IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 14, Issue 6 Ver. I (Nov - Dec 2018), PP 46-60
www.iosrjournals.org

Comparative study of linear models and neural networks for rational decision making

Tzotzis Christos[1], Makris C. Georgios[2]
[1]*Postgraduate student of Inter-Faculty Master Program on Complex Systems and Networks of Aristotle University of Thessaloniki.*
[2]*Department of Mathematics, Aristotle University of Thessaloniki, Greece*

Abstract:*Recent studies on urban economics and real estate theory have highlighted how investment can affect real estate values. In addition, city residents pay more attention to the quality of their living space. The rate of crime, air quality, access to employment centers or highways are just some of the parameters that can influence the housing and location options and hence land use, mobility and the economy of the citizens concerned. Several are the interested citizens who choose to invest either by risking or by dragging on only some parameters, without thoroughly scrutinizing all available ones. The legitimate question that arises here is whether a citizen who is interested in investing is in a position to make a rational decision when he receives a large amount of data and has limited time. A second question is whether this decision could have taken a model of mechanical learning.*
The main objective of this work is to establish a model of property valuation based on artificial neural networks for estimating property prices. This model could also help analysts simulate the interactions that arise in an urban system where location options for houses or businesses depend heavily on the real estate market. Artificial neural networks are a very popular process of identifying the best solution for various problems. In the present study twelve different artificial neural networks and four linear models are constructed to control the limitations of rational decisions taken by them. These models are applied to prediction of the average property price in Boston suburbs and their predictions are made relative to real prices. The evaluation of these models was done by the repeated cross-validation method, while for comparison, the root mean square error and the coefficient of determination were used. From the results obtained, it is concluded that both linear models and artificial neural networks are able to obtain a satisfactory and also a rational decision. The comparison of models highlights the artificial neural network as the most appropriate prediction model for the experiment.

Keywords **-**Artificial Neural Networks, Bounded Rationality, Linear Models, Elastic Net.

Date of Submission: 18-11-2018 Date of acceptance: 04-12-2018

I. Artificial Neural Networks

Artificial Neural networks are a relatively new area in the natural sciences, as they have been known and developed internationally only in recent decades. The main feature of neural networks is that they are a computational model inspired by the way the brain's biological networks process the information. However, their study and use has gone far beyond biological organisms. They are now used to solve all kinds of computer problems [1]. Their function tries to combine the thinking of the human brain with the abstract mathematical way of thinking. But of course they can also use complex mathematical functions and all sorts of tools from mathematical analysis.

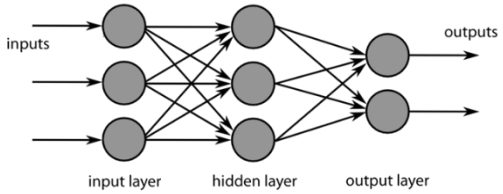

Figure 1:Artificial neural network

A neural network consists of three levels (Figure 1). Initially from the Input layer, which contains the variables that provide the signal to the neural network. No kind of calculation is happening at this level. Then, hidden layers (Hidden nodes), which have no contact with the outside of the network. They make calculations

and carry the signal from the first level to the last one. n a simple neural network, there must necessarily be the first and the last level (input and output respectively), but may have multiple or no internal levels. Finally, the Output layer, which is the last level of a neural network and is responsible for the computation and transmission of information [2], [3], [4].

II. Bounded Rationality

Bounded rationality means that the decision maker acts rationally but his cognitive abilities are limited. Man basically processes the information serially, but its computational capabilities are limited due to its very low capacity, its short-term memory and the relatively long time it takes to transfer a memo from short to long term memory. Time and ability to identify, acquire and process information-related information is also limited [5].

A rational decision is the best decision possible in the context of resource and condition data. The person does not thoroughly examine all of the data, since he or she does not even have a chance to look at them. Instead, it is chosen as the best solution, the one that seems to meet the needs and is satisfying. In most situations we do not have full information about all possible alternatives. We do not have this full knowledge because of limited time and high costs [6]. Any alternative always creates unpredictable consequences, which can prove to be particularly important. We rarely have one set of criteria, defined with clarity or absolute consistency, by which we can classify the available alternatives to choose the most desirable.

Herbert Simon proposes a different perspective and argues that people have "bounded rationality", so that while trying to be rational, their behavior is limited by their cognitive skills and limitations that are part of the organization. The person's rational capacity is marginal in at least three ways:

a) Rationality requires a thorough knowledge of the consequences that will follow after each choice. In fact, knowledge is always fragmentary.

b)Rationality requires a choice among all possible alternatives. In fact, very few of all possible alternatives concern our thinking [7].

c) In order to meet their finite rational capacity and the complexity of the problems, people adopt different strategies. These allow them to "simplify" the problem and focus, with selective representation, on its more prominent features [8].

In the first phase of the decision-making process, man explores the environment for situations requiring a decision. In the second phase, he devises, develops and analyzes possible action criteria. In the third phase, he examines the alternatives available. Finally he gets the best of them.

According to Simon, cognitive processes that trigger limited rationality are not intended to maximize profit or utility, but to find a satisfactory solution that he calls satisfying, a word that results from mixing the two words sufficing (sufficient, enough) and satisfying [9]. Decision-making, whether individual or collective, is concerned with the discovery and selection of satisfactory alternatives, and only in exceptional cases is concerned with the discovery and selection of the best alternatives [8]. An alternative is considered best if it is superior to all others, according to a single set of criteria used to compare all available alternatives. This is considered to be satisfactory if it is equal to the set of criteria defining "the least satisfactory alternatives". "Type satisficing behaviors simplify a complex world. Instead of worrying about an infinite number of gradations of the problem, the individual simplifies it in two parts - satisfactory and not satisfactory enough "[10].

III. Experiment

In the present work, various models, linear and non-linear, were designed to predict the average house price using all the available variables. The R language was used to conduct the experiment. The Housing Values in Suburbs of Boston experiment originated from the MASS library [11].

For the needs of the experiment four linear models and twelve artificial neural networks were constructed. Linear models consist of linear regression, ridge regression, lasso regression, and elastic net regression. The construction of the linear models and their training was done with the help of the libraries, “gmlnet: generalized linear net”, “mlbench package” and “psych package”. The construction of neural network technicians and their training was done with the help of the “neuralnet: Training of Neural Networks” [12], [13], [14], [15], [16]. Then, we compared the linear models with each other to find the best linear model and similarly compare the artificial neural networks to each other to find the best artificial neural network. Finally, we compared the two models to find the best one. The indices used to compare models are the root mean square error (RMSE), as well as the determinant factor (Rsquared or R^2).

Data analysis and modeling

The data which collected consists of 506 lines (values) and 14 columns (variables) and relates to property prices in the suburbs of Boston. The following is a description of the variables.

