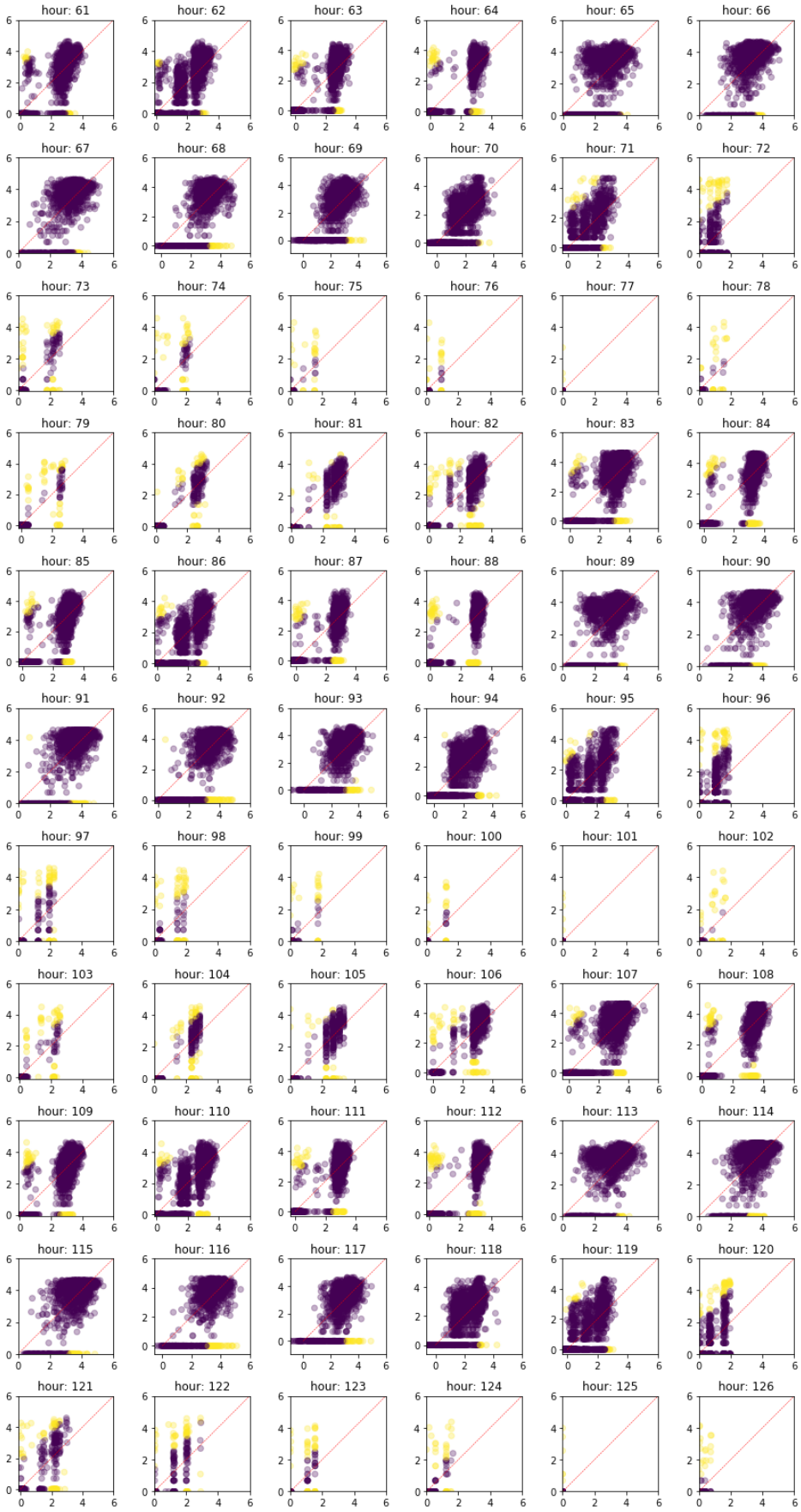

Figure 59: Multiple linear regression with lasso, logarithm transformation, fitted vs true values (part 2).

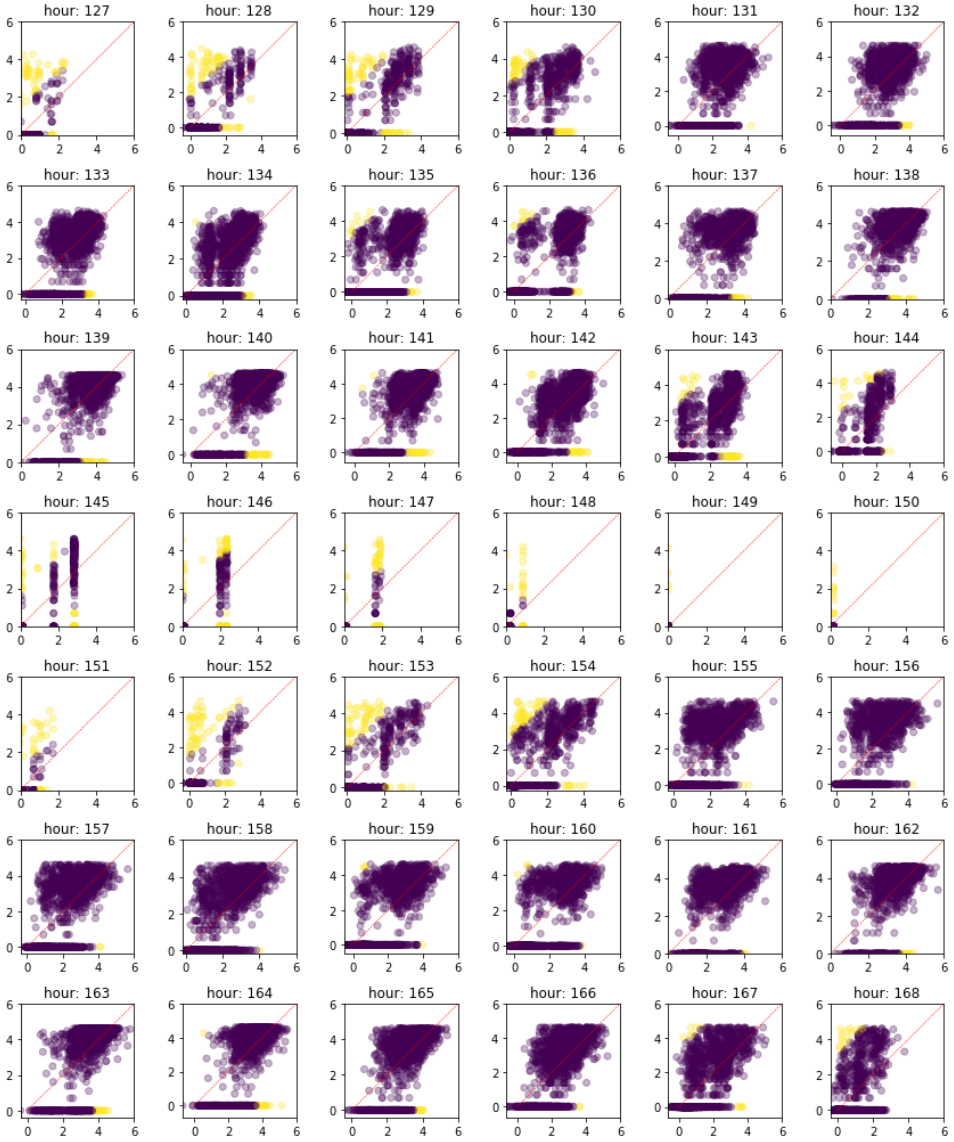

Figure 60: Multiple linear regression with lasso, logarithm transformation, fitted vs true values (part 3).

Multiple linear regression with lasso regularization (Box-Cox transformation)

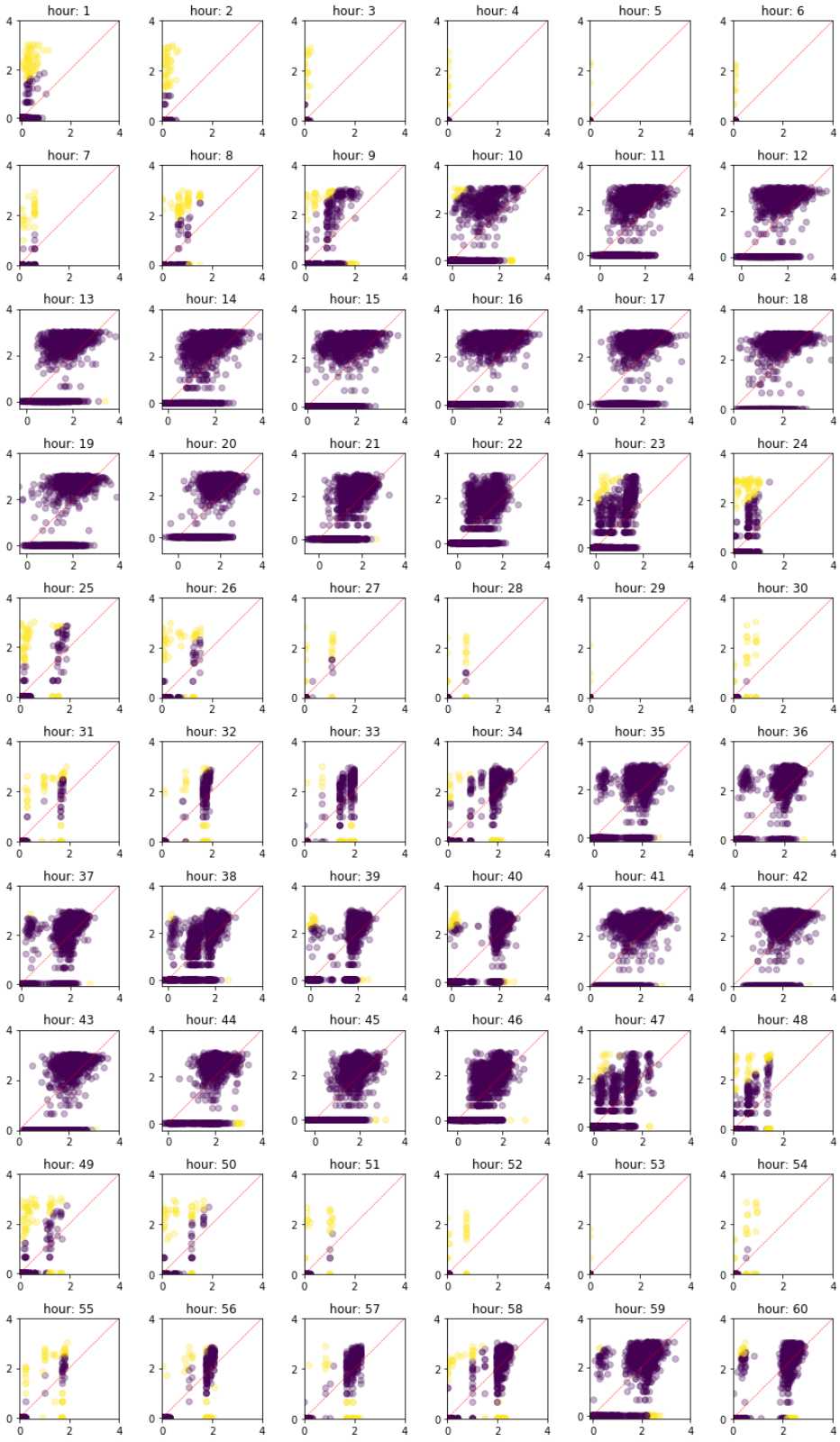

Figure 61: Multiple linear regression with lasso, Box-Cox transformation, fitted vs true values (part 1).

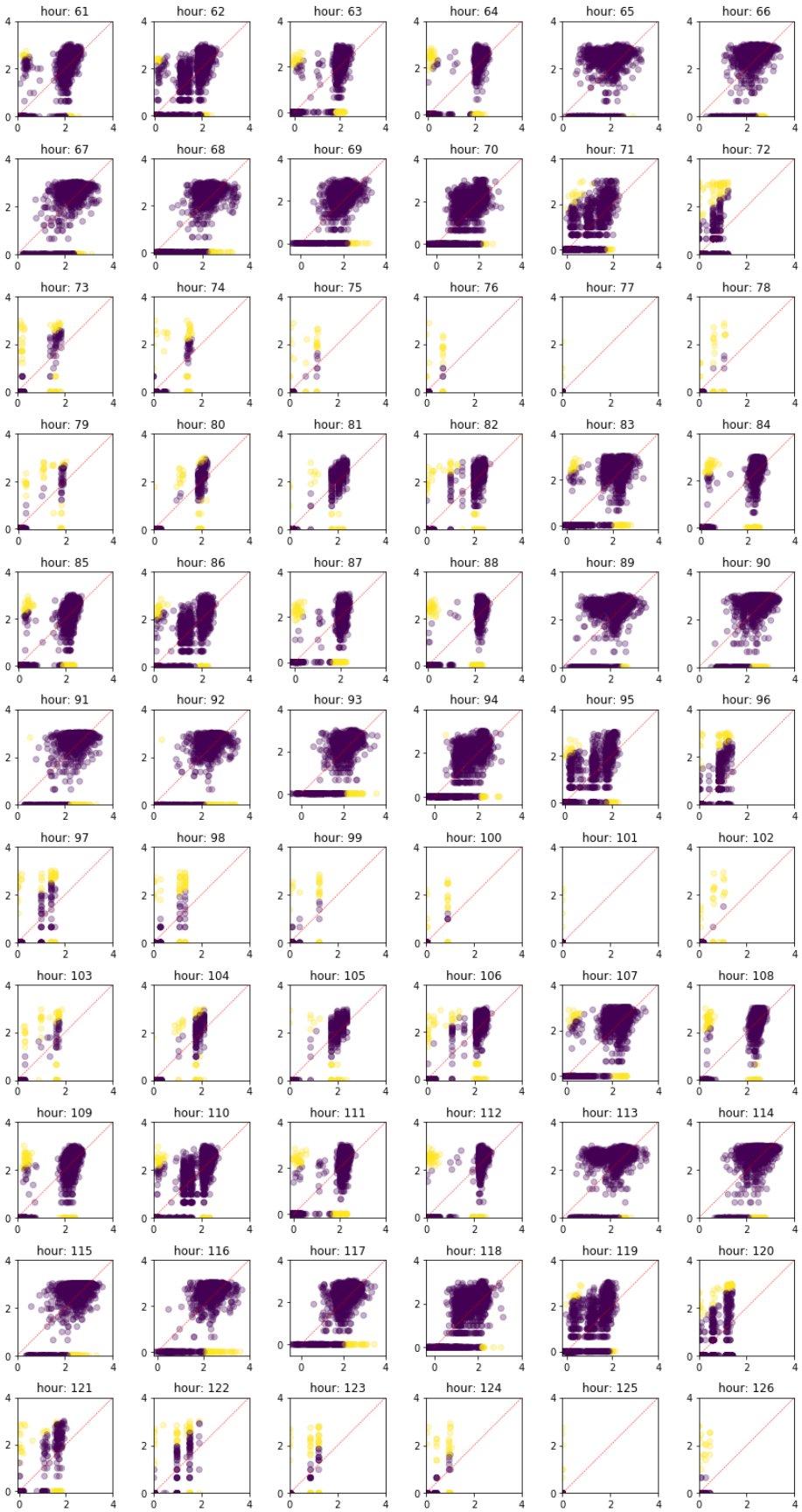

Figure 62: Multiple linear regression with lasso, Box-Cox transformation, fitted vs true values (part 2).

