Danmarks Tekniske Universitet

Computational tools for data science

Abnormality detection of danish address valuations

Long Ngo
s214606

Shadman Al Islam
s230053

Peter Bonvang
s214658

Md Amin Azad
s230049

Thomas
s211899

November 28, 2023

Contents

1 Abstract

Even small inconsistencies in the public evaluation of properties can greatly impact people's private economy. This study examine the (provisional) Danish property evaluations using two clustering algorithms to find possible abnormalities. Using valuations synthetically altered to be abnormal, the models are evaluated based on the achieved detection rate. Though all the models finds abnormally valuated danish properties, none of them demonstrate a significant performance on the synthetic dataset. Based on these findings, the recommendation is to research more advanced abnormality detection algorithms, ideally regression-based which can learn the relationship between valuation and property attributes.

2 Introduction

Accurate and precise property valuations are pivotal in real estate, influencing numerous financial decisions, including property transactions, taxation, and mortgage lending. In Denmark, the government's persistent attempts to introduce a more accurate valuation model have faced delays, primarily attributed to technical issues and intricate complexities [Ingvorsen et al., 2023]. This has triggered a distrust in the new model from the already skeptical public.

With the new building tax law coming into effect on the 1st of January 2024 [vur,], there is an urgent need to improve both the model and the public's trust in it.

A way to simultaneously do both of these is to conduct independent analyses of the property valuations. However, traditional data analysis is not feasible due to the number of parameters affecting the valuations and the sheer number of valuations. In this study, we will, therefore, be using machine learning to find abnormalities in the property valuations.

3 Method

3.1 Data

To do the analysis of the property valuations, three main categories of data are needed; Valuation, location, and property data. This means that three different data sources are required to do the analysis.

For this study, only first-party data from official Danish registries was used to ensure the highest possible data quality in terms of correctness. Furthermore, it makes the study reproducible due to the public availability of the data. An important point, as the trust of the public is one of the main objectives of this study.

3.1.1 Valuations

The Danish property valuations are stored in the Property Valuation Registry (VUR)[1]. However, the main API for this registry is not directly available to the public [2]. Furthermore, the valuations for 2022 are delayed[3], so even with access to this registry, the ideal data for this study is not available.

Therefore, the provisional valuations will instead be used. These are publicly available, however, there does not seem to be an API that is intended for the collection of this data on a large scale. Instead, the data will be retrieved from the API used to get the data for Vurderingsportalen. Through this API the valuation was gathered for one property at a time.

This limitation of only getting one valuation at a time, and not wanting to DDOS the server, led us to implement a rate limiter on our scraper to ensure zero downtime. Furthermore, the same limitation produced a temporal difficulty which evidently was recognized by the, initially, $\approx$ 100 hours it was estimated to require for a full scrape. In order to prevent compromising on the quality of the data and scrape only a subset of the properties, we decided to utilize a primitive parallel computing approach resembling distributed web scraping. By mapping partitions of indexes from DAR to multiple processors and reducing the API responses to a single file, the time was decreased, ultimately reducing the scraping time to 13 hours. The data was then forwarded from various volumes to a central volume in which a merge combined the data partitions to the final VUR data. While not adhering conventionally to it, this approach shares core parallelism and distribution characteristics with MapReduce.

3.1.2 Location data

The location data is taken from Denmark's Address Registry (DAR), publicly available through `https://datafordeler.dk/`. These addresses were used to provide a list of properties to scrape from VUR while providing the location for the properties information such as the full address, to be used when determining the abnormal evaluations.

3.1.3 Property data

The Building and Housing Registry (BBR) stores all public data on properties. As with the location data, this is made available through `https://datafordeler.dk/`, however, does require a user to download it. Luckily, a user does not require anything more than an email. Next, you subscribe to the registry, and then at the start of each week, a JSON will be published for you on Datafordeler's FTP server.

[1]`https://datafordeler.dk/dataoversigt/ejendomsvurdering-vur/`
[2]`https://datafordeler.dk/dataoversigt/ejendomsvurdering-vur/ejendomsvurdering/`
[3]`https://www.vurderingsportalen.dk/ejerbolig/ejendomsvurdering/`

3.1.4 Preprocessing

To shape the data into a usable dataset for the abnormality analysis, preprocessing was done in two phases.
First on a registry level, extracting the usable information from the JSON files received from Datafordeleren, and mapping the structure and attributes of the datasets to form a cleaned and useful dataset for the specific registry. Removing irrelevant attributes and addresses with missing information.
Secondly on a joined level, starting by inner joining the registry data based on the address id into a combined dataset, translating the attributes to English and one-hot encoding categorical attributes.

