Aalto University
School of Business

AUTOMATED REAL ESTATE APPRAISAL METHODS

A comparison of multiple linear regression and artificial neural networks in real estate valuation

Bachelor's Thesis
Teemu Saha
Aalto University School of Business
Information and Service Management
Fall 2022

Aalto University
School of Business

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of Bachelor's thesis

Author Teemu Saha

Title of thesis: Automated real estate appraisal methods: A comparison of multiple linear regression and artificial neural networks in real estate valuation

Degree Bachelor's degree

Degree programme Information and Service Management

Thesis advisor(s) Tomi Seppälä

Year of approval 2023 **Number of pages** 15+4 **Language** English

Abstract

The field of real estate appraisal is in the midst of a rapid normalization of automated mass appraisal methods. Automated mass appraisal methods are computer assisted data driven appraisal methods. These appraisal methods include multiple linear regression and artificial neural networks that both fall under the umbrella term machine learning. New companies and services are built based on the use of these mass appraisal techniques, raising the importance of proper use of them and knowledge of the possible pros and cons.

This thesis compares the multiple linear regression and artificial neural networks methods on their appraisal performance on the sales data available for individuals of the Finnish city of Tampere. The motivation for this study is to better understand the possible performance differences of these two methods and see how well they can be implemented with the limited data available for the public. For reasonable comparison, several performance metrics were used along with data engineering and statistical techniques when deemed appropriate.

The analysis results align somewhat with the previous literature of the performance of the two methods. The performance of the artificial neural network method is found to be significantly better than the performance of multiple linear regression when measured by mean squared error, mean absolute percentage error and R^2. However, the performance difference could partially be due to the choice in the data engineering part as previous studies report performance advantages to be tied to dataset size and data engineering choices.

Keywords Multiple linear regression, Artificial neural networks, Real estate, Appraisal

Contents

1 Introduction

Real estate forms a significant portion of global wealth and has importance to multiple stakeholders in society as Amri et al. (2012) point out. Real estate holds a special place in societies, as it can be seen as a necessity, investment or utility. All people need shelter and a place to live, but facilities to conduct business are also important as it allows for more specialized economic activity to thrive and business centers to emerge. For these reasons, owning real estate can be a lucrative investment. Depression that started from the US housing market in 2008 highlights the magnitude that the valuation of housing can have on the economy of a financialized society. Property markets are a large portion of global wealth and inefficiencies in its valuation are thus a universal problem with large interest around the globe. On a more local scale there are multiple stakeholders for each property sale. These are often a buyer, a seller, the bank financing the buyer and government as they often tax the transaction based on the property value. Other more indirect stakeholders are people owning real estate in the area and other buyers interested in the real estate in the area of the sale, as the sale price is often used as a datapoint to gauge changes in the property value of that area. These are just a few examples of the importance of the value of good property appraisal.

Traditionally the work of real estate appraisal is done by a professional real estate agent going physically to the location and inspecting the characteristics of the property. Typically, the agent would compare the apartment to similar ones sold recently and maybe use some computational methods, but the final valuation is subject to personal preferences and impressions of the agent. Miller and Markosyan (2003) describe in their article about the evolution of real estate appraisal how the increase in amount of data and computational power made automated valuation methods more commonplace and the possibility for mass appraisal emerged. They also point out that there is a lag between the emergence of new theory and full implementation of it into the appraisal process. Although new technology and methods for real estate appraisal exist, the field has also been criticized for not fully implementing them despite the positive trajectory. Ullah et al. (2018) discuss the drivers and barriers of technology adoption and identify the withholding of information by real estate agents and service providers to be one of the main barriers for technological progress in the field.

For these reasons I chose to compare the performance of two automated mass appraisal methods using data available for individuals on the internet of the Finnish city of Tampere. The first one is an artificial neural network based on multilayer perceptron and the other is a more traditional regression method of multiple linear regression. They offer an interesting study angle as the available data has been identified to be the key factor driving performance differences of the appraisal methods in previous studies by Nghieg and Al (2001), Tay and Ho (1992) and Worzala et al. (1995). Out of the three it is Nghieg and Al who performed their study especially the amount of data in mind building on the previous work of others and their study is later used to draw conclusions on the results of the experiment in this thesis.

My research question is:

What kind of differences in performance[1] exist in the real estate appraisal using artificial neural networks compared to multiple linear regression?

[1] Performance being measured as an ability generalize debt-free price of apartment sales using given information of the sold apartments, measured by certain performance metrics.

2 Literature review

In the literature review I will go over the basics of both artificial neural networks and multiple linear regression. I will also give brief introduction to their use in real estate appraisal and how they are used to capture information on multiple dimensions of an apartment. Both methods are also capable of mass appraisal and benefit from large amounts of data.

2.1 Artificial neural networks

Concept of artificial neural networks (ANNs) originated from the will to model neural activity of animals and humans. Early idea of neural networks was introduced by McCulloch and Pitts (1943). They presented a logical calculus of neural activity, where activity of each neuron was "all-or-none" process. As research of artificial neural networks has progressed, their complexity and capabilities have had a significant increase.