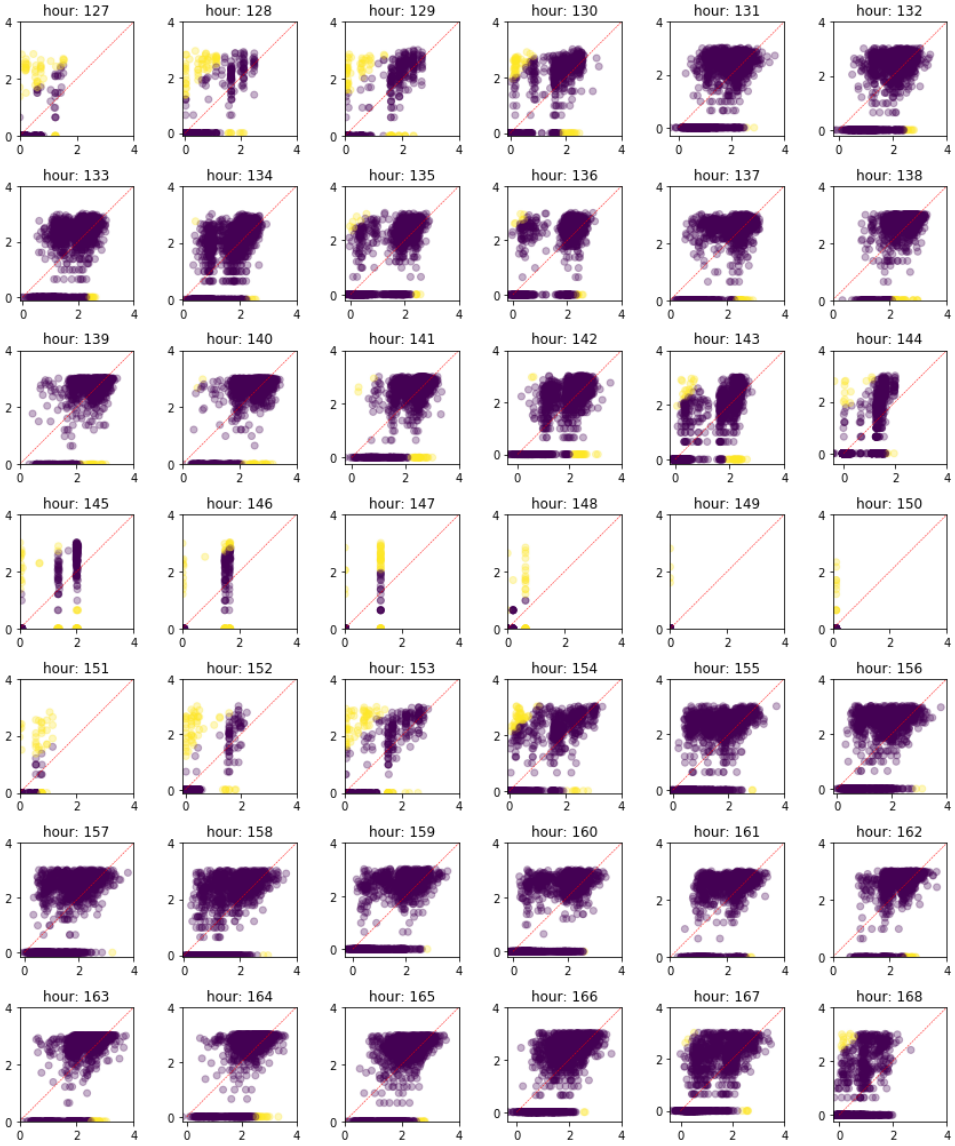

Figure 63: Multiple linear regression with lasso, Box-Cox transformation, fitted vs true values (part 3).

Gradient boosted regression (no transformation)

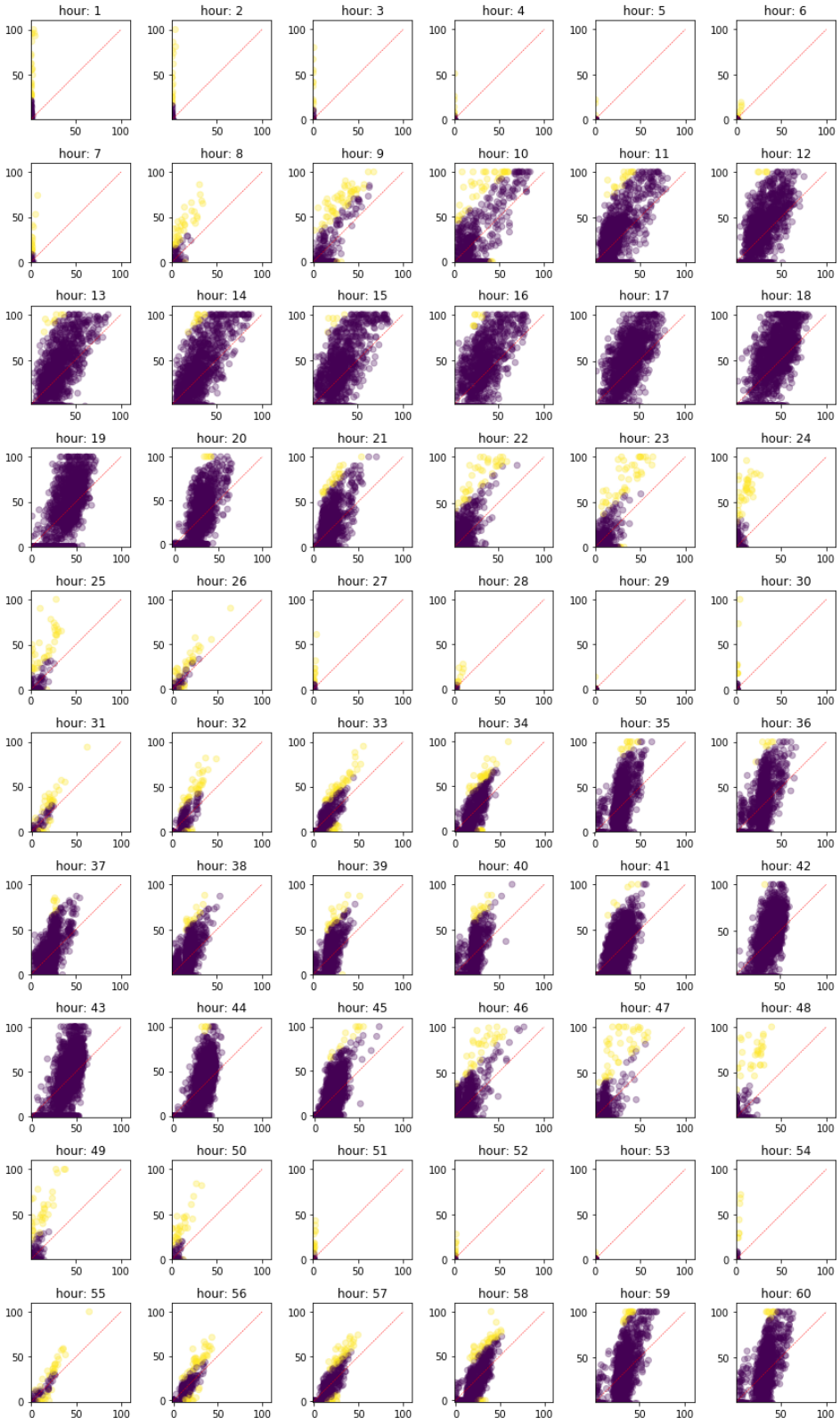

Figure 64: Gradient boosted regression, no transformation, fitted vs true values (part 1).

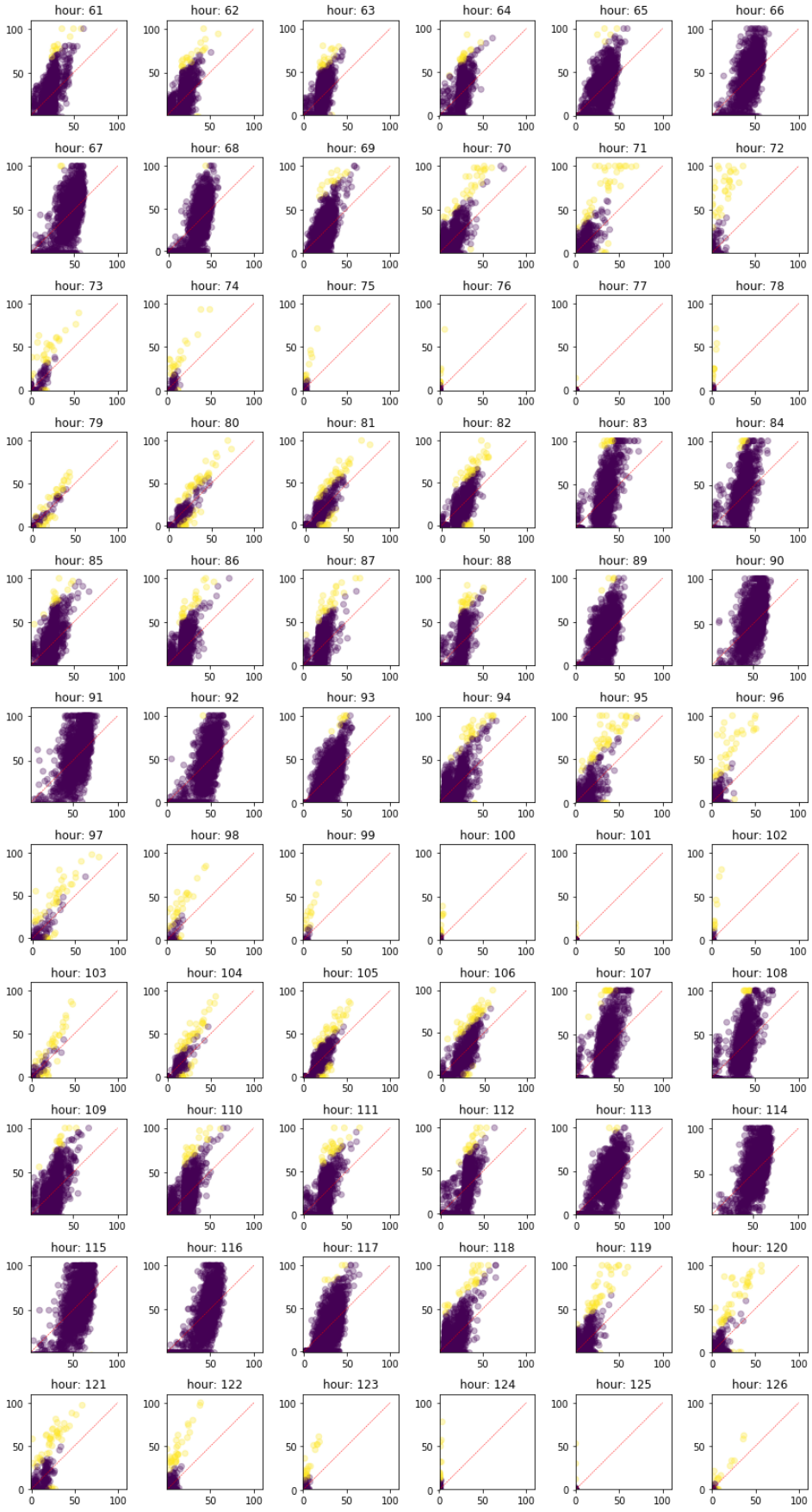

Figure 65: Gradient boosted regression, no transformation, fitted vs true values (part 2).

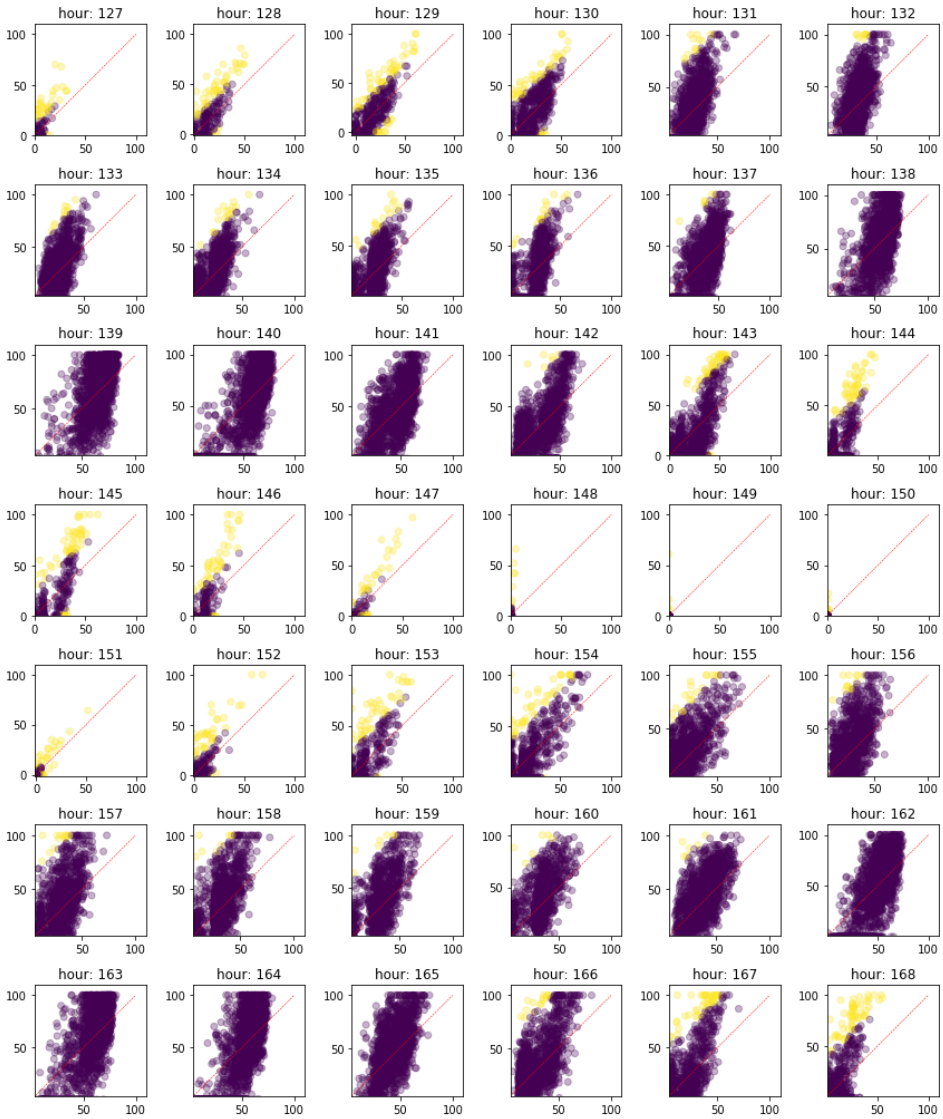

Figure 66: Gradient boosted regression, no transformation, fitted vs true values (part 3).