X_1: Crime rate

X_2: The proportion of habitable land over 25,000 m².
X_3: The percentage of non-retail businesses
X_4: If it is a riverside
X_5: Concentration of nitrogen oxide
X_6: Average number of rooms per house
X_7: Ownership rate for buildings before 1940
X_8: Average distances in 5 employment centers
X_9: accessibility index on a highway
X_{10}: Real estate tax per 10000 $
X_{11}: Proportion of pupils and teachers
X_{12}: percentage of African Americans in the city
X_{13}: Living standard percentage
Y_1: average house price per 1000 USD (predictive variable)

Construction of linear models

We want to create a model of prediction of the average price of real estate as a function of the other variables. We observe that most variables are numerical with the exception of one, which is qualitative and accepts as values 0 and 1. First, we examine the correlation between the variables. So we will exclude the fourth variable, which is qualitative as well as the fourteenth variable, which is the average property value and therefore the forecast variable.

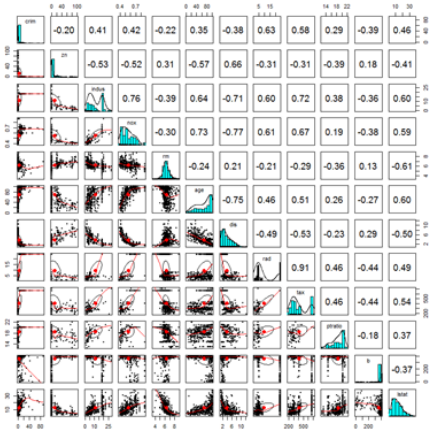

Figure 2:The correlation of all independent variables in each possible combination per two

When the correlation is high, a prediction variable in a multiple regression model can be predicted linearly by the others with a significant degree of accuracy. In this case, coefficient estimates of multiple regression are not constant, so the model is not accurate. This problem is called multicollinearity and brings to our model the overfitting phenomenon in which we do not get accurate predictions. By the term overfitting we refer to the situation where in generalization, namely the phase that follows the training and learning of the model and in which the user gets the results given by the model for new input vectors, with which the model does not has been trained, we observe that the model may have "memorized" the results of the education phase but has not learned to generalize them correctly on the new entry data. Overfitting causes a relatively small error on the outputs resulting from model training but also a significant error in the outputs occurs when new data is given to the model [17]. The solution to the problem is given by three models:

1.Ridge regression, which shrinks the coefficients to non-zero values to prevent over-education, but keeps all variables [18].

2.Lasso regression, which shrinks the regression coefficients with the sum shrinking to zero. It also helps in selecting features [19].

3. Elastic net regression, which is a combination of the two previous models [20].

We then divide the data into two separate segments, one containing 75% of the data and used for training, and the other 25% for testing. We will create the customized control parameters with the 10-fold cross validation method [21]. With this method, training data is divided into ten sections. The model is constructed from the 9 sections and the one remaining section is used to estimate the error. This process is repeated 5 times and each time a different section is used to estimate the error. So now we are training on models.

Linear regression

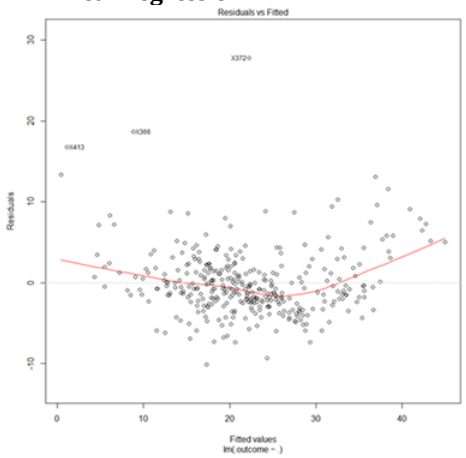

Figure 3: The residues as a function of adjusted values

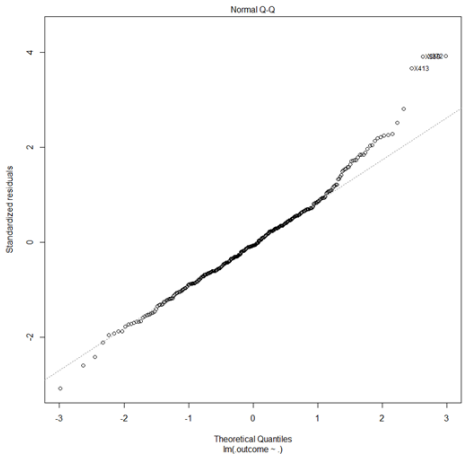

Figure 4: Standard Residues vs. Theoretical Quantities

This graph shows whether residues have non-linear patterns. There could be a non-linear relationship between the prediction variables and an outcome variable, and the pattern could appear on this diagram if the model had not recorded the non-linear relationship. In this diagram we do not see any distinctive pattern.

This graph shows whether residues are normally distributed. Residues follow a straight line or deviate seriously. It is good if the residues are aligned well in the straight dotted line.

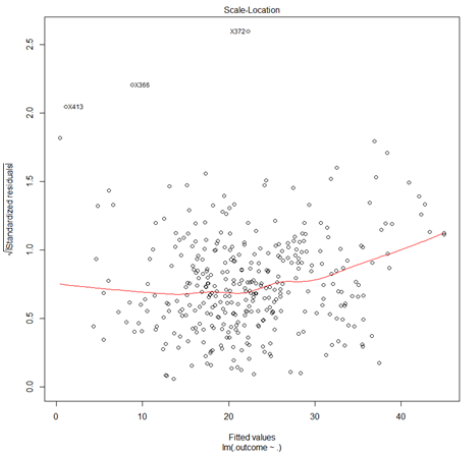

Figure 5: The root of standardized residues as a function of the adjusted values

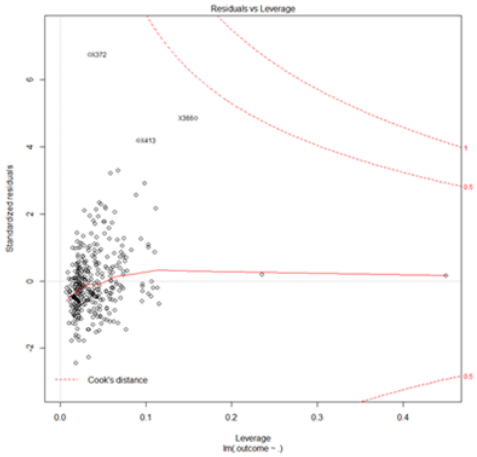

Figure 6: Standard residues vs. part of the observations

This graph shows whether the residues are distributed equally along the prognostic ranges. This is the way we can control the hypothesis of homoscedasticity. It is good if we see a horizontal line with equally (random) propagation points.