The combined dataset could now be used to further process the data. First two new attributes were created:

- **totalAddressValue** as the sum of the *propertyValue* and *groundValue* for the given address. Will serve as the valuation property
- **region** as the first digit in the postal number

Providing the full attribute list which can be seen in Appendix B

Next, all attributes concerning the area of the properties had their missing values filled with zeros. The same was done for the attributes; locationOnLakeTerritory, numberOfFloors, and shelterSpaces. Then all properties with a valuation (ground or property) of zero were removed since these are not representative of actual properties. Then, all properties missing a value for the attribute constructionYear were removed since this attribute is assumed to be vital for the valuation.
Lastly removing the attributes directly correlated with the **totalAddressValue** property, including: *groundValue*, *propertyValue*, *propertyTax*, *groundTax* and *totalAddressTax*

In the end, the dataset consists of 526,117 rows each with 279 attributes.

This proved to be quite a heavy dataset to work with, due to the high dimensionality caused by the one-hot encoded attributes, requiring more than 1 TB of RAM to do the more complex matrix computations required for the model.

Hoping to reduce this Principle Component Analysis was performed, but sadly provided poor explanation in each component, see Figure 1

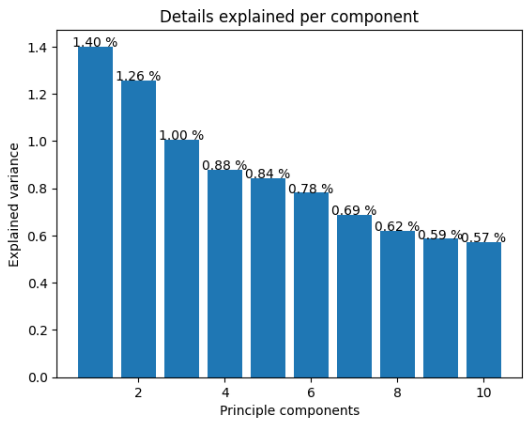


Figure 1: Explained variance per component for the first 10

As an alternative approach to reduce this problem, attributes with less than 400 observations with a value different from 0, was removed along with the data points where the attributes was different from 0, to avoid having this factor skew the valuation.

This reduction removed 159 attributes and 9,771 rows, resulting in the final dataset used for the following analysis, which can be found here: `https://drive.proton.me/urls/89G58EHFV4#8H7CXNedYN7k`

3.2 Model

The model consist of two parts:

1. Clustering of properties, providing an index for finding similar properties
2. Abnormality detection, using the clusters to determine ‘normal‘ evaluations of the properties in the clusters, then comparing each property’s valuation to the ‘normal‘ of its cluster

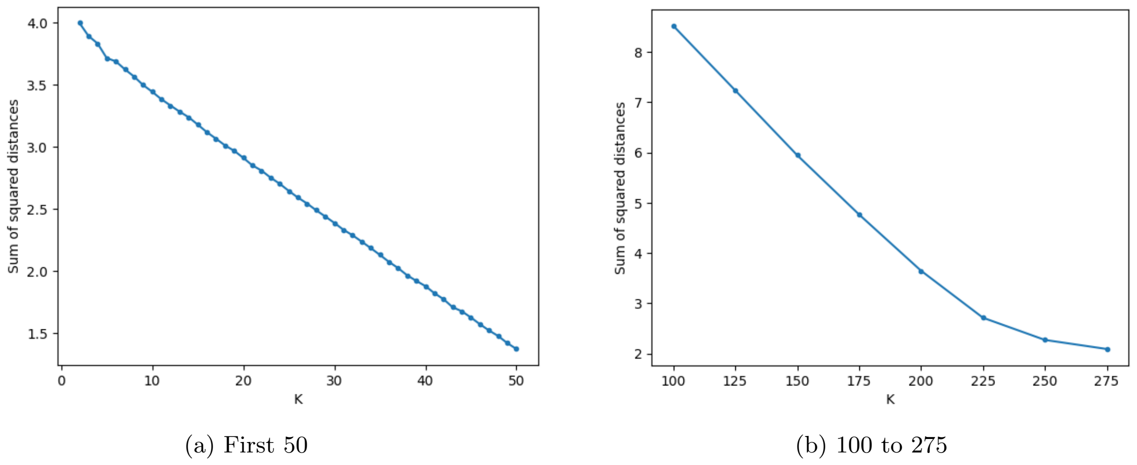
3.2.1 Clustering

The properties are clustered on the subset of numerical attributes without the **totalAddressValue** attribute, serving to cluster properties of similar location, size and attributes. Due to the huge scale difference between the attributes, every attribute except *region* was MinMax normalized ensuring the values to be in the range 0-1 to match with the one-hot encoded properties. *region* is kept as is due to the nature of the attribute, assuming properties in different regions to be fairly different in valuation, and thus hoping to induce some regional clustering.