Gradient boosted regression (logarithm transformation)

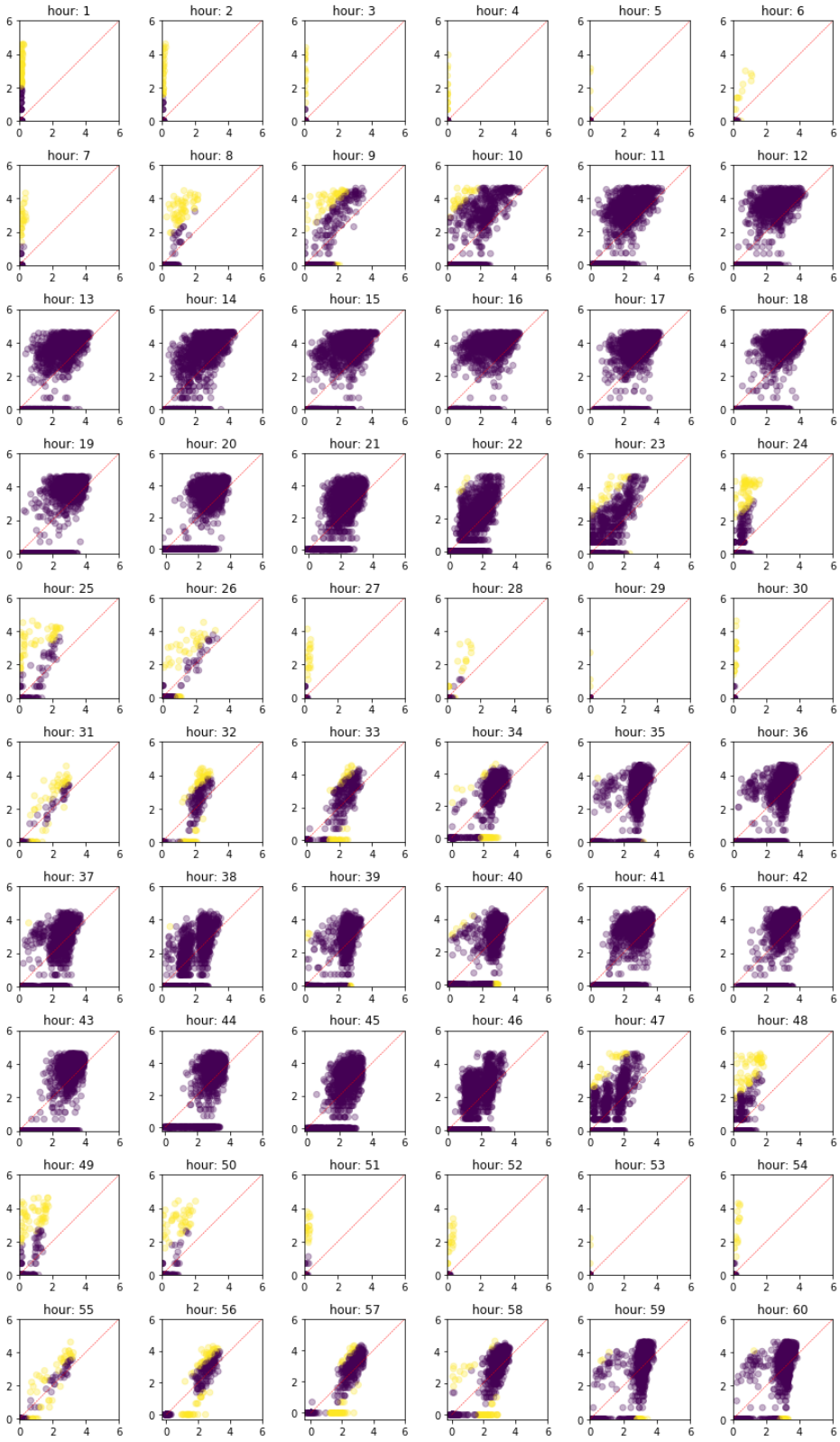

Figure 67: Gradient boosted regression, logarithm transformation, fitted vs true values (part 1).

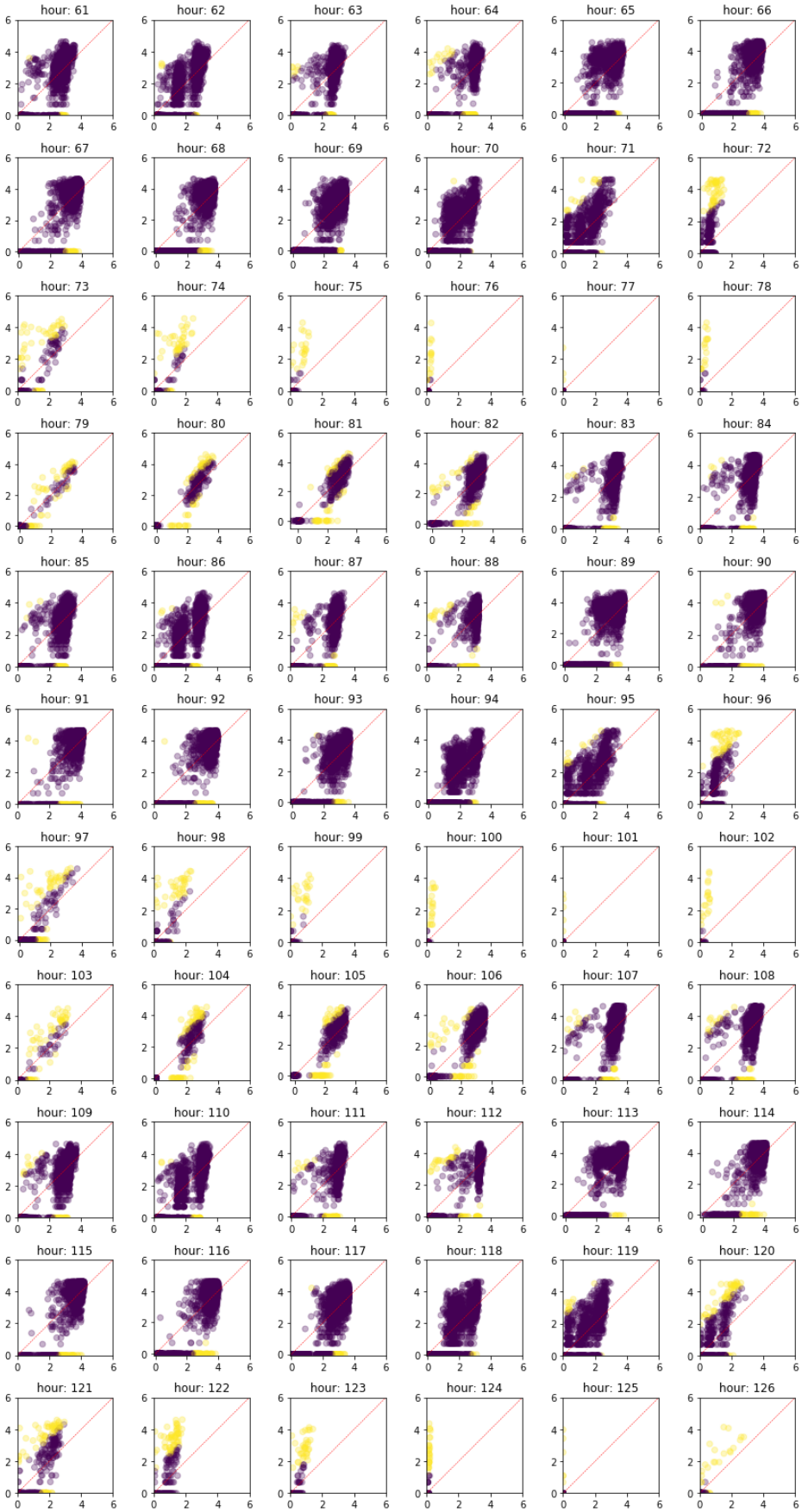

Figure 68: Gradient boosted regression, logarithm transformation, fitted vs true values (part 2).

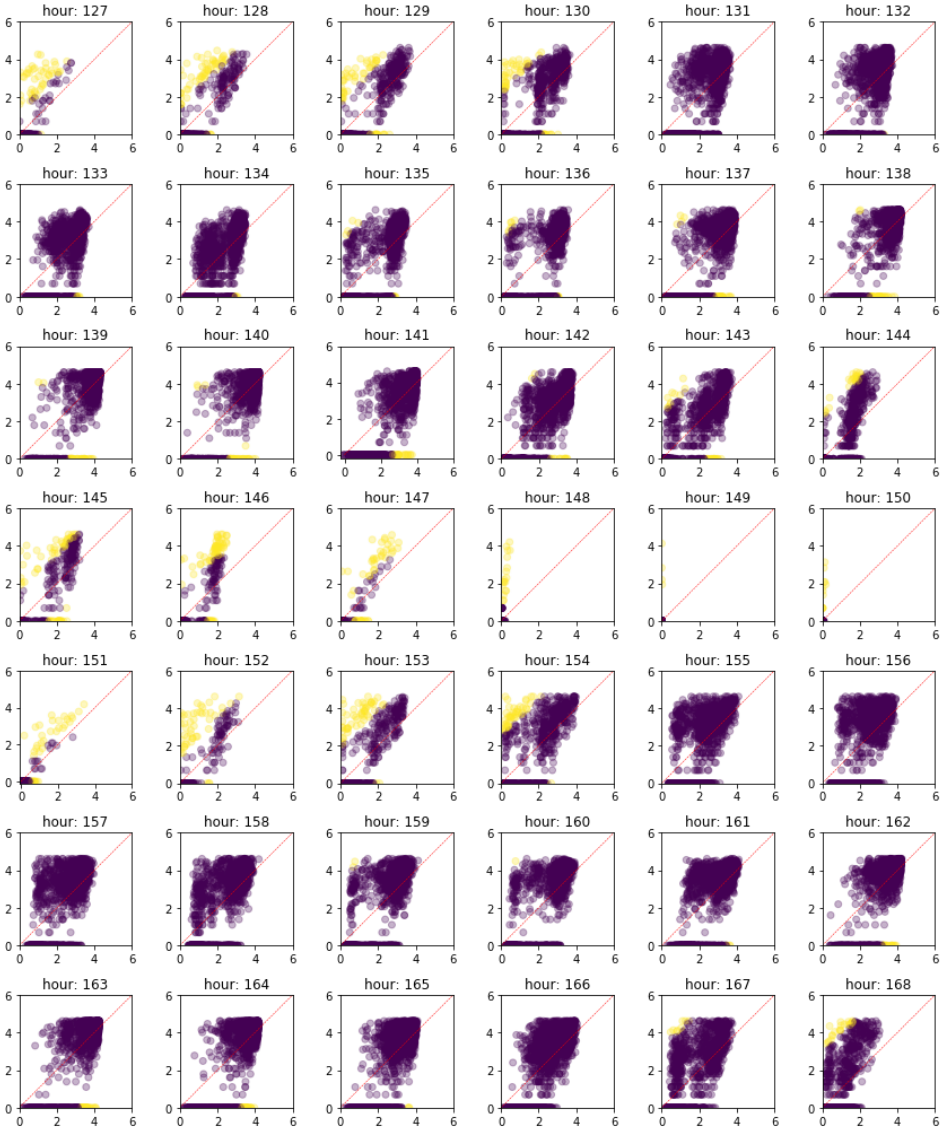

Figure 69: Gradient boosted regression, logarithm transformation, fitted vs true values (part 3).

Gradient boosted regression (Box-Cox transformation)

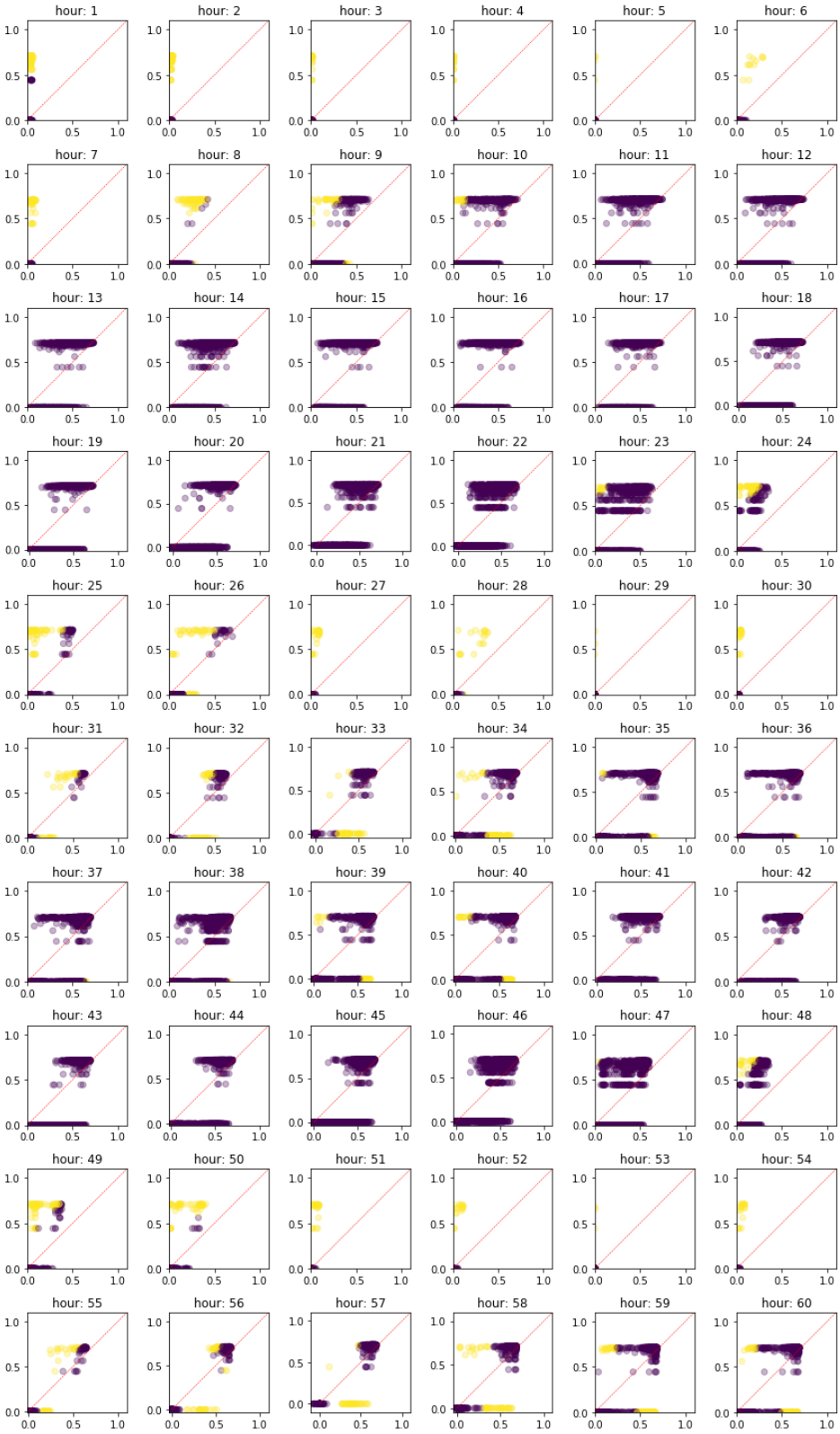

Figure 70: Gradient boosted regression, Box-Cox transformation, fitted vs true values (part 1).