In the Handbook of Neural Computation Almeida (1997) notes multilayer perceptron (MLPs) as the most used and well-known type of artificial neural network. Multilayer perceptron consists of layers of linear threshold units (LTUs) that are based on a perceptron introduced by Rosenblatt (1958). LTU is formed by input vector X (x_1, x_2 … ,x_{n-1}, x_n), weight vector W (w_1, w_2 … ,w_{n-1}, w_n), bias b and activation function f as shown in figure 1. At first the LTU calculates a weighted sum of input and weight vectors, later adding a constant term called bias. After this the calculated sum is passed to the activation function that gives the final output $\hat{h}$.

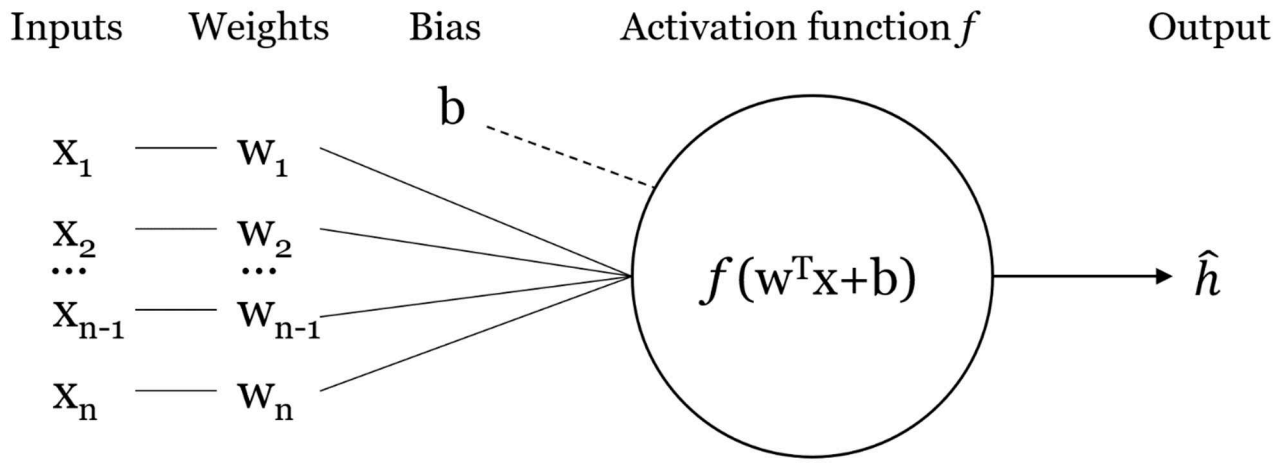

Figure 1 linear threshold unit with activation function and output. Figure by author.

2.1.1 Architecture of feedforward multilayer perceptron

Multilayer perceptron consists of input and output layer with any number of hidden layers of LTUs, as shown in the figure 2. First layer of MLP is called the input layer and is followed by the hidden layers. Input vector X is fed to each of the LTUs in the first hidden layer. The output $\hat{h}$ of one LTU then becomes an input for the LTUs in the following layer. The final layer of the MLP is an output

layer, that gives the final output $\hat{y}$. In fully connected MLP, all the LTUs of a layer are connected to LTUs of the previous and following layer.

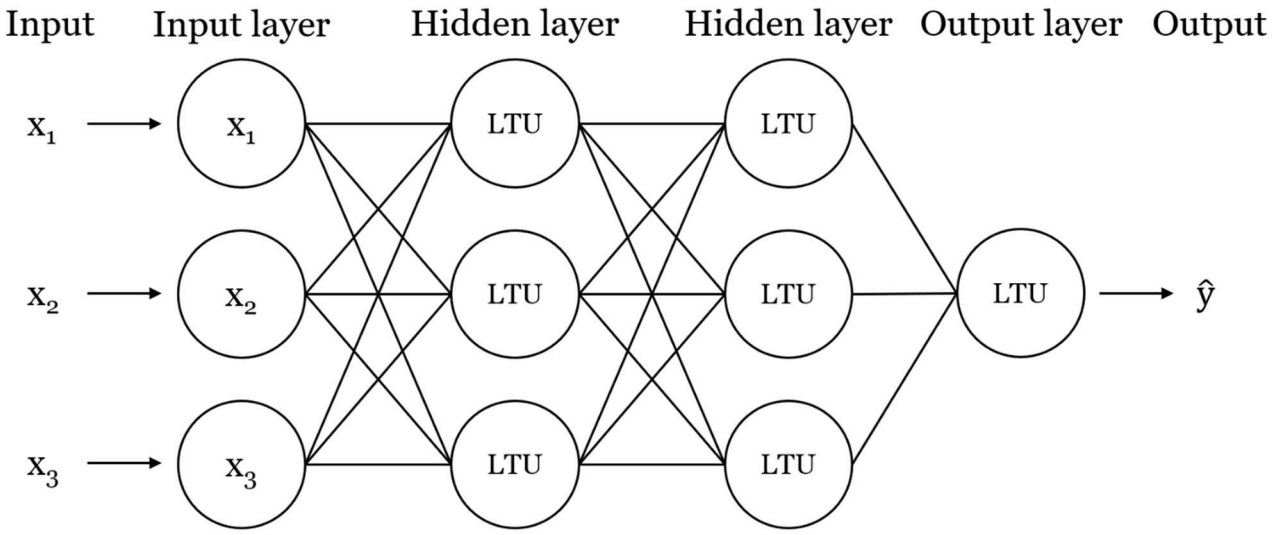

Figure 2 Fully connected multilayer perceptron with input layer, two hidden layers and one output node for scalar output. Figure by author.