To cluster the properties two methods was tried: K-means and DBSCAN

3.2.2 K-Means

Preliminary measures for K-Means include choosing a suitable value for K to find a balance in numbers of clusters and encourage the centroids to lie within natural clusters. For this, all the remaining attributes after prepossessing were used.



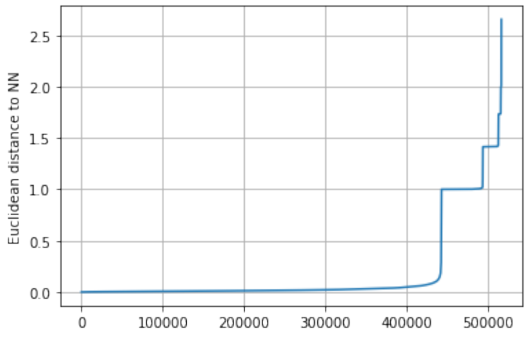
(a) First 50 (b) 100 to 275

Figure 2: Finding optimal K for Kmeans

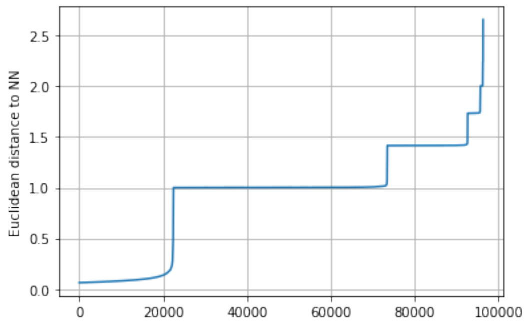
The initial search (fig. 2a) had a 11 hour runtime and no obvious elbow point, giving rise to a more coarse grained search in a K=100..300 space with 25 interval increments. This yielded (fig. 2b) an elbow point at approximately $K = 250$ clusters. If given unrestricted time, a more exhaustive search across smaller intervals in a wider range would be optimal.

3.2.3 DBSCAN

To determine the hyperparameter: max distance ϵ a distance analysis of the properties was made, determining the distance from every property to its nearest neighbour. Then plotting (Figure 3) the sorted distances to determine a sensible distance to include non-outlier looking properties.



(a) All distances



(b) Last 100,000 largest distances

Figure 3: Sorted distances to nearest neighbour for the properties

Based on these distances an ϵ of 1.1 was selected.

The min. samples hyperparameter M was selected to be 2 times the dimensionality of the data, thus 236, as suggested by [Sander et al., 1998]. Doing so produced 157 clusters and classified 46% of the properties as outliers. To try to correct for this, M was set to the dimensionality of the data (118), creating 278 clusters and classified 34% as outliers. Thus providing smaller and more specialised clusters.

3.3 Abnormality detection

Using the similarity index provided by the clusterings, abnormal valuations was determined by comparing a property's valuation to the valuations of the remaining properties in the same cluster. This was done by normalizing the property's valuation v with the mean $\hat{\mu}$ and standard deviation $\hat{\sigma}$ of the remaining valuations

$$z = \frac{v - \hat{\mu}}{\hat{\sigma}} \tag{1}$$

Then performing a t-test on the found value with a p-value of

$$p = 2 \cdot (1 - P(Z > z)) \tag{2}$$

where Z follows a 0 mean and unit standard deviation Gaussian distribution. The p value signifying the probability of seeing more extreme values than v.

The chosen significance level was $\alpha = 0.01$, essentially meaning any valuation with less than 1% probability to be observed is considered abnormal.

As DBSCAN also creates an outlier cluster, properties labelled as outliers was marked as inconclusive. The interpretation being that outlying properties are special in some sense and can thus not be fairly compared with other properties.

3.3.1 Alternative detection methods

For alternative detections, a mean and standard deviation method was tried, labelling a valuation abnormal if it changed the mean or standard deviation of the cluster more than 10%. Unfortunately this gave 0 abnormal valuations, which can be accounted to the wide ranges of valuations in the clusters. An example of these ranges can be seen in Table 1

Count	Mean	Std	Min	25%	50%	75%	Max
660	4.443e6	3.133e6	4.890e5	2.363e6	3.433e6	5.613e6	2.491e7

Table 1: Valuation properties for DBSCAN cluster nr. 69, percentage columns showing the percentiles

These methods were thus dropped.