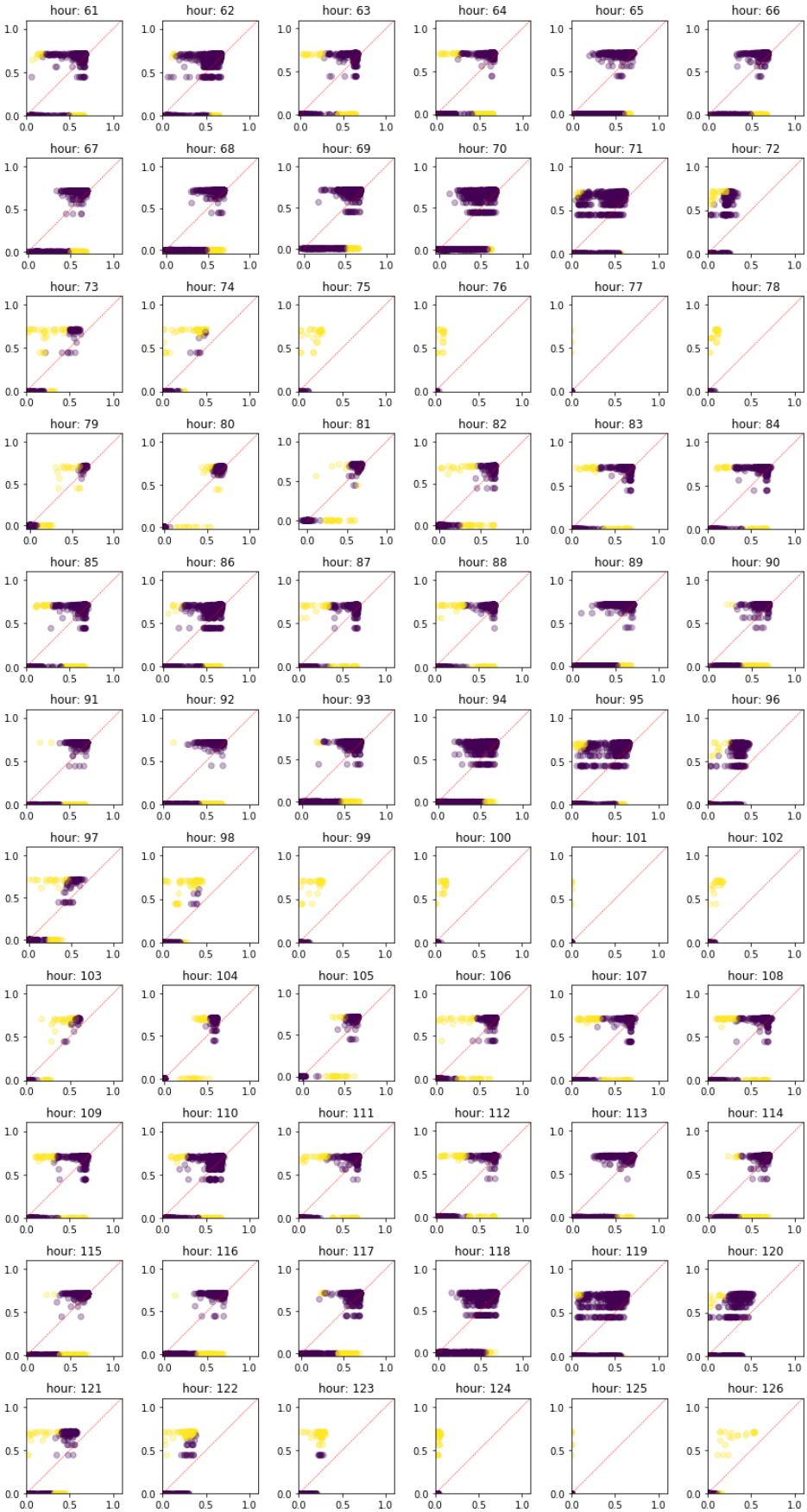

Figure 71: Gradient boosted regression, Box-Cox transformation, fitted vs true values (part 2).

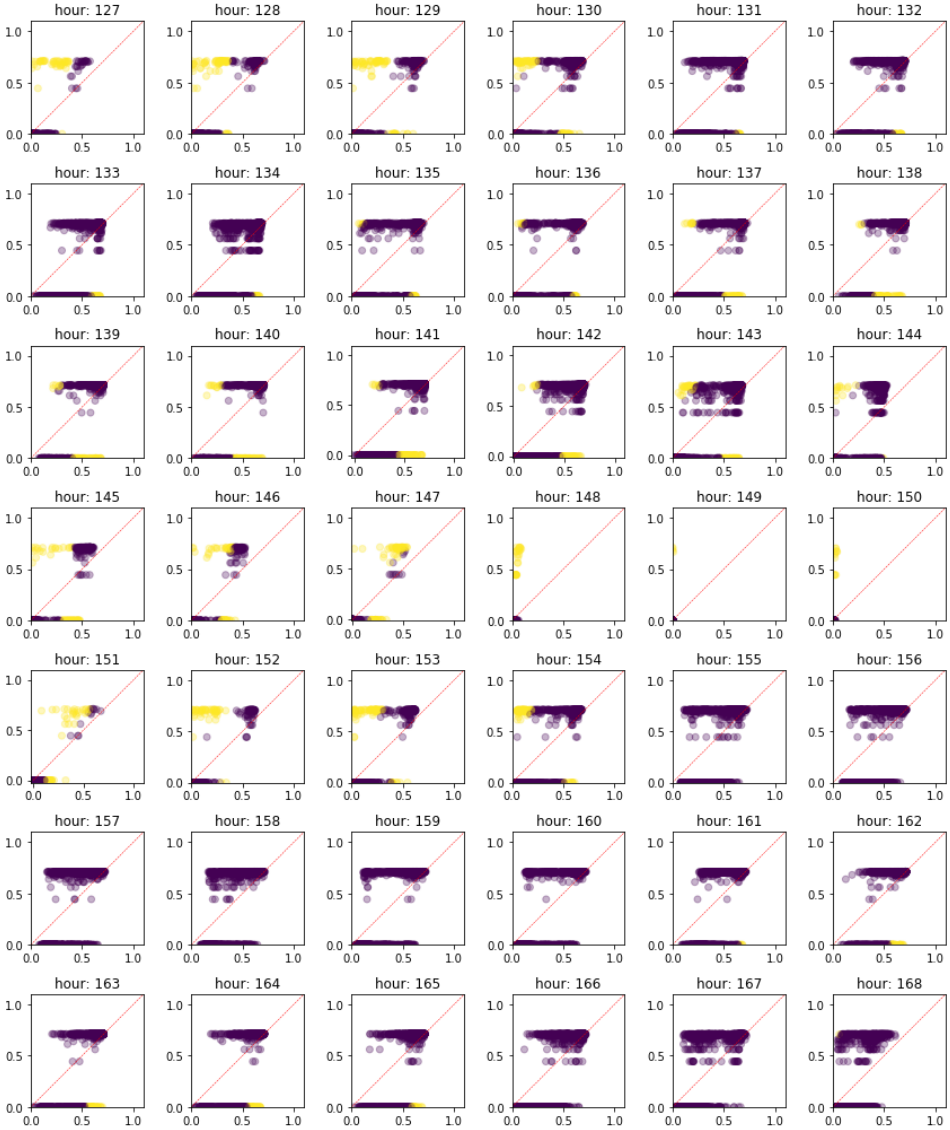

Figure 72: Gradient boosted regression, Box-Cox transformation, fitted vs true values (part 3).

8.2. Fitted values versus Residuals

Multiple linear regression with lasso regularization (residuals, no transformation)

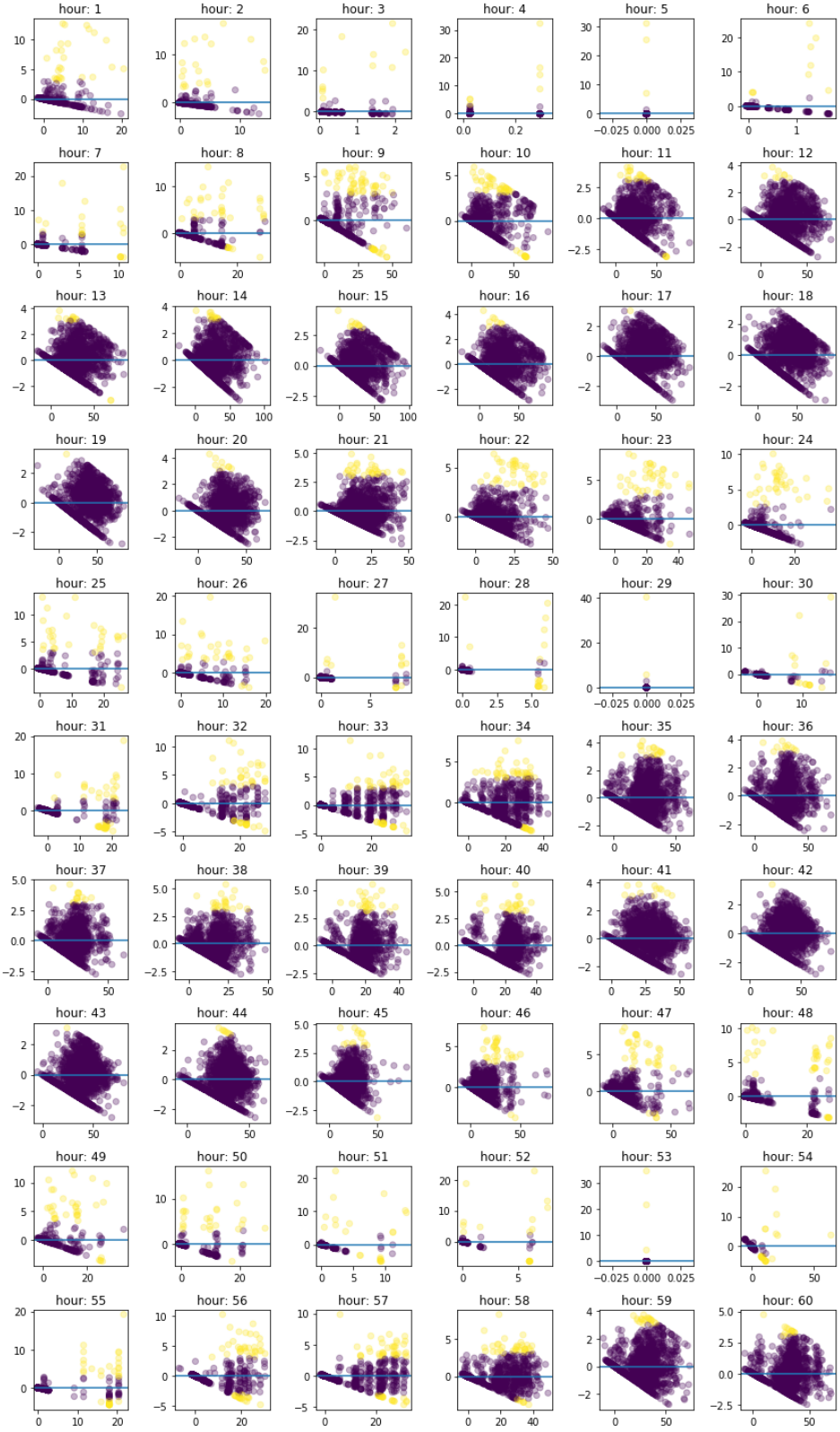

Figure 73: Multiple linear regression with lasso, no transformation, fitted values vs residuals (part 1).

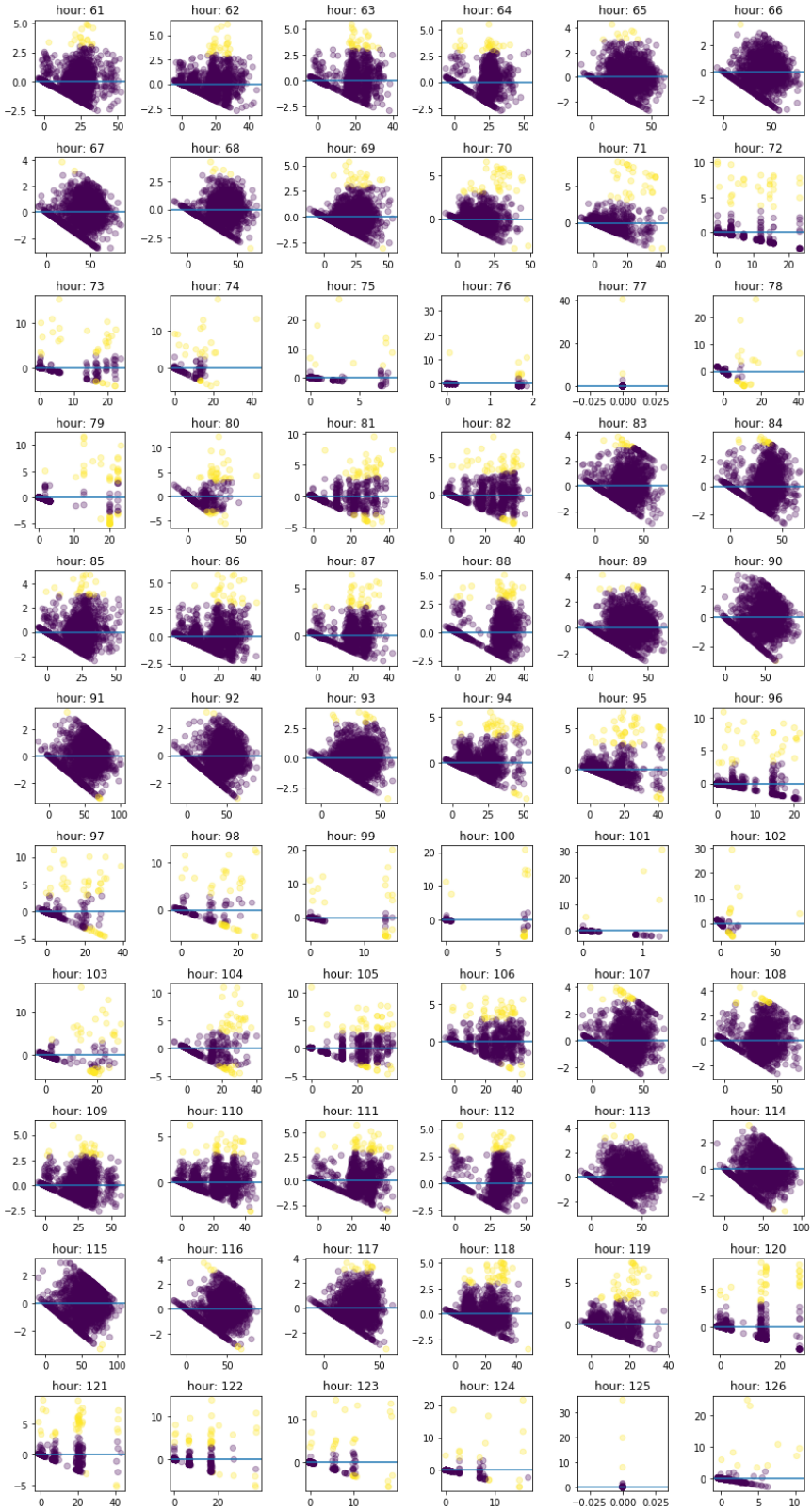

Figure 74: Multiple linear regression with lasso, no transformation, fitted values vs residuals (part 2).