Difference in the weights and biases dictate the operations the neural network can execute. Cybenko (1989) demonstrated that just one hidden layer in a feedforward neural network can approximate any continuous function. Thus, they are universal approximators. However, two problems are noted by Almeida (1997). Only the existence of such a network is guaranteed by the theorem, but procedures to finding them are not offered. The number of LTUs can be arbitrarily large even in a single layer, leading to infinite possible configurations of the network architecture.

In addition to the number of LTUs, weights and biases, the activation function also has a role in the MLP model architecture. Activation function is used to achieve the ability to approximate nonlinear functions. There are many types of activation functions for this task, but two popular types are the logistic sigmoid function (1) and rectifier linear unit (ReLU) expressed in (2). The output of the activation function can be likened to the strength of the artificial neuron impulse:

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

$$f(x) = \max(0, x) \quad (2)$$

2.1.2 Learning process

The architecture of MLPs does not include the explanation of the function approximation learning process. A separate algorithm called the backpropagation is used in combination with an optimization algorithm. The roots of the process were introduced as a multistage dynamic system optimization by Bryson and Ho (1969) and later popularized as learning process for artificial neural networks by Rumelhart et al. (1986). The learning process is iterative process that compares the output value of the network to the target value based on some predetermined loss function like mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3)$$

The calculation of the output value is named forward pass and can be written in matrix form (4), where X is the vector of inputs, f represents the activation function, $\widehat{H}$ denotes the output vector from a layer of LTUs that is passed as input to the next layer, **W** is the weight matrix and B bias vector. At the last row the output scalar $\hat{y}$ is calculated by the LTU in output layer with transposed weights vector $W^{(output)}$ and it uses the output of last hidden layer as inputs. Bias $b^{(output)}$ is the bias used in the output layer LTU.

$$
\begin{aligned}
X &= Input\ vector \\
\widehat{H}^{(1)} &= f\left(\boldsymbol{W}^{(1)}X + B^{(1)}\right) \\
\widehat{H}^{(2)} &= f\left(\boldsymbol{W}^{(2)}\widehat{H}^{(1)} + B^{(2)}\right) \\
&\vdots \\
&\textit{Output here is calculated by single LTU:} \\
\hat{y} &= f\left(W^{(output)T}\widehat{H}^{(n-1)} + b^{(output)}\right)
\end{aligned}
\qquad (4)
$$

2.1.3 Gradient descent

As there is often no way to know, what weights and biases give desired output the weights and biases are updated to minimize the error using an optimization algorithm called gradient descent. This is done by combining process called backpropagation and an optimization algorithm. Backpropagation is often misidentified as the whole learning process, but it is just the process used to calculate the gradients that are needed by the following optimization algorithm as clarified by Goodfellow et al. (2016) in their book Deep Learning. In the backpropagation the algorithm works backwards from the output layer and calculates the gradients of the parameters. Gradients are the first order derivatives of the loss function with regards to the parameters. This means that gradients include information of the parameters contribution to the error and can be used to calculate new parameters for the next iteration of the learning process as follows:

$$P^{i+1} = P^{i} - a\nabla E \qquad (5)$$

where P^{i} represents an iteration i of the parameters, ∇E is the gradients of the parameters and a is the learning rate that determines the magnitude of change in the next iteration of parameters. This way, the weights and biases gradually approach values that increase the performance of the artificial neural network. This way the choice of a loss function can have an effect on the model's performance as it is an essential part of the training process of a model.

Gradient descent can be improved by turning it into stochastic gradient descent. In this improved version of the process the aim is to decrease the noise in the gradients caused by taking just one vector of inputs into account. In the stochastic version a randomized subset of the original learning set is used to calculate the new gradients used for the parameter optimization. This subset is often called batch and is also one of the hyperparameters set by the user along the learning rate. Hyperparameter is a parameter that the user defines, like the learning rate or number of neurons in each layer of the model. Stochastic gradient descent is also faster than regular gradient descent and is described to be together with its variants the most used optimization algorithm by Goodfellow et al. (2016).

Similar to linear regression, artificial neural networks suffer from the problems of overfitting. Overfitting is a phenomenon, where a machine learning model fails to generalize the problem from the training data, and has in a way memorized the training data. As the learning process continues, the risk of overfitting the model to the test data rises. To know when to stop the iteration is one of the biggest challenges of model tuning and different methods have been developed. Separate validation data is often used to decide when the learning process should stop and the MLP model is saved with the weights and biases it has at the time. Still the validation performance varies during the process and Almeida (1997) advises to continue training even if a local minimum is found. Modern software combats this problem by adding a hyperparameter for the user that determines the number of iterations that the learning process goes through after local minimum and giving the ability to save the best model.