We monitor the remote values in the top right corner or in the lower right corner. These points are where the cases can affect the regression line. When incidents are outside the Cook's distance (meaning that they have high degrees of Cook's response), cases affect regression results. In our case, we have the typical appearance because there is no case influence. We can look at the Cook's distance lines (a red dotted line), because all the cases are within the Cook's distance lines.

Ridge regression

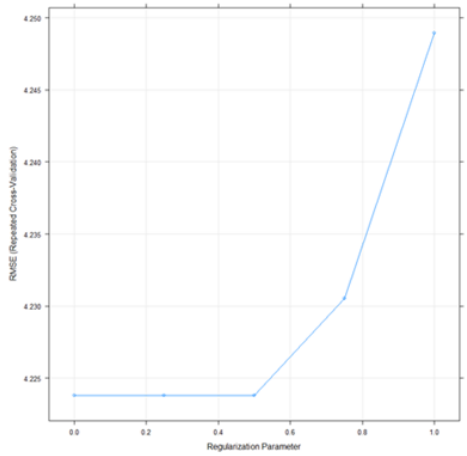

Figure 7: The root mean square error, as a function of normalization parameters

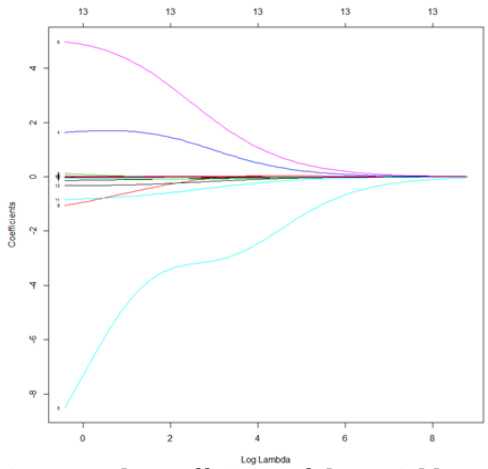

Figure 8: The coefficients of the variables as a function of the lambda logarithm

The root mean square error, calculated by repeated cross validation, as a function of normalization parameters. We notice that as the value of the lambda parameter increases, the error increases. The lowest error value is for lambda = 0.5.

For the log lambda = 9 we notice that the coefficients are zeroed, while for smaller values they begin to grow. The numbers in the top row show how many variables the model has in each case.

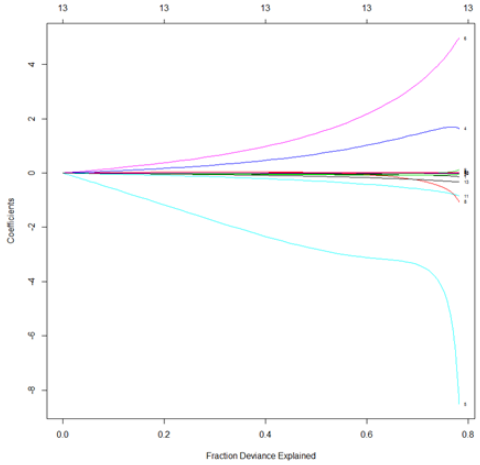

Figure 9: The coefficients of the variables as a function of the deviation of the fraction

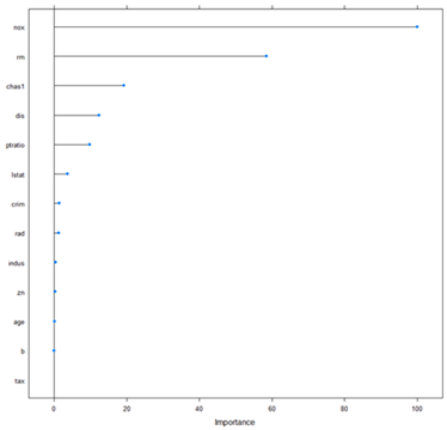

Figure 10: The significance of each variable in the model

We notice that as soon as the coefficients begin to grow, 20% of the variables have been described. When the coefficients are extremely limited and over 80% of the variables have been described, it is highly probable that our model will create overfitting.

This chart shows which variables are more important. We note that our model considers only two variables highly significant (over 30%), while eight variables are considered to be of no importance.

Lasso regression

If the correlation between some variables is high, the model selects a variable from that group and ignores the others.

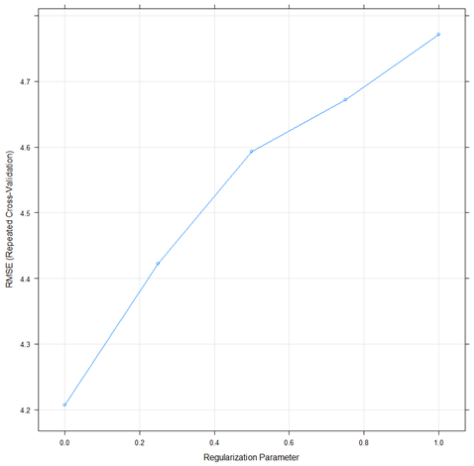

Figure 11: The root mean square error, as a function of normalization parameters

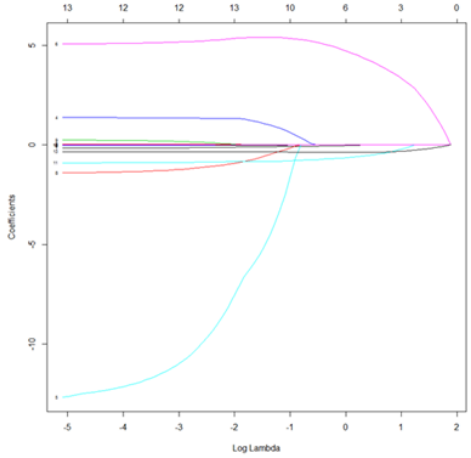

Figure 12: The coefficients of the variables as a function of the lambda logarithm

The root mean square error, calculated by repeated cross validation, as a function of normalization parameters. We notice that as the value of the lambda parameter increases, the error increases. The lowest error value is for lambda = 0.0001.

For the log lambda = 2 we notice that the coefficients are zeroed, while for smaller values they begin to grow. The numbers in the top row show how many variables the model has in each case.