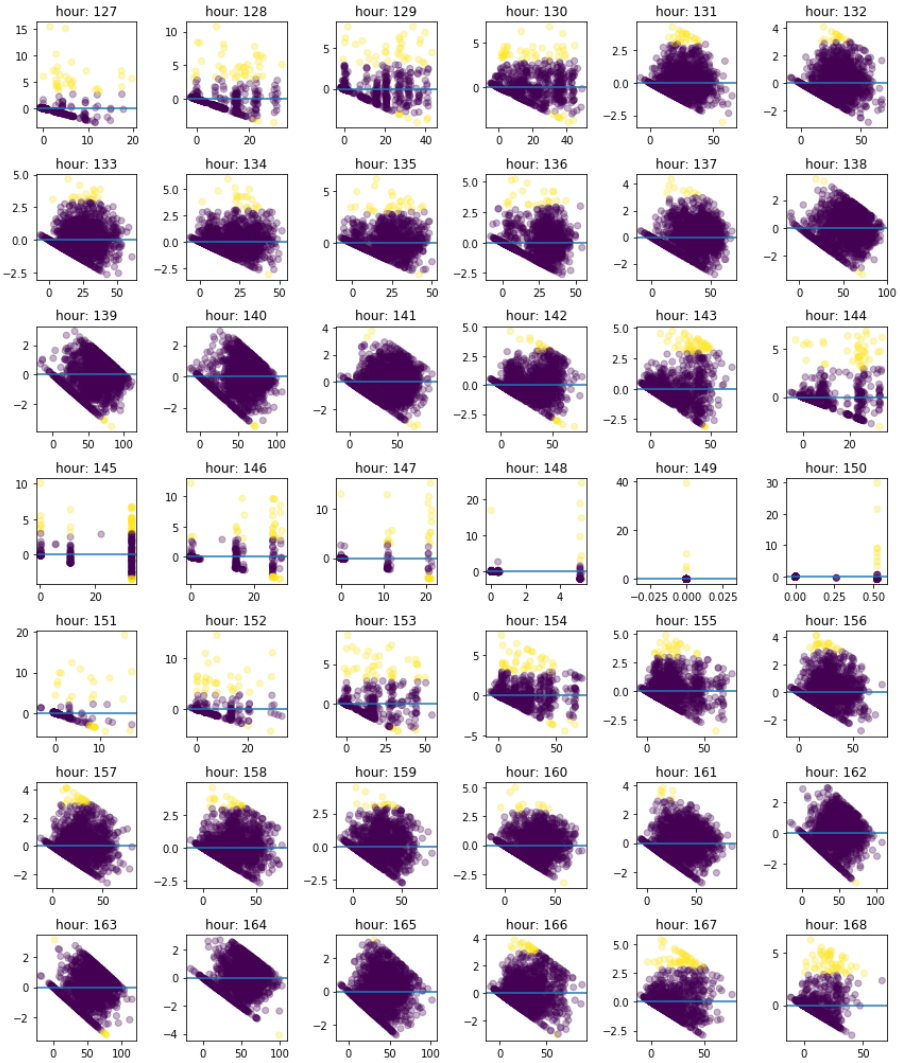

Figure 75: Multiple linear regression with lasso, no transformation, fitted values vs residuals (part 3).

Multiple linear regression with lasso regularization (residuals, logarithm transformation)

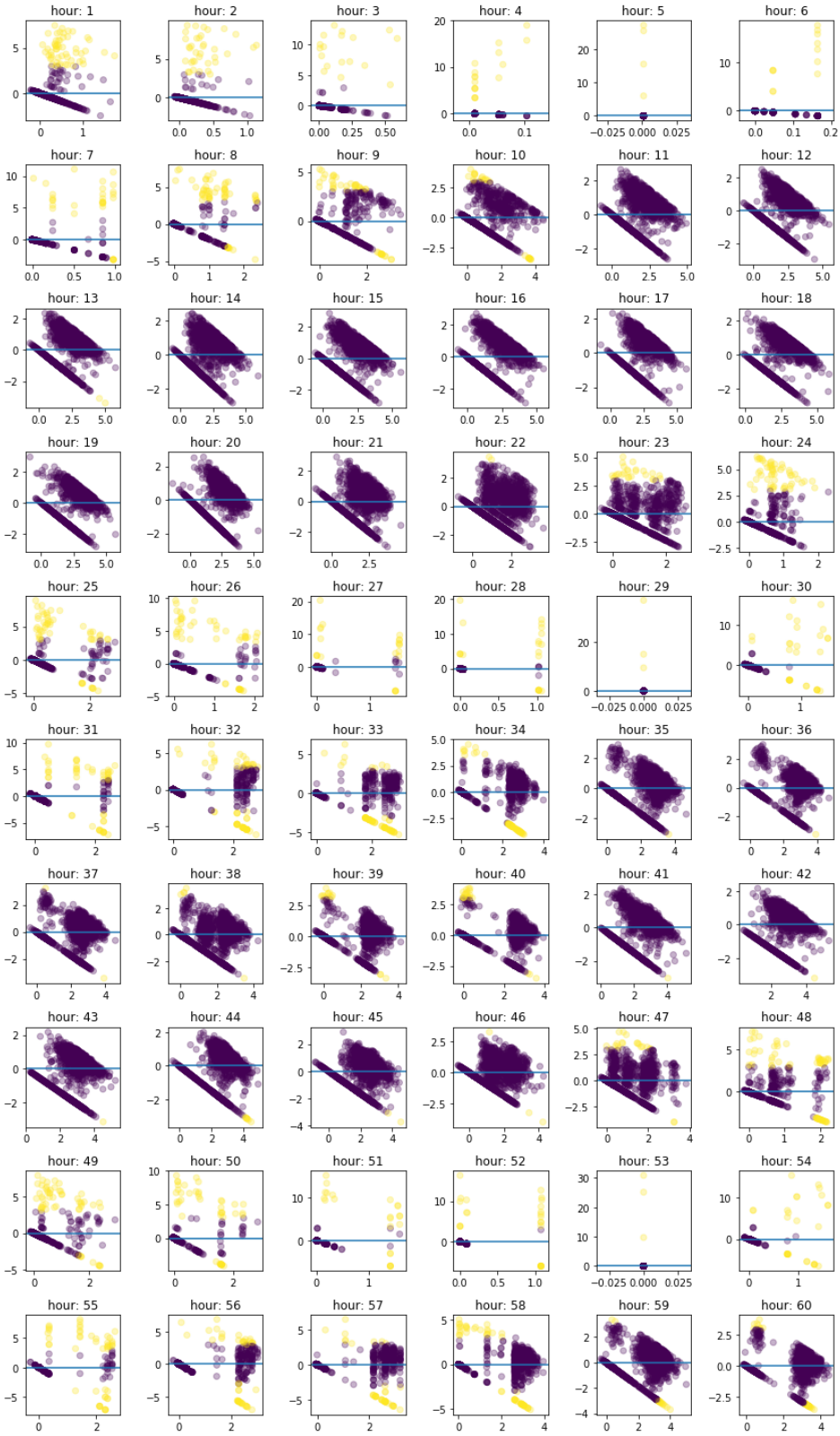

Figure 76: Multiple linear regression with lasso, logarithm transformation, fitted values vs residuals (part 1).

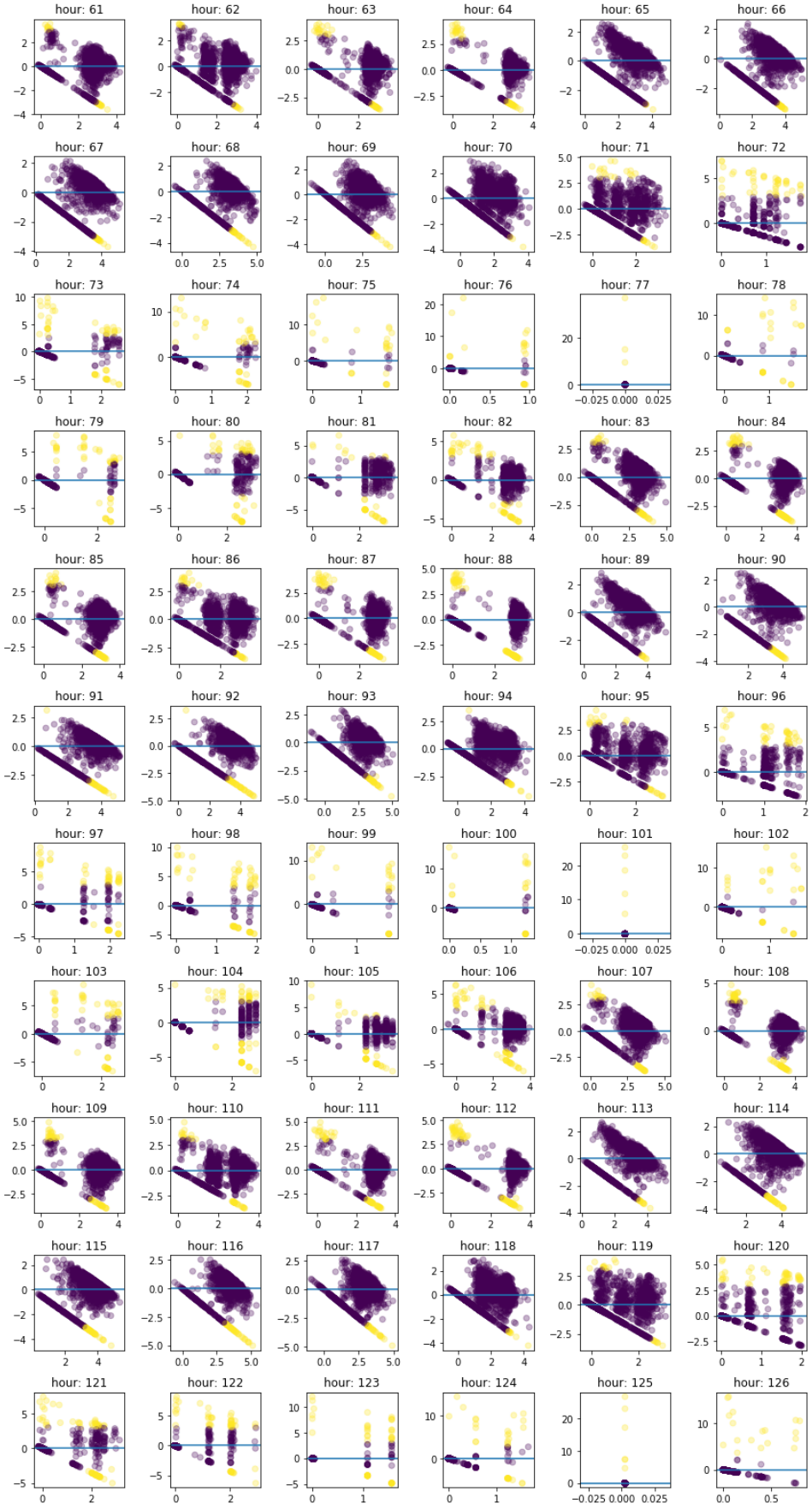

Figure 77: Multiple linear regression with lasso, logarithm transformation, fitted values vs residuals (part 2).

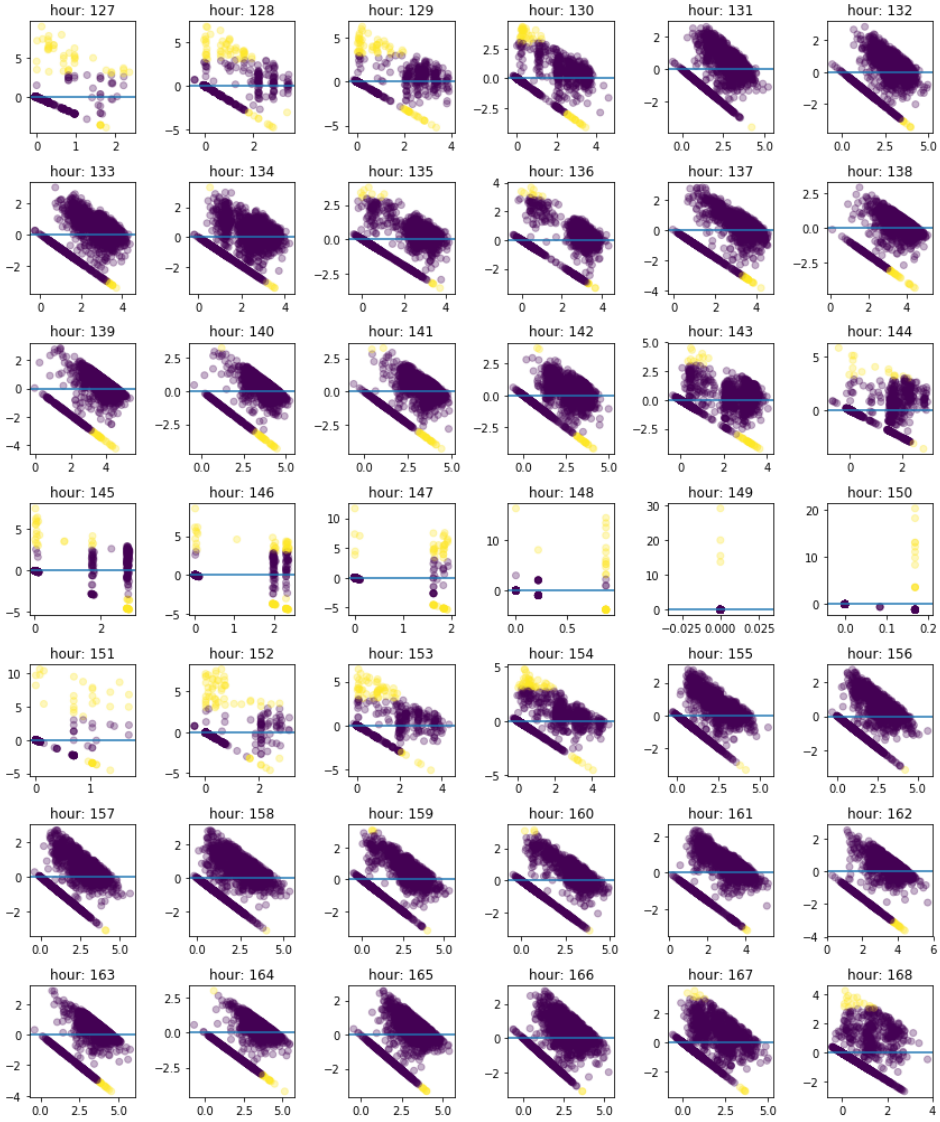

Figure 78: Multiple linear regression with lasso, logarithm transformation, fitted values vs residuals (part 3).