3.4 Evaluation

Since no labeled data exists for this problem, an alternative approach had to be used to evaluate the performance. Advanced methods exist for evaluating unsupervised methods without labeled data [Goix, 2016], however, these were not chosen since explainability to the general public is one of the main objectives of this study.

Instead, this study used synthetic data where the labels are known. This was done by taking valuations classified as normal and multiplying them by either 2 or $\frac{1}{2}$ i.e. making them abnormal and labelling them as such. And taking valuations classified as abnormal and changing them to the average of valuation of the 5 nearest properties in its cluster, i.e. making them normal and labelling them as such.

Finally applying the model to this synthetic data, to obtain a prediction of abnormality. Then using these predictions versus synthetic labels to estimate the accuracy of the model.

4 Results

4.1 DBSCAN

For the DBSCAN based model 10,345 abnormally valuated properties.

But only obtained an accuracy of $\approx 54.1\%$, primarily due to the induced normal properties all being classified as normal and almost none of the induced abnormal being classified correctly, as can be seen in the confusion matrix in Table 2.

True normal	100 %	0 %
True abnormal	91.8 %	8.2 %
	Predicted normal	Predicted abnormal

Table 2: DBSCAN confusion matrix

4.2 K-Means

For the KMeans based model 14,597 abnormally valuated properties.

But only obtained an accuracy of $\approx 56.2\%$, like the DBSCAN model due to the model only recognising a very little amount of the induced abnormalities, as can be seen in the confusion matrix in Table 3.

True normal	100 %	0 %
True abnormal	87.6 %	12.4 %
	Predicted normal	Predicted abnormal

Table 3: K-Means confusion matrix

5 Discussion

The non-satisfactory results in this paper come down to three main problems: The data quality and completeness, the parameters in the used models, and the limitations of the used models.

The dependence on first-party data from official Danish registries ensured high data credibility. However, it also posed challenges due to incomplete and non-uniform data.

Another area of improvement could be the hyperparameters used for the clustering algorithms, which could be optimized using cross-validation, instead of the heuristic approach taken in this study.

The models used in this study have three main problems. Firstly, the use of Gaussians to determine abnormalities in valuations is suboptimal since the valuations within most clusters were not Gaussian distributed. Future research could, therefore, improve the models by using non-parametric tests to determine the abnormalities. Secondly, the clustering did not take the effect of each attribute on the valuation into account. This could be done using either expert knowledge from e.g. real estate agents or by using regression-based clustering algorithms. Lastly, the current models use entire clusters for distribution fitting, risking sub-optimal normality estimations due to diverse property types. Large cluster sizes create broad variance, especially in non-cylindrical DBSCAN clusters, hindering accurate normality definition. An alternative is selecting K-nearest neighbours for a property in the entire dataset for valuation distribution fitting.

Lastly, the evaluation of the model caused some difficulties as no-labeled data existed. This could in the future be avoided by using sales data for the properties for the evaluation.

6 Conclusion

The objective of the study was to build an abnormality detection model for Danish property valuations. The proposed cluster-based models was able to label a fair amount of abnormally valuated properties in Denmark, but failed to detect the majority of synthetically made abnormalities. Thus setting into question the viability of the models in a real setting as is. Despite these limitations, the research provides a foundational framework for future advancements in this area, emphasising the need for continued refinement. This groundwork sets the stage for refining models to better detect anomalies and enhancing their practical applicability within authentic real estate settings.

References

[vur,] *Vurderingsportalen.*

[Goix, 2016] Goix, N. (2016). How to evaluate the quality of unsupervised anomaly detection algorithms?

[Ingvorsen et al., 2023] Ingvorsen, E. S., Ussing, J., Hecklen, A., and Ørskov, O. (2023). Hemmelige papirer afslører ny forsinkelse: Rammer hundredtusindvis af boligejere. *DR.*

[Sander et al., 1998] Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194.

7 Appendix

A GitHub code

All code can be found in our GitHub repository: `https://github.com/PBonvang/02807-Project`

It includes both the compiled jupyter notebook hand-in and the structure we have built and executed the code with.