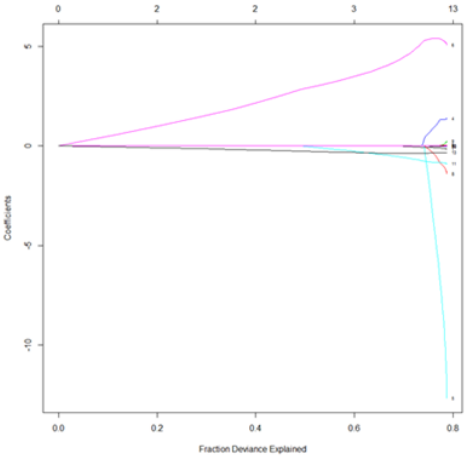

Figure 13: The coefficients of the variables as a function of the deviation of the fraction

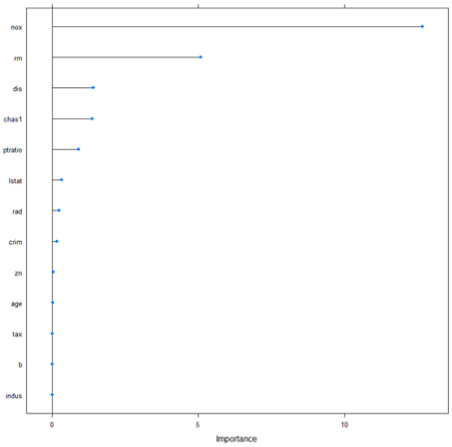

Figure 14: The significance of each variable in the model

We notice that three variables are enough to describe 60% of the data. When the coefficients are extremely limited and more than 80% of the variables have been described, it is almost certain that our model will create overfitting phenomenon.

This chart shows which variables are more important. We observe that our model considers only two variables highly significant (over 30%), while five variables are considered to be of no importance.

Elastic net regression

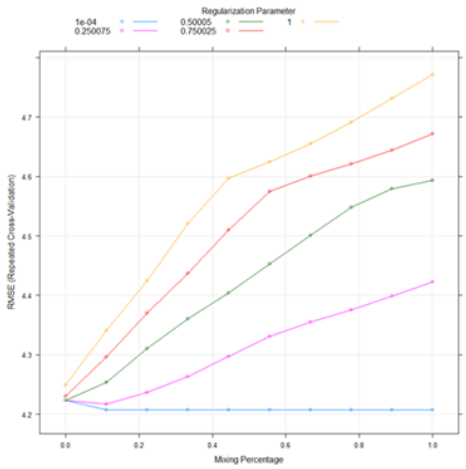

Figure 15: The root mean square error, as a function of the mixing ratio of the alpha parameter

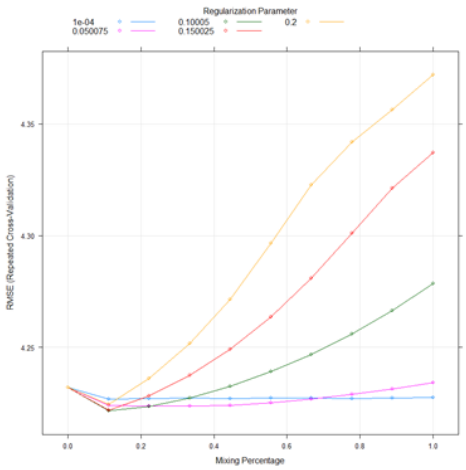

Figure 16: The root mean square error, as a function of the mixing ratio of the alpha parameter

In this diagram we see the RMSE value as a function of the mixing ratio of the alpha parameter. We note that the higher the normalization parameter gets, the higher the RMSE.

In this diagram we see the RMSE value as a function of the mixing ratio of the alpha parameter. We note that the higher the normalization parameter gets, the higher the RMSE. The maximum value now received is 0.2.

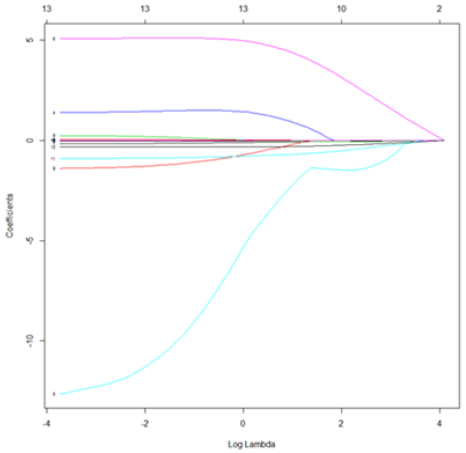

Figure 17: The coefficients of the variables as a function of the lambda logarithm

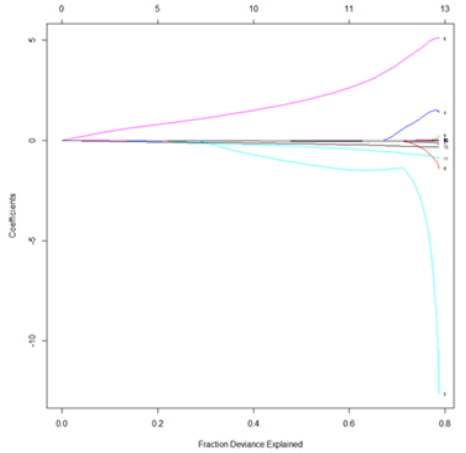

Figure 18: The coefficients of the variables as a function of the deviation of the fraction

For the log lambda = 4 we notice that the coefficients are zeroed, while for smaller values they begin to grow. The numbers in the top row show how many variables the model has in each case.

We notice that 11 variables are enough to describe 60% of the data. When the coefficients are extremely limited and more than 80% of the variables have been described, it is almost certain that our model will create overfitting phenomenon.

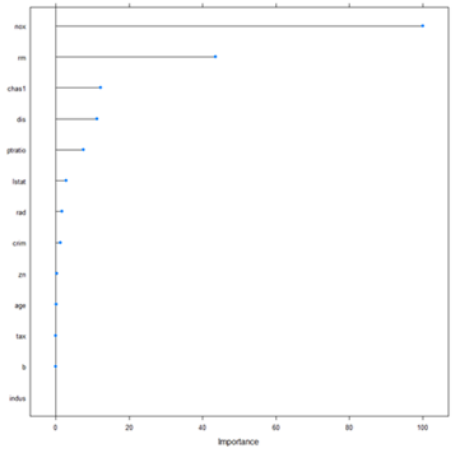

Figure 19: The significance of each variable in the model

This chart shows which variables are more important. We note that our model considers only two variables highly significant (over 30%), while six variables are considered to be of no importance

Comparison of models

The boxplot was used to compare the models.

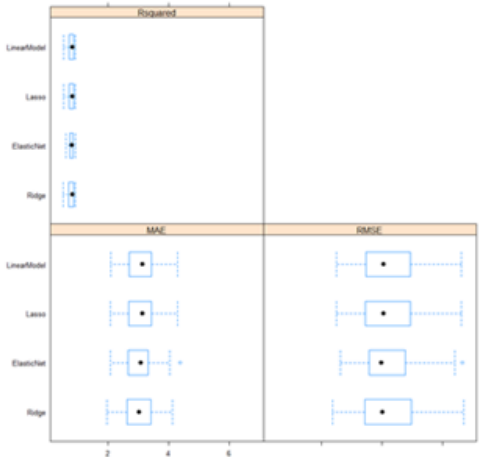

Figure 20: The boxplot for the four linear models

Model	RMSE	R^2	Training Time (minutes)	Memory (MB)
Linear	4.936135	0.8562100	5	11.4
Ridge	5.010936	0.8559697	6	13.7
Lasso	5.010936	0.8569684	5.7	12.2
ElasticNet	4.935745	0.8559697	6.3	14.1

Table 1: The results for the four linear models

We notice that the elastic regression has the lowest value compared to the rest of the models. However, because the models do not show significant differences, their difference is not clear, so we need to look at the console's results.