Multiple linear regression with lasso regularization (residuals, Box-Cox transformation)

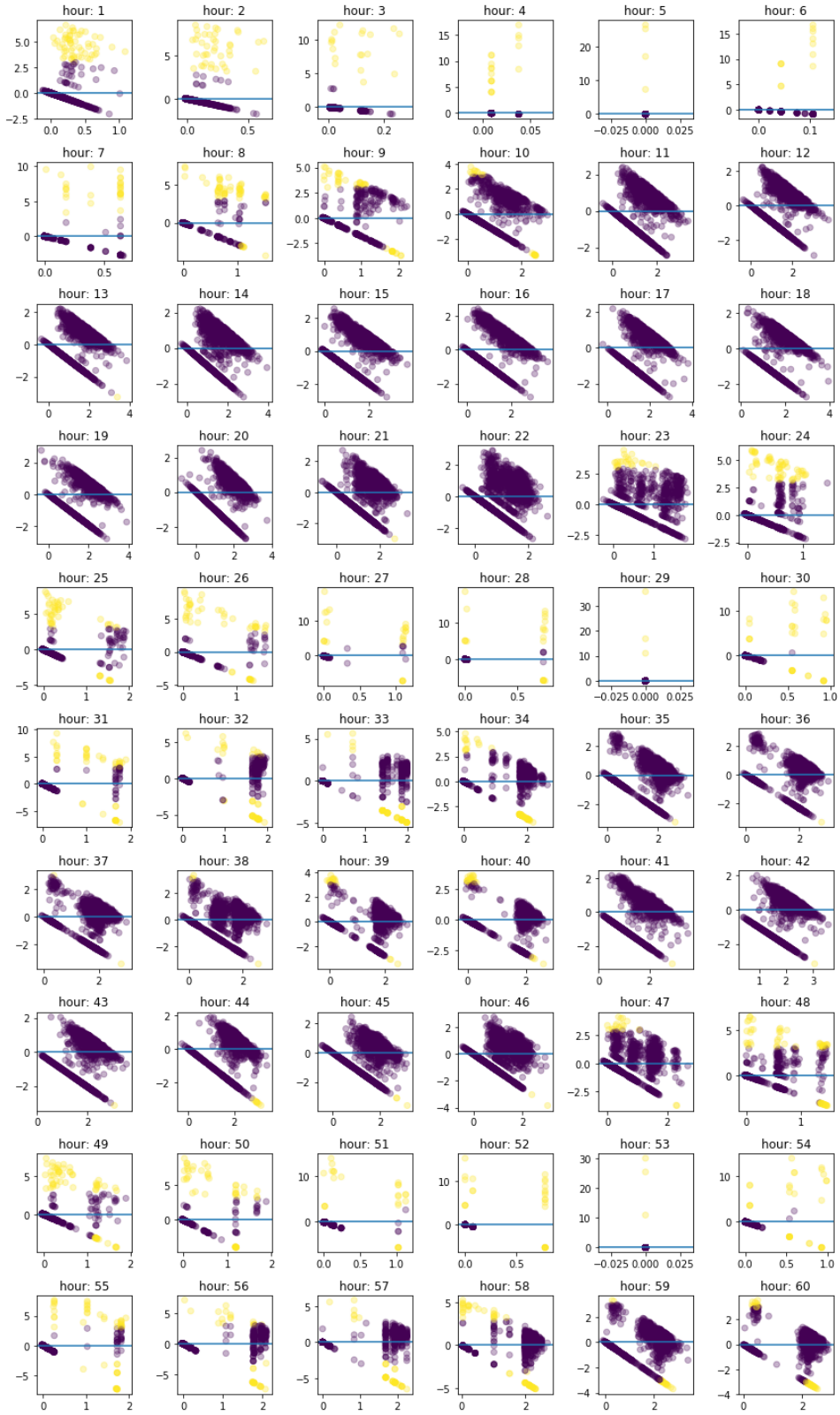

Figure 79: Multiple linear regression with lasso, Box-Cox transformation, fitted values vs residuals (part 1).

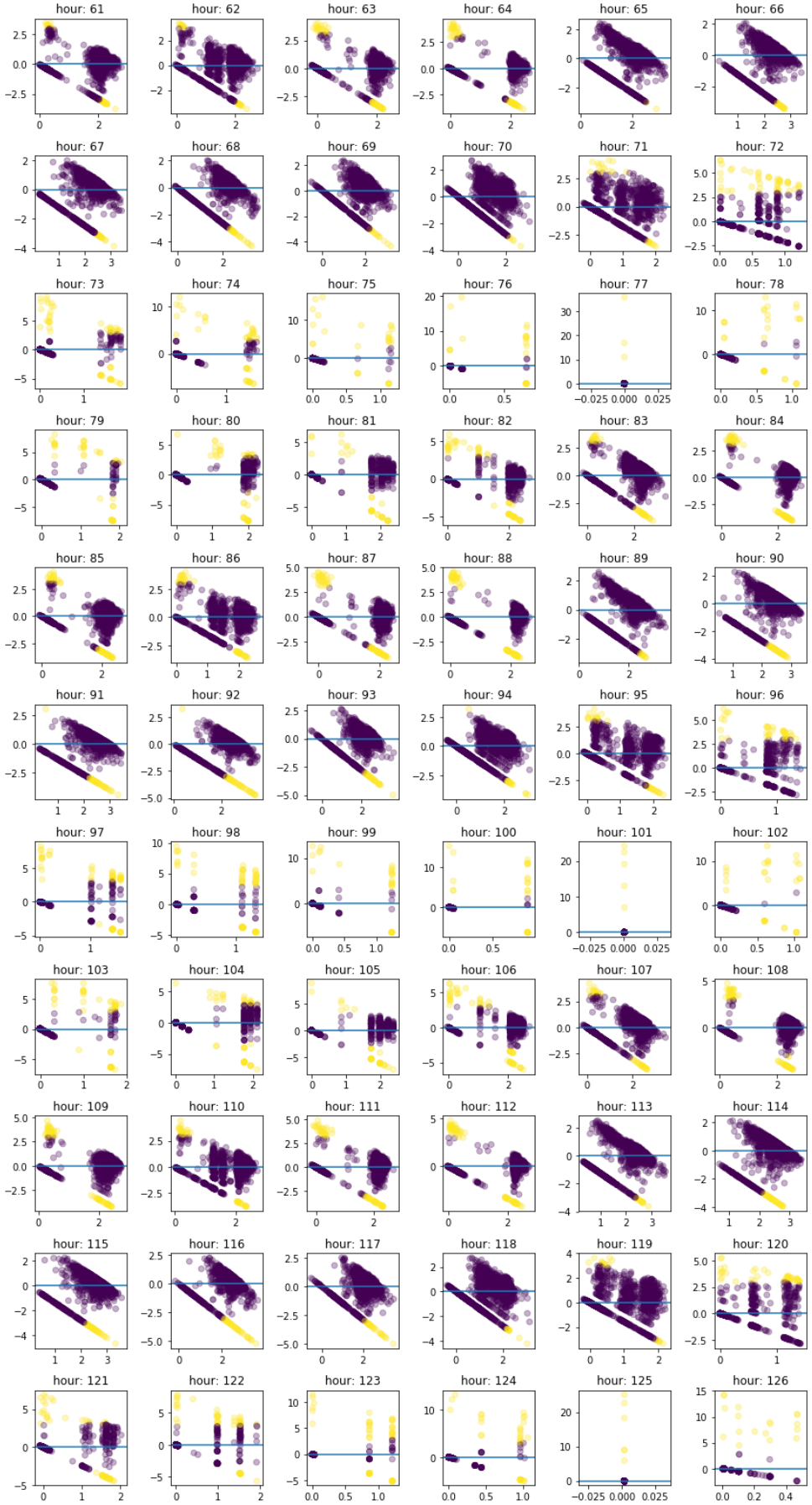

Figure 80: Multiple linear regression with lasso, Box-Cox transformation, fitted values vs residuals (part 2).

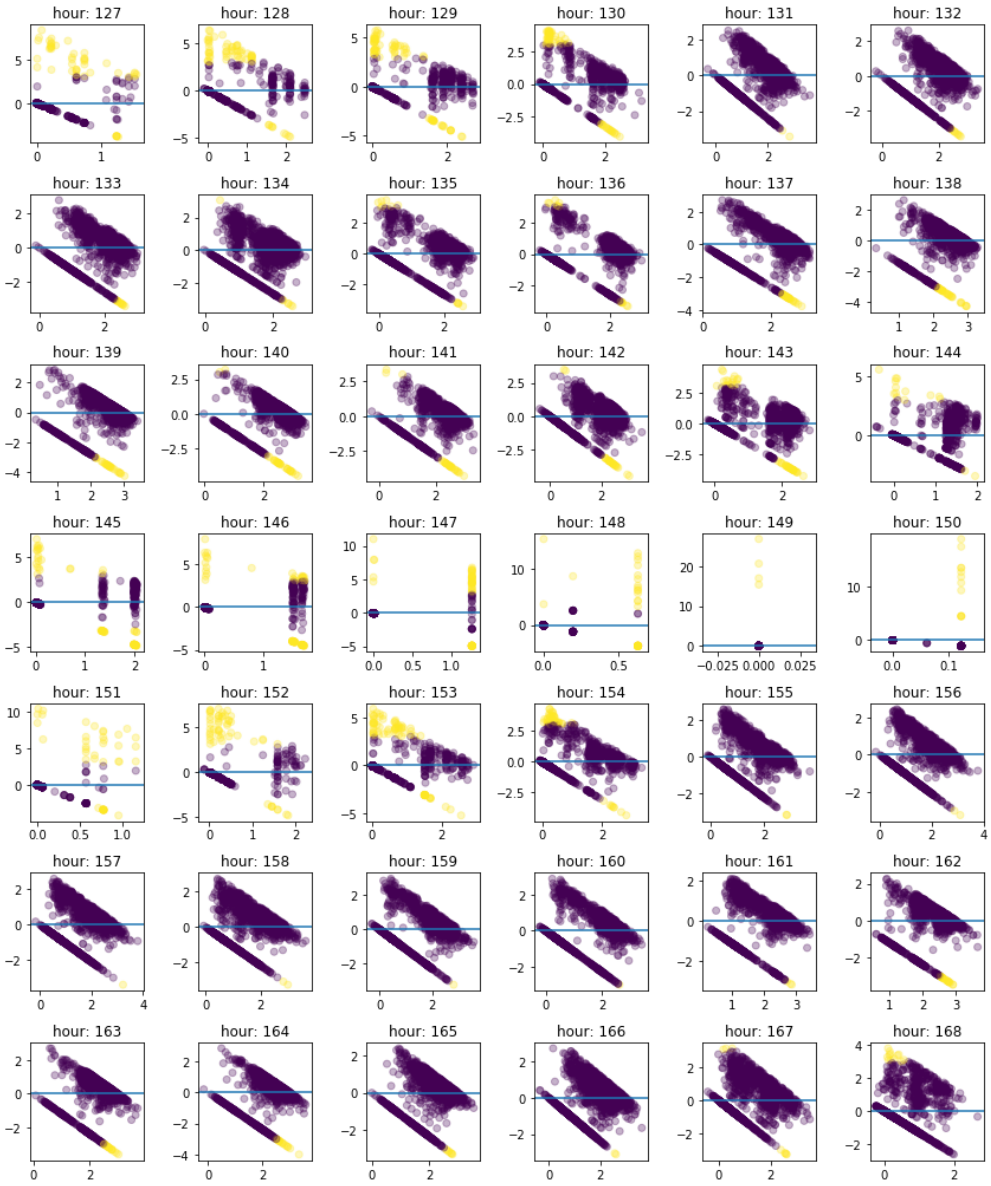

Figure 81: Multiple linear regression with lasso, Box-Cox transformation, fitted values vs residuals (part 3).

Gradient boosted regression (residuals, no transformation)

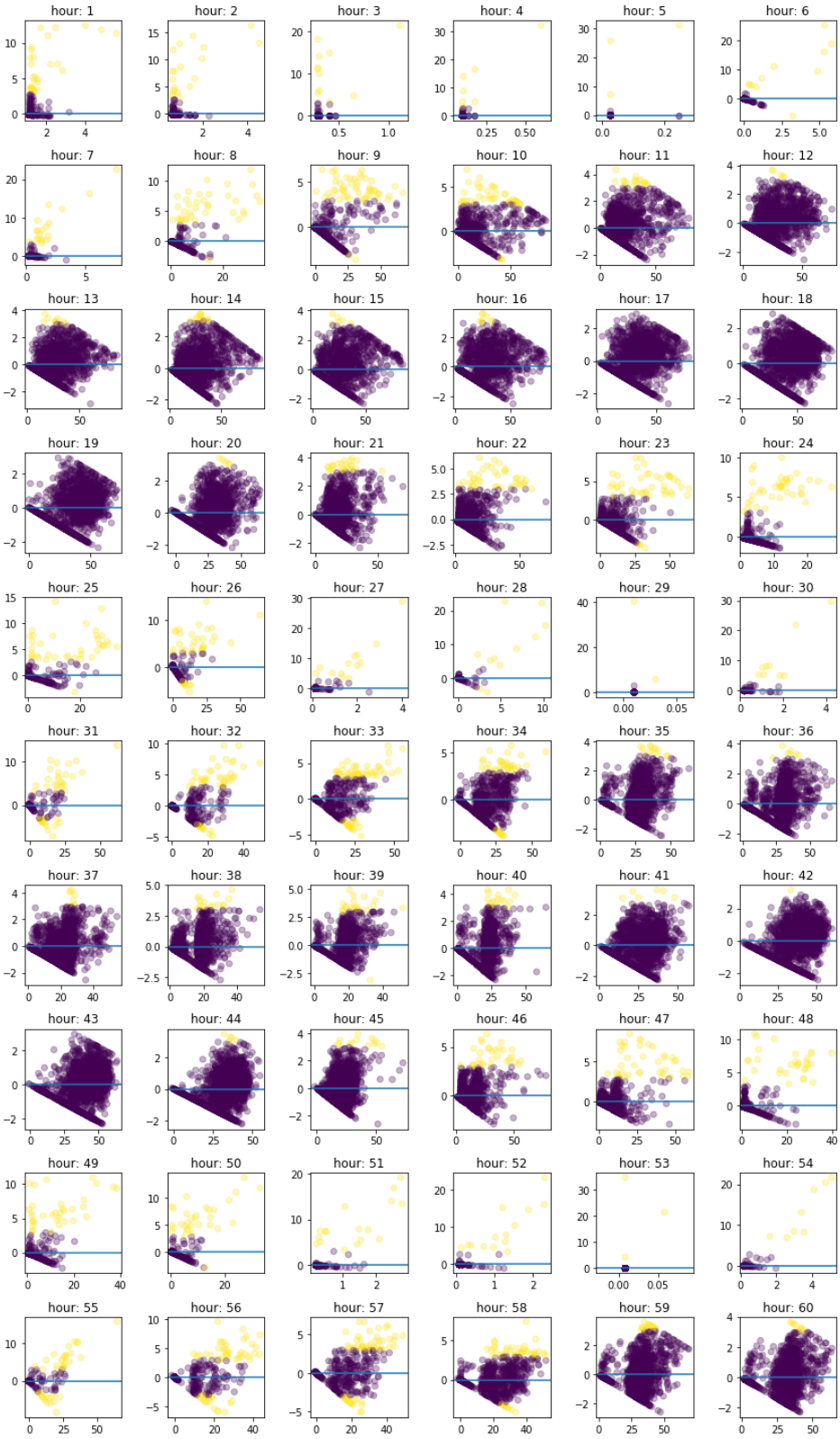

Figure 82: Gradient boosted regression, no transformation, fitted values vs residuals (part 1).