2.2 Multiple linear regression

Linear regression was first introduced by Legendre (1806), when he formulated the ordinary least squares method to calculate orbits of comets. Linear regression is one of the statistical models used in regression analysis and machine learning. It can be expressed in algebraic form as in (6) or in matrix notation (7).

$$\hat{y}_i = \beta_{0,i} + \beta_{1,i}x_{1,i} + \beta_{2,i}x_{2,i} \dots \beta_{n,i}x_{n,i} \quad (6)$$

$$\hat{Y} = X\beta \quad (7)$$

In the algebraic form $\hat{y}_i$ is the estimate of the dependent variable, ($\beta_{o,i}$ $\beta_{1,i}$,..., $\beta_{n,i}$) are unknown constants known as coefficients and the independent variables are denoted by ($x_{o,i}$ $x_{1,i}$,..., $x_{n,i}$) values. In the matrix notation $\hat{Y}$ is the vector of estimates for the dependent variable, X is the matrix of independent variables and β is the coefficient vector. The model is called linear regression as it is linear in terms of its coefficients as explained by Myers (1990). Independent variables can still be data that have non-linear relation to the dependent variable, like square meters or age of an apartment. If a linear regression model has only one independent variable, it is called a simple linear regression model. Likewise, the name multiple linear regression comes from the fact that there are multiple independent variables.

It is often impossible to include all possible independent variables to the regression model and an estimator is needed for this missing information. Ordinary least squares (OLS) is the name of one of the methods used to choose the unknown parameters in the linear regression model. In OLS the goal is to find the coefficients that give the smallest sum of squared residuals (8). Residuals are the differences between observed dependent variable values Y and the values $\hat{Y}$ estimated by function:

$$SSR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (8)$$

$$Residuals = Y - \hat{Y} \quad (9)$$

Taking the square of the residuals is not a trivial matter, as it affects the values of the coefficients that minimize the sum. In contrast another common method is the least absolute deviation (LAD), where the sum of the absolute values of the residuals is minimized (10).

$$SSR = \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (10)$$

The OLS method can provide the maximum likelihood estimators for the coefficients of linear regression, but only if certain conditions are met.

The residuals:

1. Must be independent
2. Follow normal distribution with mean zero
3. Have uniform variance

For these reasons the OLS method suffers if the data has outliers, fat tails or is heteroscedastic. In these cases, Narula and Wellington (1982) suggest the use of least absolute deviation regression. Later Dielman (1986) underlines the importance of examining the assumptions of OLS in his article

which discusses the differences between OLS and LAD. One way to illustrate the difference between the calculation of OLS and LAD is as a function of single residual as in figure 3.

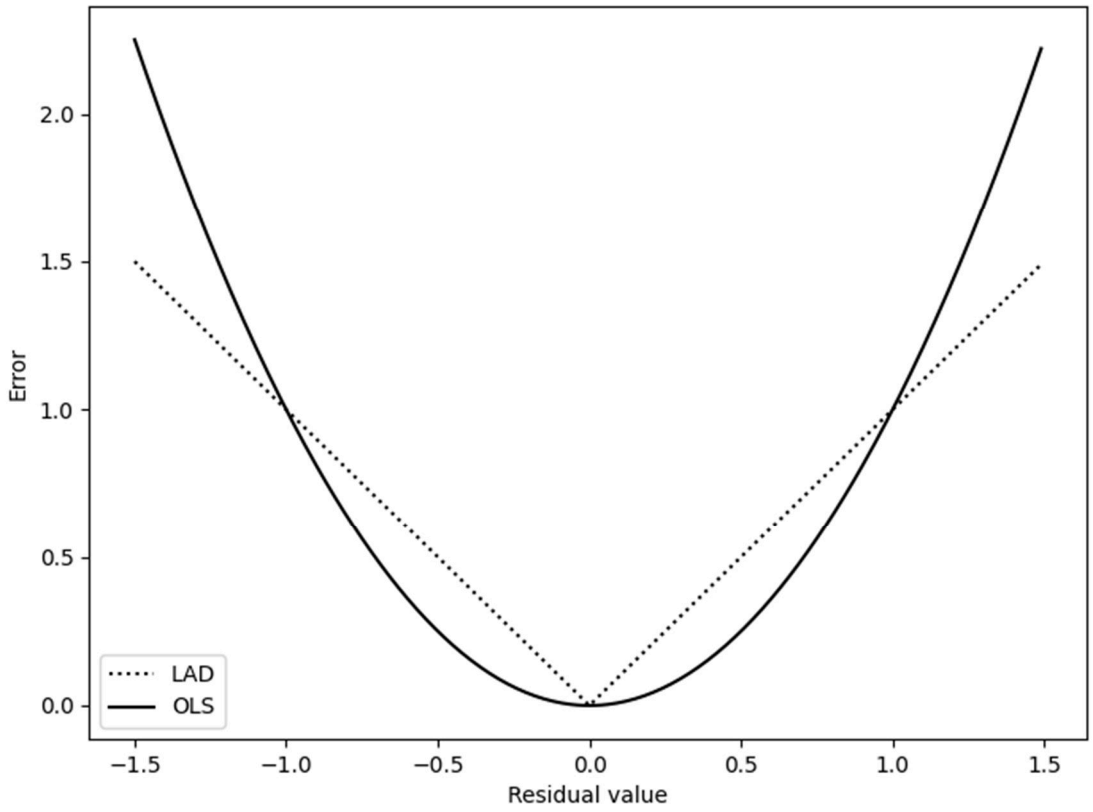

Figure 3 illustrates the difference in the resulting error value as a function of the original residual. Figure by author.