Looking at the results given by the console, we notice that the value recorded for the root mean square error in the elastic net regression is 4.935745, for the lasso regression is 5.010936, for the ridge regression is 5.010936 and for the linear is 4.936135. We note that the value noted for R^2 in the elastic net regression is 0.8559697, for the lasso regression is 0.8569684, for the ridge regression is 0.8559697 and for the linear is 0.8562100. Since the elastic net regression is lower than the other models, with respect to the root of the mean

square error and because it receives a higher value than the other models for R^2, we come to its choice as the most appropriate linear model for our data.

Prediction of the model

Finally, by predicting using the training data and the verification data, we calculate the root mean square error values for the two predictions.

```
> p1 <- predict(fm, train)
> sqrt(mean((train$medv-p1)^2))
[1] 4.113352
> p2 <- predict(fm, test)
> sqrt(mean((test$medv-p2)^2))
[1] 6.154483
```

Figure 21: The values of the root mean square error for the two predictions

Construction of Artificial Neural Networks

We then proceeded to construct and educate artificial neural networks. The procedure we followed was to divide the data into two random parts (not continuous) in a proportion of 75% -25%, where 75% was used at the training stage and 25% at the forecast stage. Then normalization was performed with the min-max method in the data to make the neural network perform better [26]. Since we are dealing with a regression problem, we will use the root mean square error (RMSE) as a measure of how far our forecasts are away from the actual data. We will use the determinant factor (Rsquared or R^2) as a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model.

In the first step, it is important to normalize our data before we begin to train the neural network. By avoiding the normalization of the data, we can lead to useless results or a very difficult training process where the algorithm will often not converge before the number of maximum repeats allowed.

There is no fixed rule on how many levels and neurons to use. Usually, if not necessary, a hidden level is enough for a huge number of applications. As for the number of neurons, it should be between the size of the input level and the output level respectively, usually 2/3 of the input size. When a large number of neurons are preferred to the hidden level, then there is a strong possibility that the network, after completing the training, will create overfitting phenomenon.

We set the threshold to 0.01. This is the maximum value that the error rate can reach at each iteration. If the error rate reaches the threshold, repetition stops.

Cross validation is another very important step in building prediction models. While there are different kinds of cross validation, the basic idea is to repeat the following procedure a series of time:

• Separation of data
• Implement the model
• Checking the model on the data
• Calculate the prediction error
• Repeat the K times procedure

With training, the neural network redefines the weights of the network edges. The training algorithm used for artificial neural networks is back propagation.

The goal of an artificial neural predictive network is generalization. Generalization is the phase that follows the training and learning of the network and in which the user gets the results given to him by the network for new input vectors, with which the network is not trained. By optimizing generalization, a user helps to avoid overfitting the values he gets as an output in the generalization phase [22].

Training and predicting process

In the neural network the linear regression model $\alpha_1 X_1 + \alpha_2 X_2 + \cdots + \alpha_{13} X_{13} = Y_1$ was simulated. The variables X_1 to X_{13} were used as input neurons and the Y_1 variable as an exit neuron. Learning the neural network is a process that has many parameters. The threshold for the rate of change of error, the maximum number of steps for a repetition, the number of repetitions of education and the learning rate [26].

All parameters in the default values were kept constant in the construction of artificial neural networks and only the number of iterations was changed. Specifically, the maximum error rate should be below 0.01 and the training steps should be at most 100000. 12 different technical neural networks were constructed [26].

During the training of each neural network, the training time and the size it occupies in memory are stored. These two values increased as the number of iterations increased. It was thought that the memory required by the neural network corresponds to the experiences of a human being [26]. At the predictive stage, the only parameter that existed is the number of iterations. Also, at this stage, the training time was saved, which was considered as the time available for a decision-maker [26].

The first neural network was trained with 10 repetitions and predicted with 10 repetitions. In each of the next 4 neural networks, the number of iterations of education doubled, while the number of iterations of the

forecast remained constant. The next 4 neural networks doubled the number of iterations of the forecast, while the number of iterations of the training remained constant. Having the same number of iterations of training and forecasting, the next 3 neural networks doubled both repetitions. There were a total of 12 neural networks with a different combination of repetitions for training and forecasts [26].