B Dataset attributes list

Can also be found here: `https://drive.proton.me/urls/HF4YGSPGAR#EjypwFBVmOAo`

```
{
    "id": GUID # aka. adgangsAdresseID(VUR), husnummer_id(BBR,DAR),
    "postal_nr": str # DAR,
    "address": str # DAR,
    "evaluationYear": int # VUR,
    "propertyValue": float # VUR,
    "groundValue": float # VUR,
    "propertyTax": float # VUR,
    "groundTax": float # VUR,
    "totalAddressTax": float # VUR,
    "buildingUsageType": one hot encoded # aka. byg021BygningensAnvendelse(BBR),
    "constructionYear": int # aka. byg026Opførelsesår(BBR),
    "reconstructionOrExtensionYear": int # byg027OmTilbygningsår(BBR),
    "waterSupplyType": one hot encoded # aka. byg030Vandforsyning(BBR),
    "drainageType": one hot encoded # aka. byg031Afløbsforhold(BBR),
    "outerWallMaterialType": one hot encoded # aka. byg032YdervæggensMateriale(BBR),
    "roofMaterialType": one hot encoded # aka. byg033Tagdækningsmateriale(BBR),
    "supplementOuterWallMaterialType": one hot encoded # aka. byg034SupplerendeYdervæggensMateriale(BBR) -
    "supplementRoofMaterialType": one hot encoded # aka. byg035SupplerendeTagdækningsMateriale(BBR) - code
    "asbestosHoldingMaterialType": one hot encoded # aka. byg036AsbestholdigtMateriale(BBR),
    "totalBuildingArea": float # aka. byg038SamletBygningsareal(BBR),
    "totalResidenceArea": float # aka. byg039BygningensSamledeBoligAreal(BBR),
    "totalIndustrialArea": float # aka. byg040BygningensSamledeErhvervsAreal(BBR),
    "builtArea": float # aka. byg041BebyggetAreal (BBR),
    "areaOfBuiltInGarage": float # aka. byg042ArealIndbyggetGarage (BBR),
    "areaOfBuiltInCarport": float # aka. byg043ArealIndbyggetCarport (BBR),
    "areaOfBuiltInShed": float # aka. byg044ArealIndbyggetUdhus (BBR),
    "areaOfBuiltInConservatoryOrSimilar": float # aka. byg045ArealIndbyggetUdestueEllerLign (BBR),
    "totalAreaOfClosedCoveringsOnTheBuilding": float # aka. byg046SamletArealAfLukkedeOverdækningerPåBygni
    "areaOfWasteRoomAtGroundLevel": float # aka. byg047ArealAfAffaldsrumITerrænniveau (BBR),
    "otherArea": float # aka. byg048AndetAreal (BBR),
    "areaOfCoveredArea": float # aka. byg049ArealAfOverdækketAreal (BBR),
    "accessArea": float # aka. byg051Adgangsareal (BBR),
    "numberOfFloors": int # aka. byg054AntalEtager (BBR),
    "deviantFloors": one hot encoded # aka. byg055AfvigendeEtager (BBR),
    "heatingInstallation": one hot encoded # aka. byg056Varmeinstallation (BBR),
    "heatingMedium": one hot encoded # aka. byg057Opvarmningsmiddel (BBR),
    "supplementaryHeating": one hot encoded # aka. byg058SupplerendeVarme (BBR) - code list is Opvarmnings
    "shelterSpaces": int # aka. byg069Sikringsrumpladser (BBR),
    "preservation": one hot encoded # aka. byg070Fredning (BBR),
    "stormCouncilsFloodSelfRisk": one hot encoded # aka. byg111StormrådetsOversvømmelsesSelvrisiko (BBR) -
    "areaOfExternalInsulation": float # aka. byg130ArealAfUdvendigEfterisolering (BBR),
    "locationOnLakeTerritory": bool # aka. byg136PlaceringPåSøterritorie (BBR),
    "totalAddressValue": float # sum of propertyValue and groundValue,
    "region": int # first digit of postal_nr
}
```

C Contribution table

Student / Task	Thomas	Long	Shadman	Amin	Peter
Gathering data	33%	34%	0%	0%	33%
Preprocessing and building dataset	35%	30%	0%	0%	35%
Coding KMeans	10%	60%	10%	10%	10%
Coding DBSCAN	0%	0%	30%	30%	40%
Coding abnormality detection	10%	10%	10%	10%	60%
Coding evaluation	20%	20%	10%	10%	40%
Abstract	80%	0%	0%	0%	20%
Introduction	40%	0%	0%	20%	40%
Method	33%	33%	0%	0%	34%
Results	33%	33%	0%	0%	34%
Discussion	20%	10%	30%	20%	20%
Conclusion	0%	0%	40%	20%	40%

Table 4: Contribution table