Despite its problems, the ordinary least squares regression has prevailed the test of time and is the most widely used regression to this day. This can be explained by its clear introduction by Legendre (1806) that allowed the method to become popular among scholars and the fact that if the assumptions are satisfied it is the optimum method for the estimators.

2.3 Use in real estate appraisal

Multiple linear regression has been used to value real estate and estimate the relative importance of different characteristics of housing from the 1960s as computers made automated valuation models possible, explain Miller and Markosyan (2003). Kain and Quigley (1970) describe housing as a joint purchase of multiple services at a certain location. By this they mean the amount of living space, different types of rooms, an address, accessibility to employment, public services and other characteristics of the apartment including the surrounding area. This illustrates the many dimensions housing has and the variety of independent variables that can be used to formulate the model used for appraisal. Kain and Quigley use multiple linear regression models to measure the value of housing quality from both owners' and renters' perspectives. However, early use of regression methods was limited due to the size of the computers and available data at the time. Regression methods became more widespread as access to computers and data became common. Pagourtzi et al. (2003) listed multiple regression method as traditional valuation method, highlighting the position it had already gained at the time. Artificial neural networks were listed in the same article about different appraisal methods as advanced valuation method.

One of the first known uses of artificial neural networks in real estate appraisal was in the early 1990s by Tay and Ho (1992). They used the same kind of neural networks consisting of multilayer perceptrons and utilizing backpropagation with gradient descent as described before. Similar to the multiple linear regression models, an artificial neural network can take the same independent variables and estimate the dependent variable. One of the advantages however is the nonlinearity that the neural network can achieve. Goodman and Thibodeau (1995) show that age of the house has a nonlinear relationship with its value as older properties depreciate less per year than new ones. In previous literature artificial neural networks have been found to perform better than multiple linear regression models, even when the issues of multiple linear regression are addressed. Nghieg and Al (2001) did multiple comparisons between the two appraisal methods with different datasets and models, in addition to converting nonlinear relationships to linear before the regression analysis. But they noted that in order for the ANN models to perform better in terms of mean absolute percentage error (MAPE), moderate or large sample size is to be used. Otherwise, the results could be the unsupportive and performance of ANN models inconsistent as Worzala et al. (1995) had with their study that was based on 288 sales of homes in Fort Collins, Colorado.

Now at the start of the 2020s there are multiple companies utilizing automated valuation models and automated valuation models have become more commonplace as smart real estate (SRE) has become more prominent in the industry, offering cost benefits to multiple stakeholders that utilize new technologies like cloud computing and big data. Ullah et al. (2018) underline the benefits of automated valuation methods in the real estate business accompanied by other modern trends like software as a service (SaaS) from the sustainability point of view.

3 Data and feature engineering

A raw dataset was collected from a Finnish website asuntojen.hintatiedot.fi on 22.4.2022. The web service is provided by the Finnish Ministry of the Environment and The Housing Finance and Development Centre of Finland for the individuals seeking housing sales data. The data provided by the service does not include all housing sales in Finland and those included in the service are limited to the latest 12 months. Despite these challenges I found it to be the best source of data for the model comparison, as it includes information that can be used as features in the final dataset.

The raw data consisted of neighborhood, description of rooms, living area in square meters, debt-free price, debt-free price per square meter, year of construction, floor, existence of elevator, assessment of apartment quality, is the land rented or owned and energy classification. Only apartment buildings are included as they are the most common housing type and thus offer more datapoints. The raw data was not ready for use, as many of the features were in impractical form as strings or otherwise not suitable for the model training and validation. I will now explain the final features chosen for the dataset used in model training and validation, along with the other attributes of the dataset.

The dependent variable is the debt-free price of an apartment and is not changed in any way. Debt-free price per square meter is omitted as it is directly linked to the dependent variable. Living area in square meters and the year of construction were kept as is, but other independent variables needed some work. Assessment of apartment quality and energy classification were enumerated from their string form to corresponding number values. Description of rooms was in the original data in one cell, but it often had multiple attributes of the apartment. It had the number of regular rooms, if there was a kitchen or a sauna in the apartment along with many other special room types. I chose to include the number of regular rooms as an independent variable as the records of other room types had inconsistent notation. I decided to use relative floor level as the independent variable, dividing the apartment floor number with the total number of floors. This is done assuming that the floor number of the apartment is more comparable to the number of floors in the same apartment building rather than all the floors of the other buildings in the dataset. If the relative floor ratio of an apartment is 1, it means that the apartment is at the top of the apartment building, regardless of its height. The final size of the dataset was 807 observations. Shorter descriptions of variables are given in table format:

Dependent variable	**Type**
Debt-free price	Price
Independent variables	**Type**
Living area	Square meters
Year of construction	Year
Apartment quality	Enumerated to 1-4
Energy class	Enumerated to 1-6
Number of rooms	Number of regular rooms
Floor	Relative floor value
Elevator	Dummy / Binary
Land owned / rented	Dummy / Binary
Neighbourhoods	Dummy / Binary

Table 1.