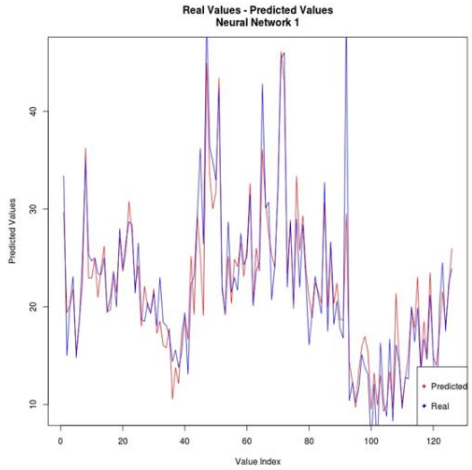

Figure 22: Predictive values to the actual values for the 1st neural network

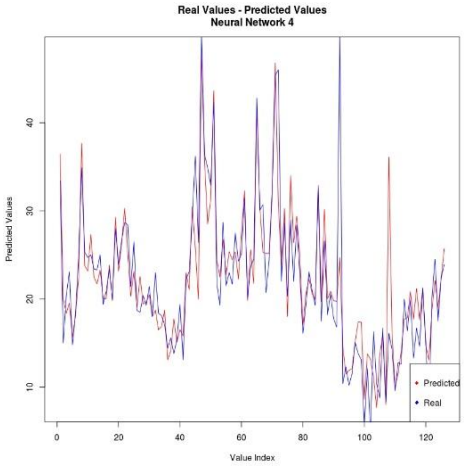

Figure 23: Predictive values to the actual values for the 4th neural network

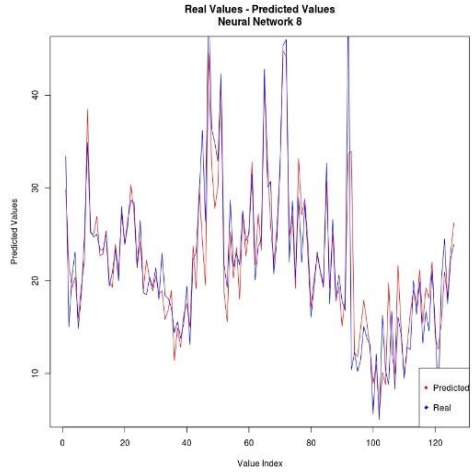

Figure 24: Predictive values to the actual values for the 8th neural network

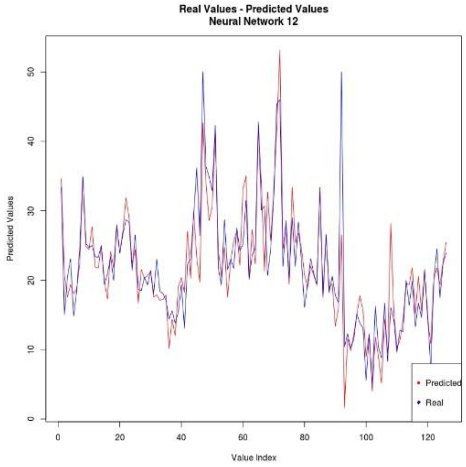

Figure 25: Predictive values to the actual values for the 12th neural network

The previous charts show the predictive values to the actual values. Because the prediction was made for 25% of the data and not for a single value, the root mean square error (RMSE) was used as the general error indicator. RMSE does not decrease as the neuronal memory increases, nor does the speed of training increase.

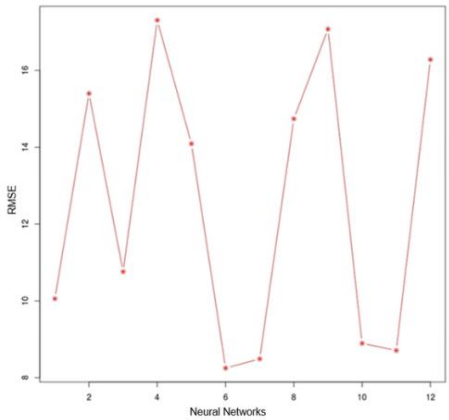

Figure 26: The RMSE in each neural network

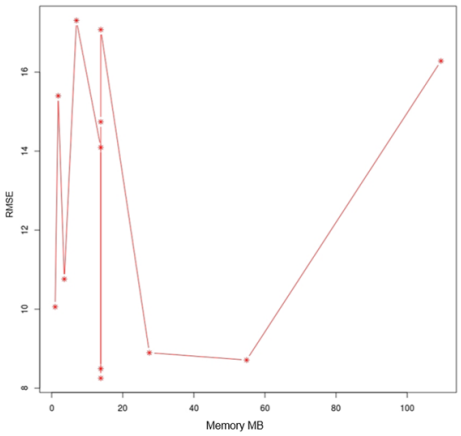

Figure 27: The RMSE compared with the neural network memory

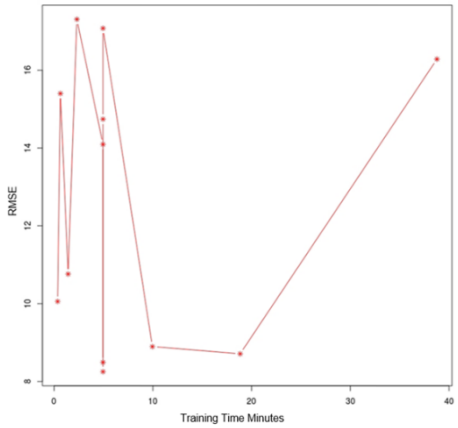

Figure 28: The RMSE compared with the training time of the neural network.

The RMSE does not decrease in neuronal progression, but it is observed that RMSE has the lowest value for the 6th neuronal, which had a number of repetitions with a number of repetitions of 8: 1 prediction. At this point, the overfitting phenomenon discussed above was observed. The network has "memorized" the results of the education phase but has not learned to generalize them correctly on the new entry data. This phenomenon brings about a relatively small error on the outputs resulting from the training of the network but a rather significant error in the outputs when given to the network new data as we see here [25].

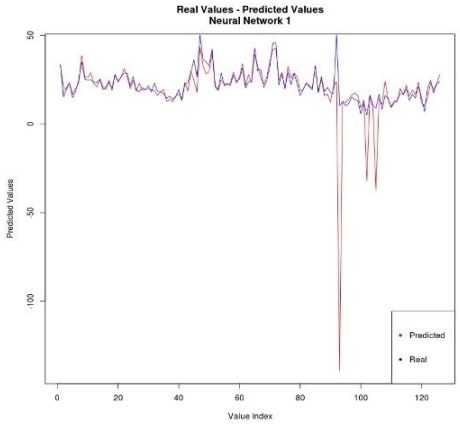

Figure 29: The actual values compared with the predicted values for the 1st neural network

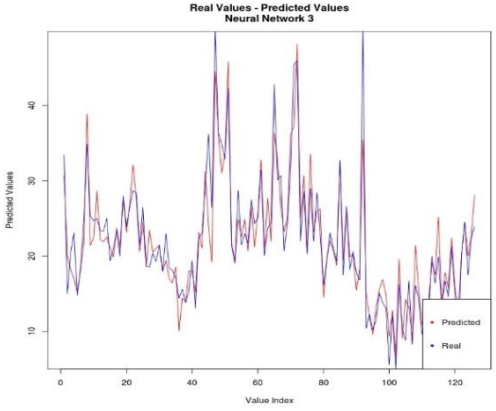

Figure 30: The actual values compared with the predicted values for the 3rd neural network

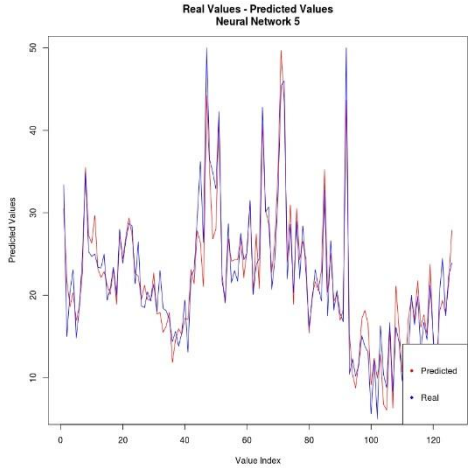

Figure 31: The actual values compared with the predicted values for the 5th neural network

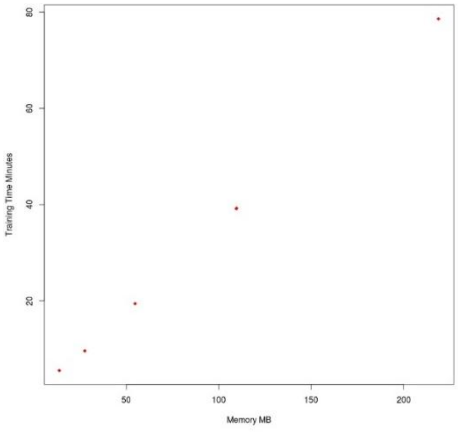

Figure 32: The memory of the neural network compared with the training time of it

There is an adjustment of the forecast to the actual value outside the extreme values.Note that there is a proportion of training and memory time.