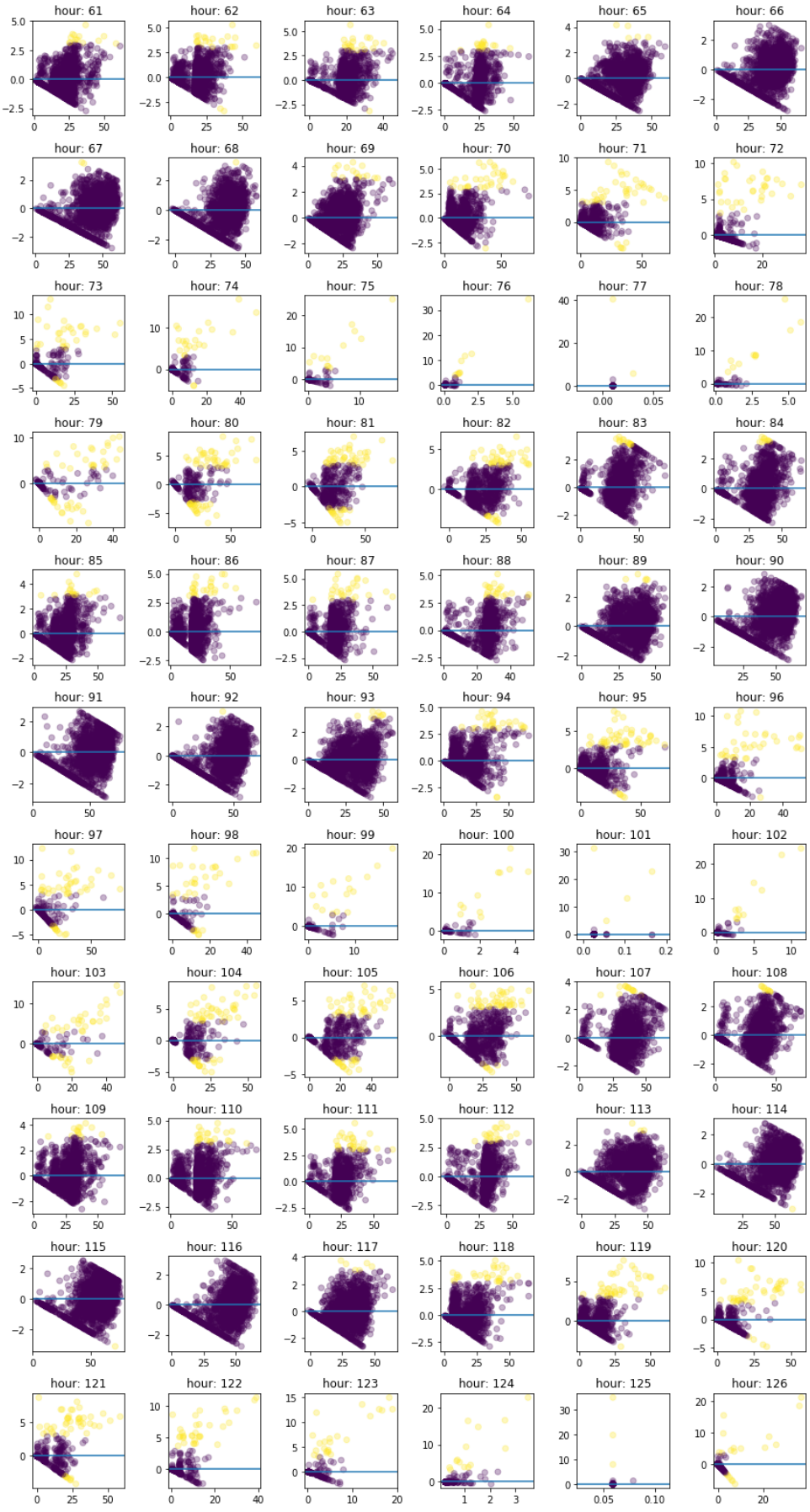

Figure 83: Gradient boosted regression, no transformation, fitted values vs residuals (part 2).

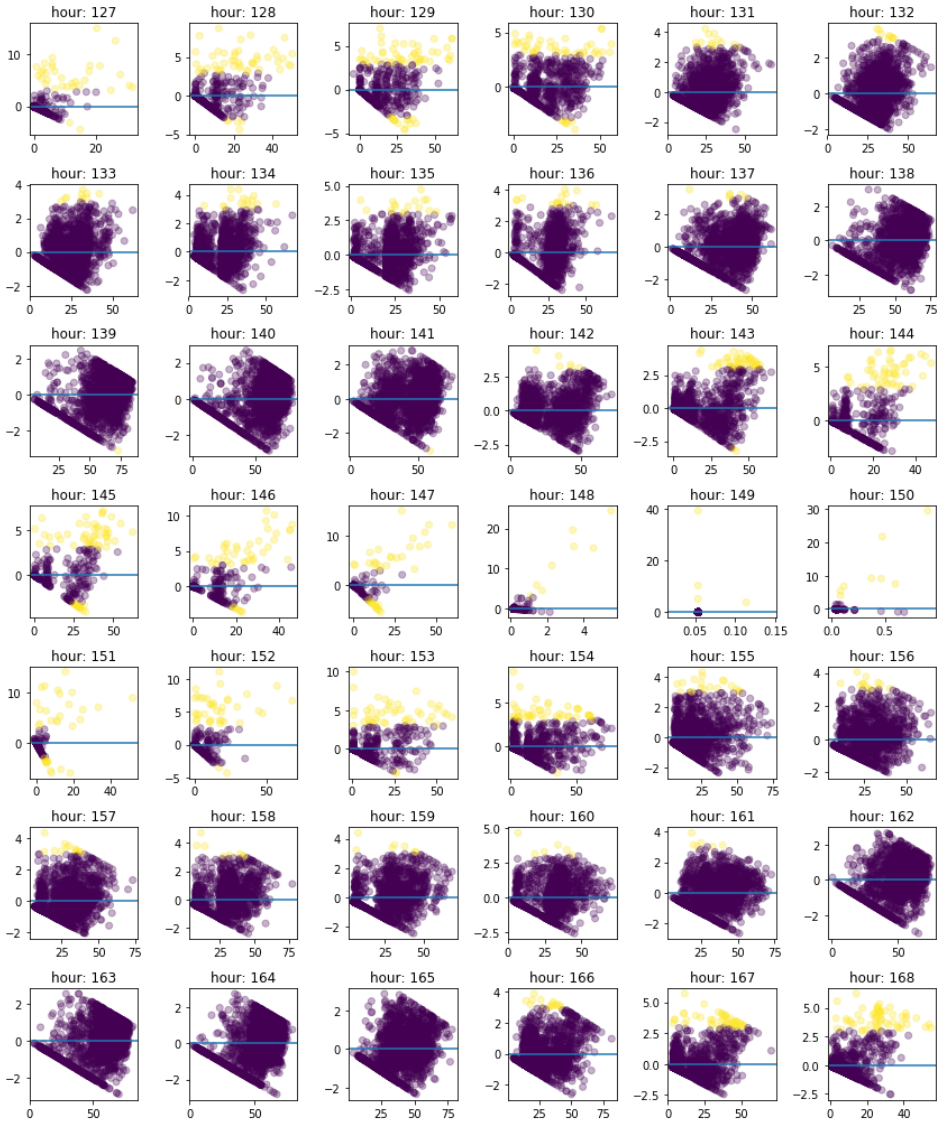

Figure 84: Gradient boosted regression, no transformation, fitted values vs residuals (part 3).

Gradient boosted regression (residuals, logarithm transformation)

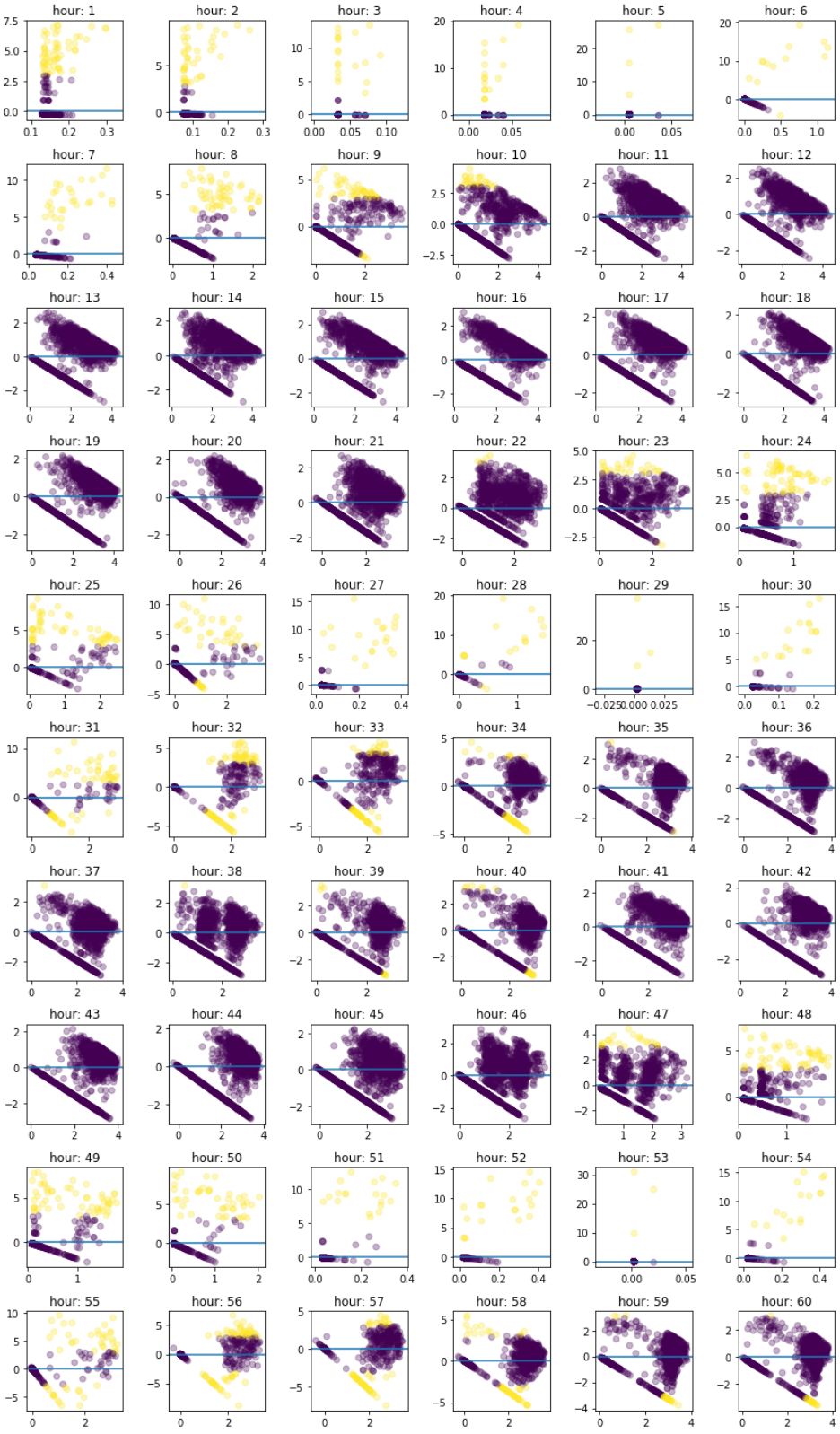

Figure 85: Gradient boosted regression, logarithm transformation, fitted values vs residuals (part 1).

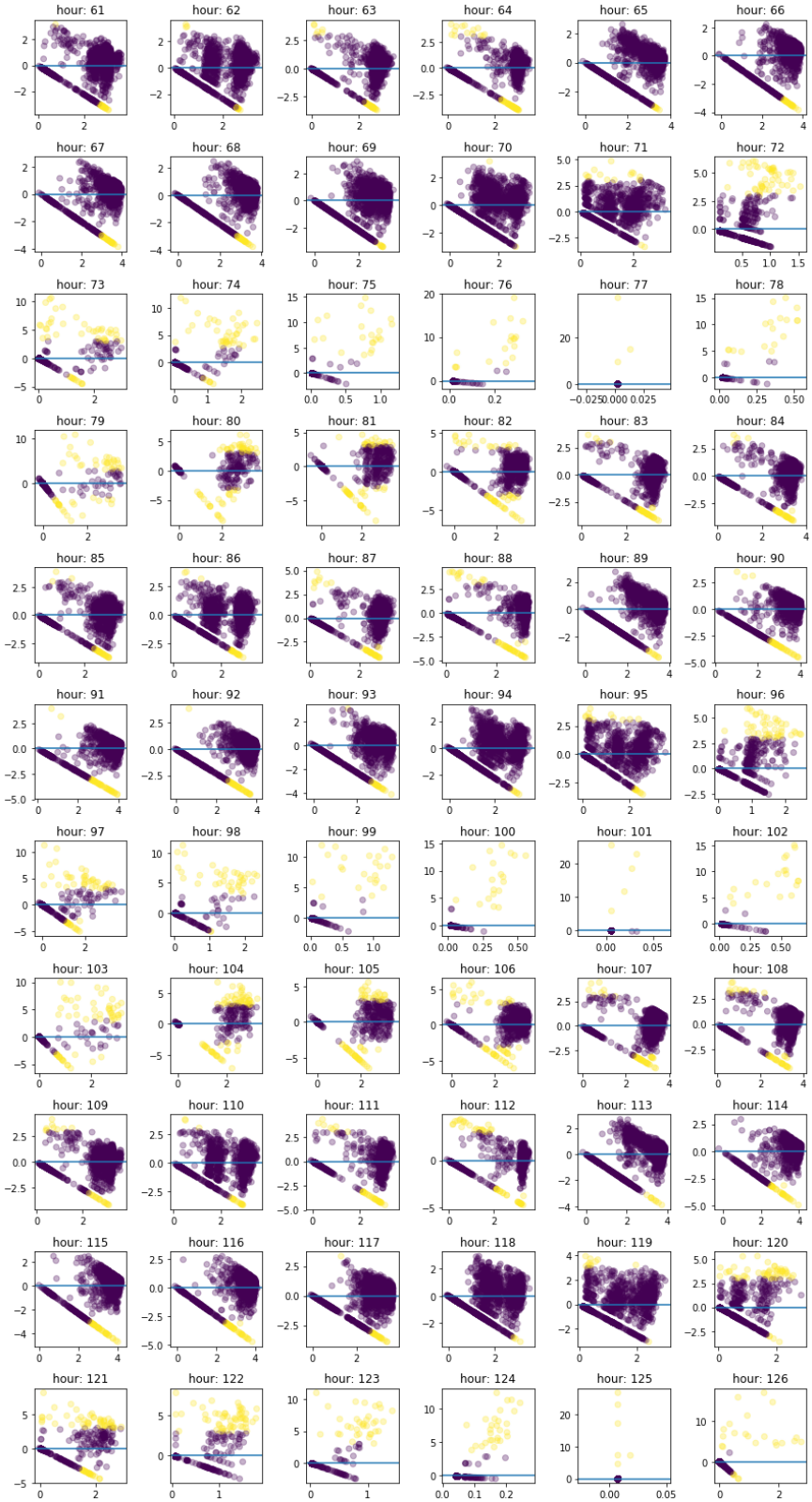

Figure 86: Gradient boosted regression, logarithm transformation, fitted values vs residuals (part 2).