Rest of the features of the dataset are dummy variables corresponding to the existence of elevators, if land is owned or rented in addition to all the 61 neighborhoods included in the final dataset. Neighborhoods with only one sale were excluded as they made training the models impractical. Four outliers were also excluded for their relatively low debt-free price, which made the model training inconsistent. The number of dummy variables is ample, but it became clear in the early testing and training that both the multiple linear regression and artificial neural network models benefited from them.

4 Methodology

I build the model training and evaluation environment in python using scikit-learn and TensorFlow as the main libraries. They are both widely used in machine learning applications and offer built- in functionality for many of the tasks needed in model training and validation. TensorFlow was used to build and train the ANN models, whereas scikit-learn was used for the linear regression models and metrics.

4.1 Models

The multiple linear regression model is just a regular regression model with intercept, but the architecture of the artificial neural network can have multiple working configurations. As mentioned before, the multilayer perceptron can generalize functions, but there is no way to know the best possible configuration. To tune the hyperparameters and find reasonable model architecture, I trained multiple permutations of models with different architecture and settled on the following hyperparameters and model architecture.

The network has one hidden layer of size 250, an input layer of the size of the input vector and an output layer with one neuron. ReLU was chosen as the activation function and batch size is 10. The maximum number of epochs the models can do is 3000, but training will stop if no improvement in validation performance is found after 100 epochs from the last best epoch, after which the best model is saved.

Optimization algorithm used is a variation of the stochastic gradient descent called Nesterov Accelerated Adaptive Moment Estimation (Nadam) introduced by Timothy Dozat (2016). It is itself a combination of two optimization algorithms Nesterov's accelerated gradient (NAG) and Adam which both aim to tune the learning rate used to update the weights in the ANN during training. Adaptive learning rate can lead to better model performance, as it helps to find good local minimums faster and more accurately than stationary ones. Loss function used in the optimization is the mean squared error, as the aim is to compare the performance of artificial neural network to multiple linear regression.

I decided to use Nadam as an optimization algorithm as it is one of the most recent widely used algorithms and is generally able to find better values for the parameters that are updated in the training of the models to minimize residuals. Advanced optimization algorithms try to combat the problem where the model's training reaches a stop as the learning process gets stuck to a bad local minimum. This means that the model could possibly be trained to give more accurate results, but the optimization algorithm cannot reach the weights and biases that result in a such configuration of the neural network

4.2 Cross-validation and data splitting

The performance of a model can vary greatly when the dataset is small. The small sample size leads to uncertainty in the estimated average test error, as explained by Goodfellow et al. (2016). To combat this problem, the original dataset can be split into k separate segments and each segment can be used as a validation dataset for the rest of the segments. This leads to k-number of models, each containing their own approximation of the model parameters and performance. Now the mean of the models' performance can be calculated and used as an approximation of the test error. This is one of the most popular modern cross-validation techniques called k-fold cross-validation. It

sacrifices computational energy and time for more information of the method's performance. There are other cross-validation methods like the leave one out cross-validation where the idea is the same as in k-fold cross-validation, but now each datapoint is used to validate model trained on the rest of the dataset for each of the datapoints. This is a special case of k-fold cross -validation where the k equals the number of datapoints in the dataset.

Rodriguez et al. (2009) showed that K-fold cross validation can be improved by repeating the process multiple times with a randomly shuffled dataset for each repetition and using the mean of resulting models as the estimator for error. As my data only includes 807 datapoints, I have deemed it justified to use repeated k-fold cross validation for the approximation of the error of multiple linear regression and artificial neural network models.

As the goal is to evaluate the generalization performance, data is split into three different parts for each individual model. The training data is used to train the models and the performance is then measured using the validation data. After that the validation data performance is used to select a subgroup of models representing the machine learning method. If the performance of this subgroup was graded based on the validation data, a selection bias would be introduced to the final performance scores. To avoid that, a third portion of the dataset not used in either training or validation is used to score the subgroup of models. This third portion of the dataset is called the test data.

Both methods are evaluated on their test set generalization performance based on three different metrics. These metrics are the mean squared error (11), R^2 (12), and mean absolute percentage error (13).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (11)$$

$$R^2 = 1 - \frac{\Sigma(y_i - \hat{y}_i)^2}{\Sigma(y_i - \bar{y}_i)^2} \quad (12)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \cdot 100\% \quad (13)$$

The training process was divided to three different runs shown in figure 4. In each run the data was shuffled to 10 different datasets with unique order of observations. After this the k-fold cross validation method was applied with k = 10 on each of the shuffled datasets. This resulted in 100 models per each run, for both appraisal methods. After this, the top 10 performing models were chosen to the test group based on the mean squared error. After the three training and validation runs, a total of 30 models were left for the test groups for both appraisal methods. Finally, the performance of these models is measured with the test data for each of the three metrics.