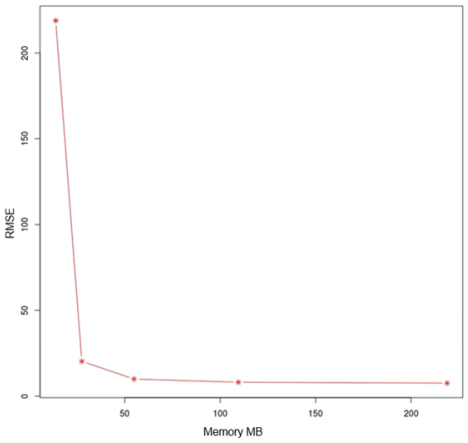

Figure 33: The RMSE of the neural network compared with the memory of it

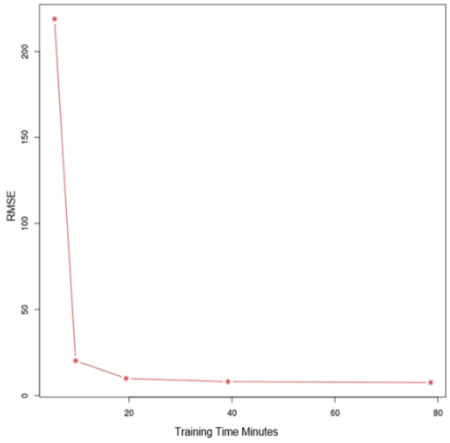

Figure 34: The RMSE of the neural network compared with the training time of it

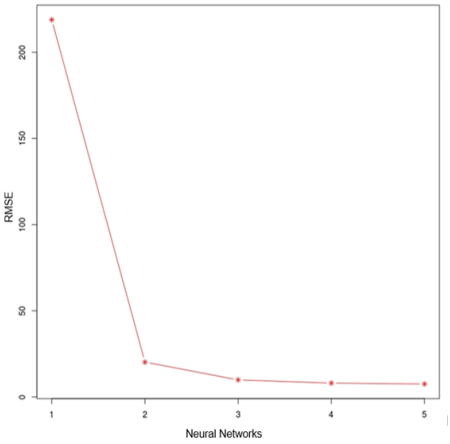

Figure 35: The RMSE of the neural networks

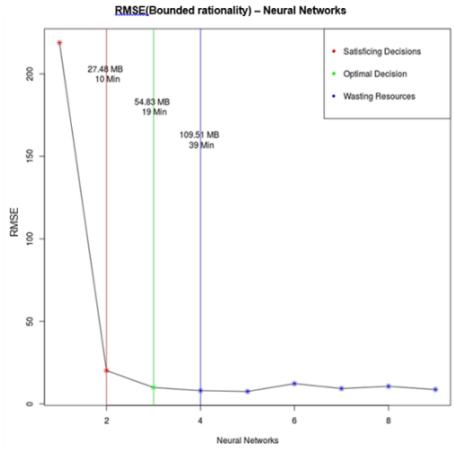

Figure 36: The RMSE as bounded rationality index for each neural network

The RMSE decreases as the neuronal memory increases, the training time, and the evolution of the neuronal. Bounded Rationality was considered as the RMSE. When the RMSE is equal to zero, the neural network will make a rational decision. Specifically, the third neural network required 54.83 MB of memory and 19Min of training time, with 640 training reps and 80 predictions repeats, to reach a rational decision. The fourth neural network required 179.58 MB of memory and 52Min training time, with 1934 iterations of training and 195 predictions, to reach a rational decision [26].

Model	RMSE	R^2	TrainingTime (minutes)	Memory (MB)
1st NN	8.011983	0.8852661	5.3	13.81
2nd NN	4.378397	0.9244428	10	27.49
3rd NN	4.077216	0.9186303	19	54.83
4th NN	4.209829	0.9415047	39	109.51
5th NN	4.351998	0.9057341	89	218.89
6th NN	4.159934	0.9097635	177	437.64
7th NN	4.001369	0.9005690	353	875.14
8th NN	4.172741	0.9080403	710	1.7GB

Model	RMSE	R^2	TrainingTime (minutes)	Memory (MB)
9^{th} NN	4.238851	0.9274631	1Day	3.6GB

Table 2: The values of RMSE and R^2 for the 9 "better" neural networks

Comparison of the elastic net regression with the artificial neural network

Observing the boxplot, we find that the neural network has a higher value for R^2, while a smaller value for the RMSE is compared with the elastic net regression. This observation is due to the fact that the neural network is better adapted to the data compared to the elastic net regression. However, because the models show small-scale differences, which, as shown in the figure, are not entirely clear, looking at the console's results, these differences are obvious.

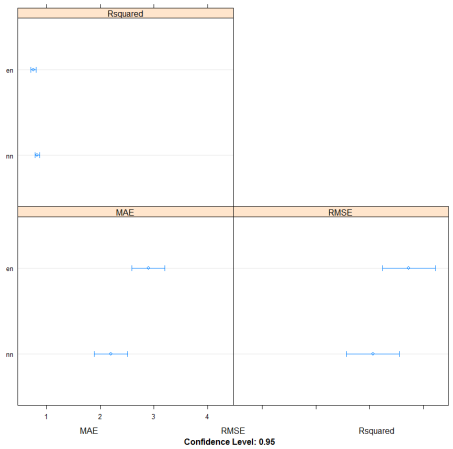

Figure 37: Boxplot for elastic net regression and neural network.

Model	RMSE	R^2	Training Time (minutes)	Memory (MB)	Comparison to the 3^{rd} NN (RMSE)	Comparison to the 3^{rd} NN (R^2)
ElasticNet	4.935745	0.8559697	6.3	14.1	+0.858529	-0.0626606
3^{rd} NN	4.077216	0.9186303	19	54.83	0	0

Table 3: Results for elastic net regression and neural network

Looking at the console's results, we notice that the value recorded for the root mean square error in the elastic net regression is 4.935745, while for the neural network the value is 4.077216. We note that the value noted for Rsquared in the elastic net regression is 0.8559697, while for the neural network the value is 0.9186303. Since the neural network has a lower value than the elastic net regression for the root mean square error and because it receives a higher value than the elastic net regression for R2, we come to the selection of the neural network as the most appropriate model for our data.

IV. Conclusion

Summarizing, after the construction of the four linear models and the twelve neural networks, in order to predict the average price of real estate homes on the outskirts of Boston, it is concluded that both linear models and neural networks are capable of receiving a satisfactory but also rational decision.