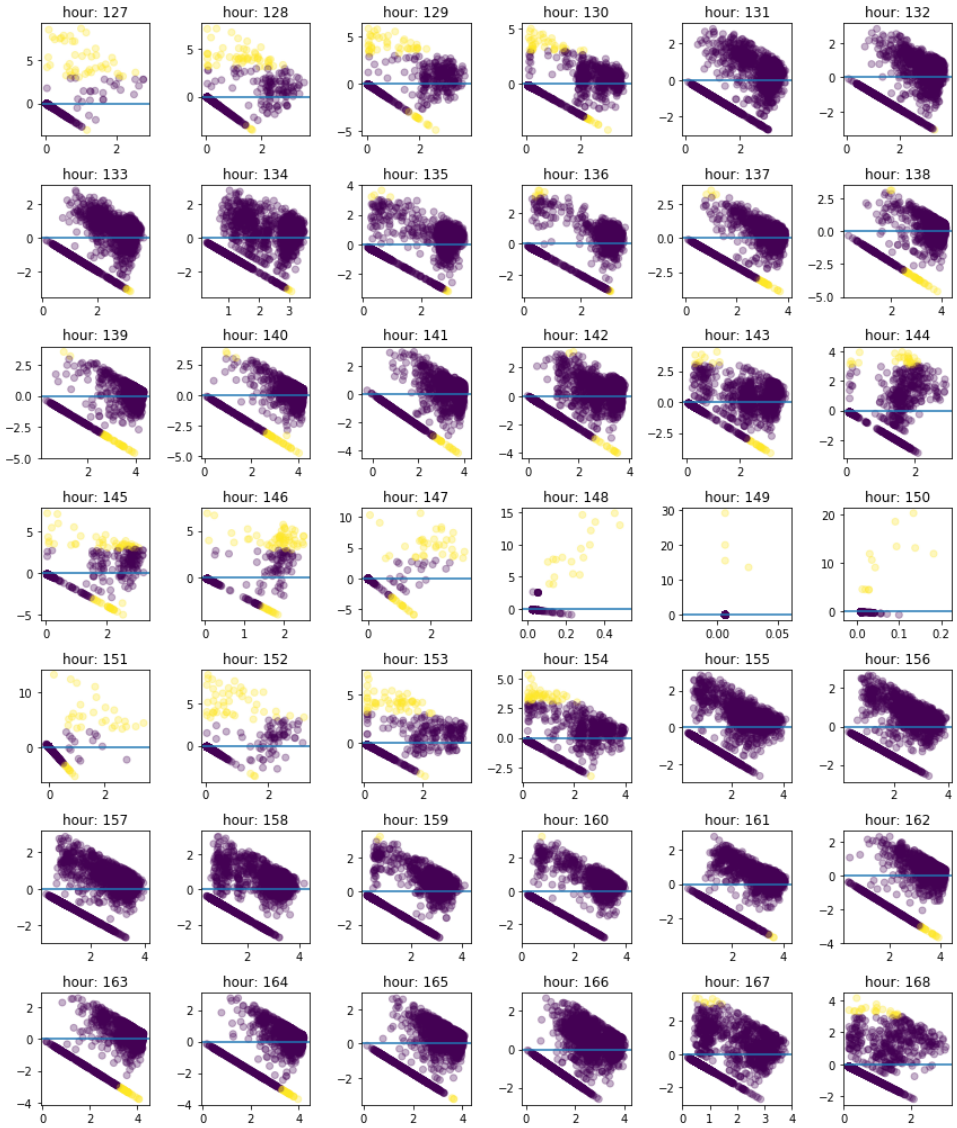

Figure 87: Gradient boosted regression, logarithm transformation, fitted values vs residuals (part 3).

Gradient boosted regression (residuals, Box-Cox transformation)

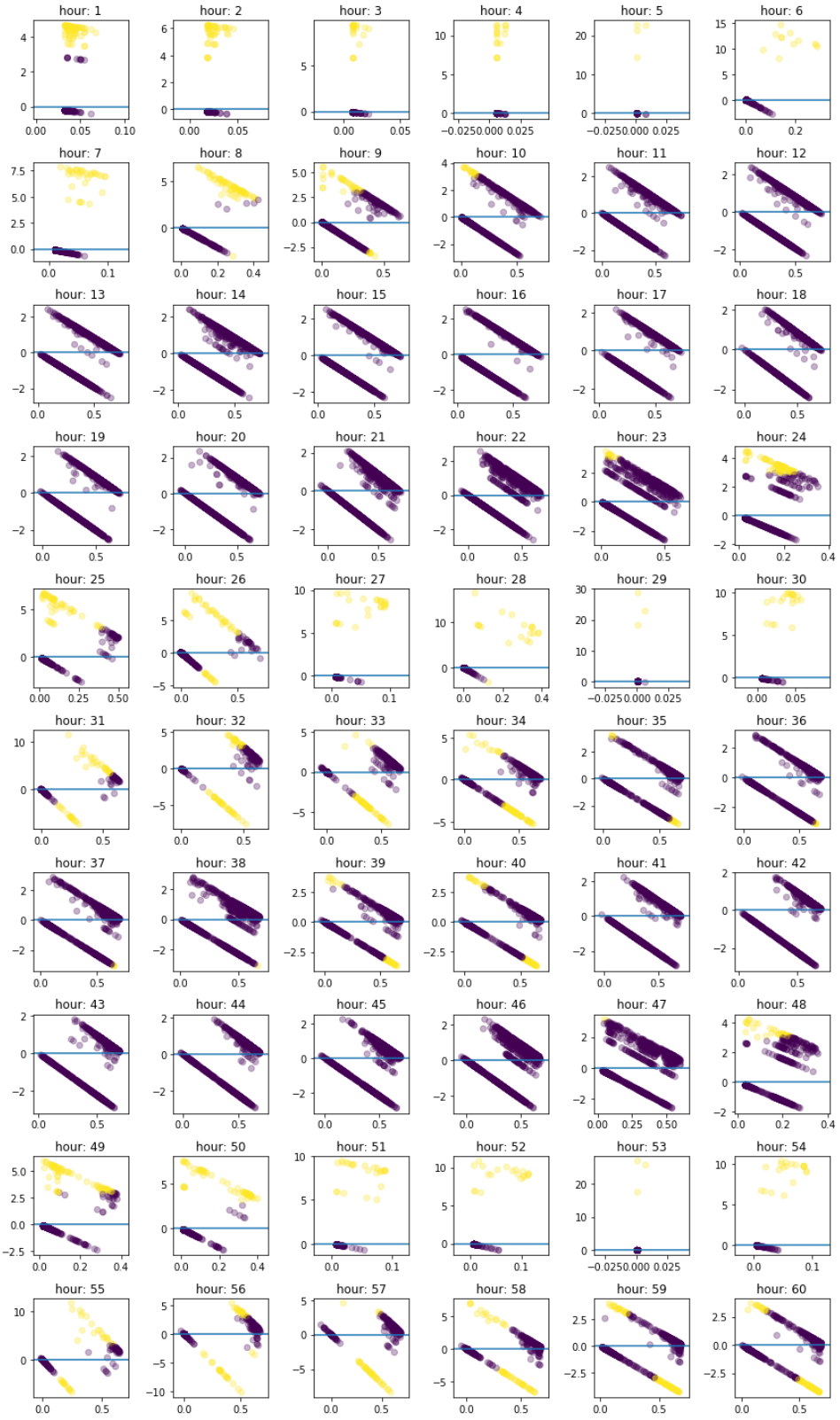

Figure 88: Gradient boosted regression, Box-Cox transformation, fitted values vs residuals (part 1).

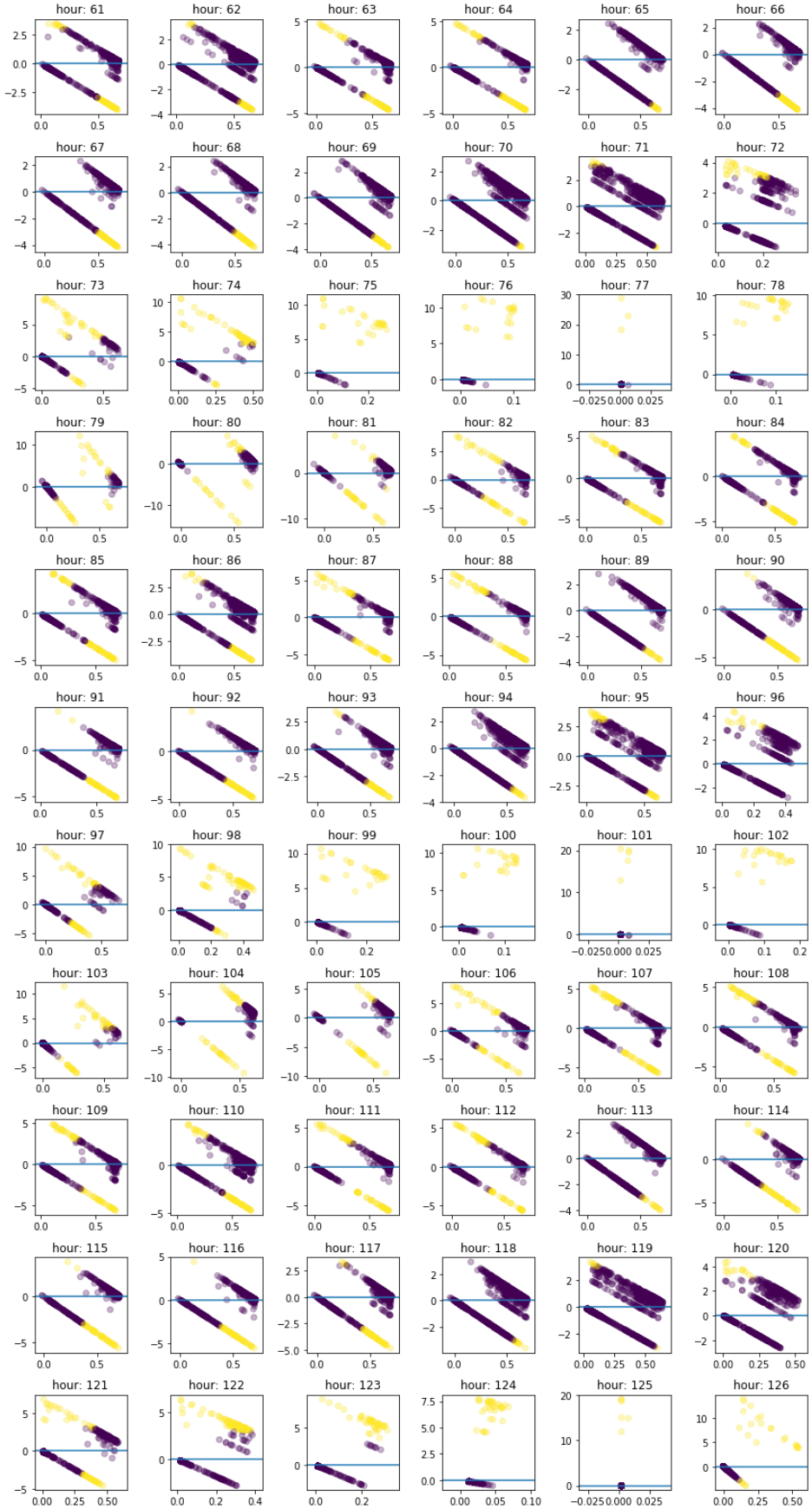

Figure 89: Gradient boosted regression, Box-Cox transformation, fitted values vs residuals (part 2).

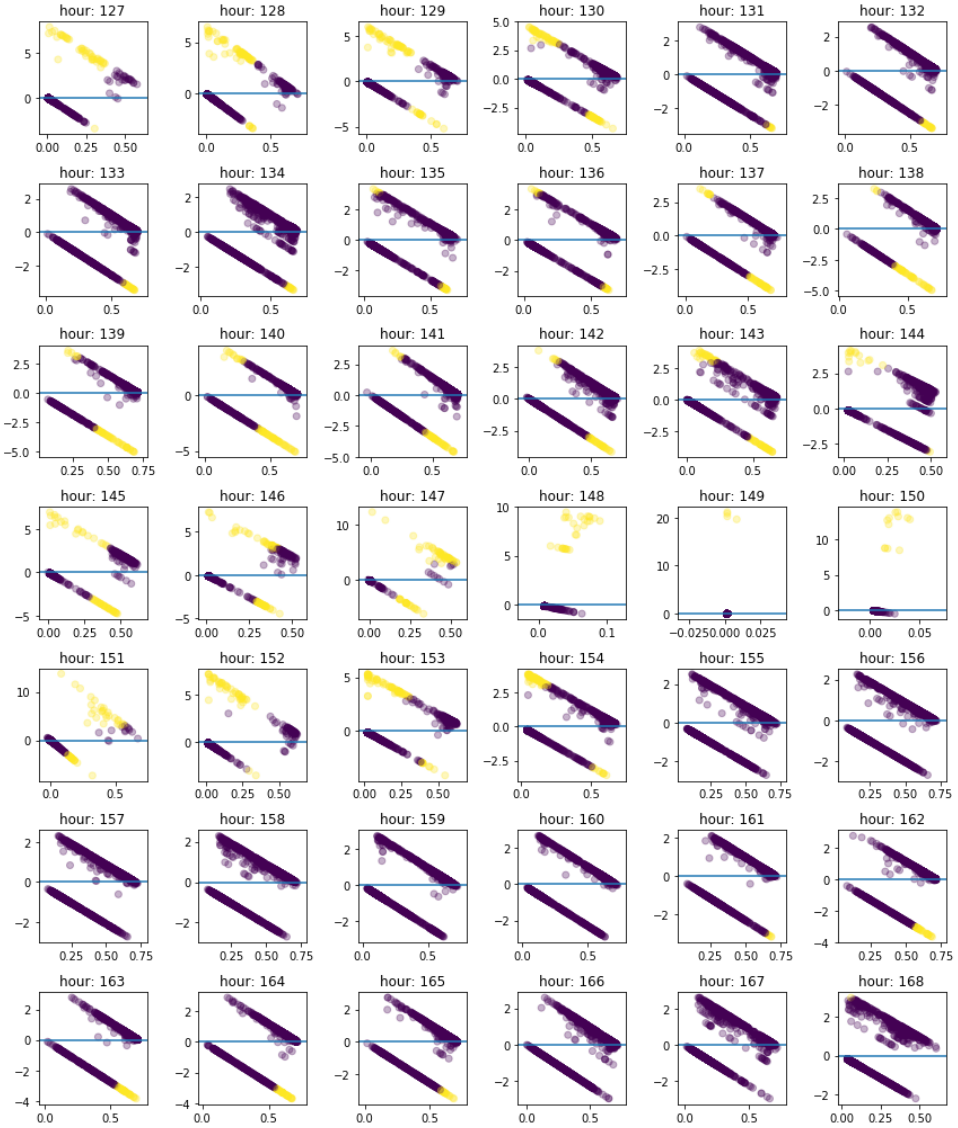

Figure 90: Gradient boosted regression, Box-Cox transformation, fitted values vs residuals (part 3).