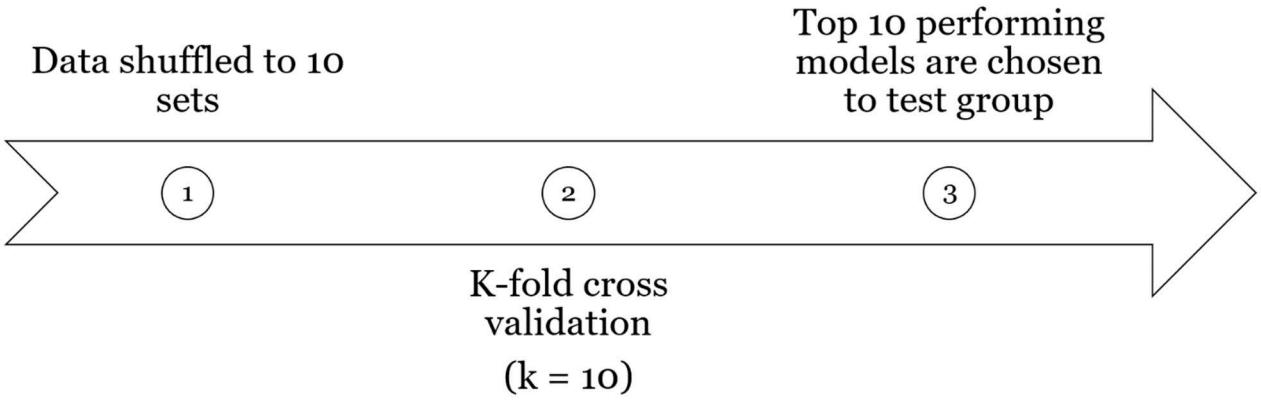

Figure 4 illustrates the training process of one run

I chose to train 30 models in total for the test groups, as it is commonly accepted as a reasonable sample size for central limit theorem. According to the central limit theorem means of a sufficiently large samples of population adhere to a normal distribution. This allows us to apply parametric test such as the student's t-test, which require the assumption that observations are normally distributed as explained by Kwak et al. (2017). An independent samples two tailed student's t-test was used to evaluate the differences between the performance of the artificial neural network and the multiple linear regression models for each of the performance metrics. Significance was decided based on the resulting p-value of the test. An independent samples two tailed student's t-test is a modification of the standard students t-test and is often called the Welch's t-test. The t value is calculated as follows:

$$t = \frac{\overline{Z_1} - \overline{Z_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{14}$$

Where the $\overline{Z_1}$ is the mean of first sample, N_1 the size of the first sample and s_1^2 the standard deviation of the first sample. Same values are denoted with subscript 2 for the second sample.

Most significant difference to the standard student's t-test in Welch's t-test is the way degrees of freedom are calculated. Here v is the degrees of freedom for Welch's t-test:

$$v = \frac{\left(\frac{s_1^4}{N_1} + \frac{s_2^4}{N_2}\right)^2}{\frac{s_1^4}{N_1 - 1} + \frac{s_2^4}{N_2 - 1}} \tag{15}$$

I chose to apply the Welch's t-test as it does not require the assumption of equal variance between the samples as shown by Welch (1938), offering more robust results if the variance of the two samples differs in the testing.

5 Results

The results are divided into two parts. First is the means and standard deviation of the performance metrics, second the p-values from the Welch's t-test. In the table 1. we can see the means and standard deviations for each of the performance metric distributions. They show the mean performance of the 30 models for both of the appraisal methods and the standard deviation for each metric. For the metrics smaller mean squared error and mean percentage error is good, whereas higher R^2 is better.

	Artificial Neural Network		
	MSE	R^2	MAPE (%)
Mean	1013 069 391	0,872	14,221
Standard Deviation	186 024 265	0,054	1,605

	Multiple Linear Regression		
	MSE	R^2	MAPE (%)
Mean	1617 802 863	0,744	18,389
Standard Deviation	301 555 187	0,102	1,595

Table 2.

As the results show, artificial neural network models overperform multiple linear regression in all metrics, while having smaller standard deviation in the performance when measured by mean squared error or R^2. The performance deviation measured by mean percentage error is almost equal. However, to see how significant the observed differences are, Welch's t-test was applied to each distribution pair.

As I applied the Welch's t-test to each performance distribution pair, three tests were conducted in total. The null hypothesis for each test was that the samples of the means of the performance distributions are from the same distribution. Meaning that the difference in means for the performance metrics are result of a random chance, rather than true difference in performance.

H_0 : Samples of the means of the performance distributions are from the same distribution.
H_1 : Samples of the means of the performance distributions are not from the same distribution.
The p-values for the tests for each performance metric are as follows:

$P_{\text{Mean squared error}}$	<	0.0001
$P_{\text{R-squared}}$	<	0.0001
$P_{\text{Mean absolute percentage error}}$	<	0.0001

Table 3. As the p-values were very small, simpler < 0.0001 notation was chosen instead of the exact numbers.

As can be seen from the results of the two tailed Welch's t-test, the difference in performance is highly unlikely to be a result of random chance. Null hypothesis is rejected with very high confidence in all of the three tests. It appears that artificial neural networks have clear advantage using the methods and data of this thesis. They have sizable lead in all three metrics with significantly better mean performance. They also seem to be more reliable method, as there is smaller measured standard deviation in performance measured by MSE and R^2.