Specifically, with regard to linear models, it is observed that they do not differ greatly, as we see that the difference in the root mean square error is in the second decimal place. Similarly, the same behavior is observed in the determination factor. The differences in question are negligible, which does not lead to the rejection of the linear model based solely on these values.

The advantage of linear models is that they reduce the regression coefficients near zero by keeping the variables, thus avoiding overfitting.

Conversely, neural networks do not avoid overfitting (using the present library). In order to correct the problem of overfitting in the neural network, the number of repetitions of training with a number of predictions was maintained and the neural network was trained from the beginning. One way to avoid over-training in

neural networks is to reduce the number of hidden nodes or reduce the number of hidden neurons. However, this action can bring about significant changes in the results from the neural network, possibly and unlike what is desired.

In neural networks it is observed that as the memory and speed increases, without the analogy being compromised, the results are improved, which confirms the limitation of limited rationality. With limited memory and time, a decision of limited rationality is taken, and as the memory and time increases, the answer improves.

It is observed that in the model that was implemented, after the third neural network, the mean square error index remains at the same level as if the neural network resources increased. It is believed that the rational decision has been taken from the third neural network and all subsequent neural networks continue to make the same decision, "wasting" more resources. This is perceived by the time, memory size, and the number of repetitions that the next neural networks need in order to reach a rational decision. The equation of the mean square error with zero would mean the perfect fit of the model to the data. So the decision that would end up with the model would be ideal.

The resulting conclusion is that the first two neural networks are those that have taken a satisfactory decision, while the third neuron is what makes the rational decision.

From the comparison of the elastic net regression (as the best linear model) with the third neural network (as the rational neural network), it is concluded that the two models show a difference in the root mean square error value of almost one unit, marginally significant difference. The difference between the two models for the determinant factor is very small, leading marginally to the choice of the neural model as the best model for prediction.

Model	**RMSE**	**R^2**	**Training Time (minutes)**	**Memory (MB)**	**Comparison to the 3rd NN (RMSE)**	**Comparison to the 3rd NN (R^2)**
Linear	4.936135	0.8562100	5	11.4	+0.858919	-0.0624203
Ridge	5.010936	0.8559697	6	13.7	+0.933720	-0.0626606
Lasso	5.010936	0.8569684	5.7	12.2	+0.933720	-0.0616619
Elastic Net	4.935745	0.8559697	6.3	14.1	+0.858529	-0.0626606
1st NN	8.011983	0.8852661	5.3	13.81	+3.934767	-0.0333642
2nd NN	4.378397	0.9244428	10	27.49	+0.301181	+0.0058125
3rd NN	4.077216	0.9186303	19	54.83	0	0
4th NN	4.209829	0.9415047	39	109.51	+0.132613	+0.0228744
5th NN	4.351998	0.9057341	89	218.89	+0.274782	-0.0128962
6th NN	4.159934	0.9097635	177	437.64	+0.082718	-0.0088668
7th NN	4.001369	0.9005690	353	875.14	-0.075847	-0.0180613
8th NN	4.172741	0.9080403	710	1.7GB	+0.095525	-0.0105900
9th NN	4.238851	0.9274631	1Day	3.6GB	+0.161635	+0.0088328

Table 4: The results for all models

Adaptation of the models to these data, as well as the results obtained from the models are not decisive. In different types of data, such as e.g. medical data, environmental data, etc. or data described by fewer variables, the adaptation of these models and the results to which they may be may be completely different.

References

[1]. P. Argyrakis, (2001), Neural Networks and Applies, Greek Open University , pages 192
[2]. D. Graupe, (2013), Principles of Artificial Neural Networks.3rd Edition, World Scientific Publishers,
[3]. Auer, Peter; Harald Burgsteiner; Wolfgang Maass (2008). "A learning rule for very simple universal approximators consisting of a single layer of perceptrons". Neural Networks. 21 (5): 786–795. doi:10.1016/j.neunet.2007.12.036.
[4]. https://en.wikipedia.org/wiki/Feedforward_neural_network (date of access: 1/10/2018)
[5]. Simon, Herbert A. (1997). Administrative Behavior: a Study of Decision-Making Processes in Administrative Organizations (4th ed.). New York: Free Press.
[6]. David Rettinger, Reid Hastie, (2001), Content Effects on Decision Making, Organizational Behavior and Human Decision Processes 85(2), pp 336-359

[7]. Simon, Herbert A. (1976). Administrative Behavior: a Study of Decision-Making Processes in Administrative Organizations (3rd ed.). New York: Free Press. pp 81
[8]. James March, Herbert Simon, (1993), Organizations, 2nd edn Oxford
[9]. H. A. Simon (1973), "The Structure of Ill-Structured Problems", Artificial Intelligence, 4: 181-201.
[10]. March, James G. (2006). "Rationality, Foolishness, and Adaptive Intelligence". Strategic Management Journal. 27: pp 201–214
[11]. https://cran.r-project.org/web/packages/MASS/index.html(date of access 1/10/2018)
[12]. https://cran.r-project.org/web/packages/caret/caret.pdf (date of access 1/10/2018)
[13]. https://cran.r-project.org/web/packages/glmnet/glmnet.pdf (date of access 1/10/2018)
[14]. https://cran.r-project.org/web/packages/mlbench/mlbench.pdf (date of access 1/10/2018)
[15]. https://cran.r-project.org/web/packages/psych/psych.pdf (date of access 1/10/2018)
[16]. https://cran.r-project.org/web/packages/neuralnet/index.html (date of access 1/10/2018)
[17]. https://en.wikipedia.org/wiki/Overfitting (date of access 7/10/2018)
[18]. https://en.wikipedia.org/wiki/Tikhonov_regularization (date of access 7/10/2018)
[19]. https://en.wikipedia.org/wiki/Lasso_(statistics) (date of access 7/10/2018)
[20]. https://en.wikipedia.org/wiki/Elastic_net_regularization (date of access 7/10/2018)
[21]. https://en.wikipedia.org/wiki/Cross-validation_(statistics) (date of access 7/10/2018)
[22]. https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/ (date of access 15/3/2018)
[23]. Burnham, K. P.; Anderson, D. R. (2002), Model Selection and Multimodel Inference (2nd ed.), Springer-Verlag
[24]. https://papers.nips.cc/paper/1895-overfitting-in-neural-nets-backpropagation-conjugate-gradient-and-early-stopping.pdf (date of access 25-07-2018)
[25]. Tzotzis Christos, KalampakasArgyrios, Makris C. Georgios, (2018), Examining the Limited Rational Decisions of Neural Networks, IOSR Journal of Mathematics (IOSR-JM), pp 50-57.

Tzotzis Christos." Comparative study of linear models and neural networks for rational decision making." IOSR Journal of Mathematics (IOSR-JM) 14.6 (2018): 46-60.