6 Conclusions

In this thesis I compared the performance of two different automated valuation methods for real estate mass appraisal. I used data from a Finnish website asuntojen.hintatiedot.fi that included information of apartment sales from the city of Tampere. This data was then engineered and cleaned for the model training and analyzing process and final dataset had a total of 807 datapoints.

The methods compared were the multiple linear regression and artificial neural network consisting of multilayer perceptrons. To measure the generalization performance of both methods, I trained three groups of 100 multiple linear regression and another three groups of 100 ANN models and applied repeated k-fold cross-validation to gather the subgroups representative of both methods with size of 30. The performance of these subgroups was then tested on a separate test data for each model. Meaning that each model had different training, validation and testing data. The performance was presented as an aggregate of the testing performance of both appraisal methods by each performance metric. Those metrics were the mean squared error, R2 and mean absolute percentage error. The significance of the observed performance differences was evaluated using the independent samples two tailed student's t-test, otherwise called the Welch's t-test.

The results of this experiment were strongly in favor of the artificial neural network method. The artificial neural network models had significant lead in all performance metrics. This is somewhat surprising, as my dataset only had 807 observations, and needed to be split into smaller subsets of training, validation and test data. Previous studies like the one by Nghieg and Al (2001) have found small datasets to favor multiple linear regression. In their study, the artificial neural network started to outperform the multiple linear regression when the training dataset reached size of 1506.

The difference could be a result of differences in the data engineering, where the data used by Nghieg and Al (2001) was more suited for multiple linear regression models, than the data used here. Another reason might be differences in the architecture of the artificial neural networks. Perhaps advances in the optimization algorithms have lessened the data needed to train more robust models. Possible further improvement and topic of further research could be to test differently engineered features or use different validation and model selection criteria for the final test group. The effects of choosing different optimization algorithms for this particular task could also be a source of further study. Data from other cities like the city of Helsinki could also be used to gain a larger dataset to see if the improvement of the artificial neural network method found in previous studies holds in Finnish apartments with the data available from them.

7 References

Almeida, L. B. (1997). C1. 2 Multilayer perceptrons. *Handbook of Neural Computation C*, *1*.

Amri, S., & Tularam, G. A. (2012). Performance of multiple linear regression and nonlinear neural networks and fuzzy logic techniques in modelling house prices. *Journal of Mathematics and Statistics*, *8*(4), 419-434.

Bryson, A. E., & Ho, Y. C. (1969). Applied optimal control. 1969. *Blaisdell, Waltham, Mass*, *8*(72), 14.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303-314.

Dielman, T. E. (1986). A comparison of forecasts from least absolute value and least squares regression. *Journal of Forecasting*, *5*(3), 189-195.

Dozat, T. (2016). Incorporating nesterov momentum into adam.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goodman, A. C., & Thibodeau, T. G. (1995). Age-related heteroskedasticity in hedonic house price equations. *Journal of Housing Research*, 25-42.

Kain, J. F., & Quigley, J. M. (1970). Measuring the value of housing quality. *Journal of the American statistical association*, *65*(330), 532-548.

Kwak, S. G., & Kim, J. H. (2017). Central limit theorem: the cornerstone of modern statistics. Korean journal of anesthesiology, 70(2), 144-156.

Legendre, A. M. (1806). *Nouvelles méthodes pour la détermination des orbites des comètes; par AM Legendre...* chez Firmin Didot, libraire pour lew mathematiques, la marine, l'architecture, et les editions stereotypes, rue de Thionville.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, *5*(4), 115-133.

Miller Jr, N. G., & Markosyan, S. (2003). The academic roots and evolution of real estate appraisal. *The Appraisal Journal*, *71*(2), 172.

Myers, R. H., & Myers, R. H. (1990). *Classical and modern regression with applications* (Vol. 2, p. 488). Belmont, CA: Duxbury press.

Narula, S. C., & Wellington, J. F. (1982). The minimum sum of absolute errors regression: A state of the art survey. *International Statistical Review/Revue Internationale de Statistique*, 317-326.

Nghiep, N., & Al, C. (2001). Predicting housing value: A comparison of multiple regression analysis and artificial neural networks. *Journal of real estate research*, *22*(3), 313-336.

Pagourtzi, E., Assimakopoulos, V., Hatzichristos, T., & French, N. (2003). Real estate appraisal: a review of valuation methods. *Journal of Property Investment & Finance*.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, *32*(3), 569-575.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.

Tay, D. P., & Ho, D. K. (1992). Artificial intelligence and the mass appraisal of residential apartments. *Journal of Property Valuation and Investment*.

Ullah, F., Sepasgozar, S. M., & Wang, C. (2018). A systematic review of smart real estate technology: Drivers of, and barriers to, the use of digital disruptive technologies and online platforms. *Sustainability*, *10*(9), 3142.

Welch, B. L. (1938). The significance of the difference between two means when the population variances are unequal. *Biometrika*, *29*(3/4), 350-362.

Worzala, E., Lenk, M., & Silva, A. (1995). An exploration of neural networks and its application to real estate valuation. *Journal of Real Estate Research*, 10(2), 185-201.