Production System Modelling
MEC-E7001

Assignment 2

Data fitting

Name: Nguyen Xuan Binh

Student ID: 887799

Mail: binh.nguyen@aalto.fi

Date: 09.02.2024

1. INTRODUCTION

Purpose of the report: This assignment uses two techniques, linear regression and neural networks, to determine how much it costs to making anodized extruded aluminum profiles. Making these aluminum metal pieces involves a lot of processes such as using machines and people to move the materials around. Additionally, we also know that the costs are connected to the thickness, perimeter and weight of this coating.

The data used in this report is obtained from a manufacturing plant with a variety of aluminum profiles. By applying regression on the data, the report tries to assign anodizing costs to each product. Through this approach, we try to build a reliable model for cost prediction that can be helpful in the financial planning of making anodized aluminum profiles.

2. PROBLEM AND MODEL

Modeling method and software

- Metamodelling is fitting a model with linear regression or neural network) to data obtained using another model (simulation)
- Linear regression is done by Excel Data Analysis Toolpak
- Neural network regression is done by the Matlab nftool package. Actually, neural networks' industry-standard implementation is in Python. However, for the sake of this assignment only, I will use Matlab for building and training neural networks.
- The features are Thickness (micrometer), profile Perimeter (relates to coated area in mm), and Weight (kg). The label is the cost of the anodizing aluminum profile, measured in euro/m2. There are 40 data points in total
- For regression, we apply regression on both original data and standardized (both features and labels) data.

Model description

There is no need to re-explain the Multilinear Regression and Fully Connected Neural Networks (FCNN) as they are very popular. Here are some good online explanation for these two models

Multilinear regression: https://statsandr.com/blog/multiple-linear-regression-made-simple/

Neural networks:
https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html

3. EXPERIMENT

a. The used values in this modeling assignment are reported below

• Model features are the factors that we varied in simulation experiments, while label is the cost. Here are the first 5 datapoints

Thickness	Perimeter	Weight	Cost
5	20	0.09	5.6
10	36	0.2	5.2
15	65	0.6	5.4
20	117	1.6	5.4
5	210	3.9	4.1

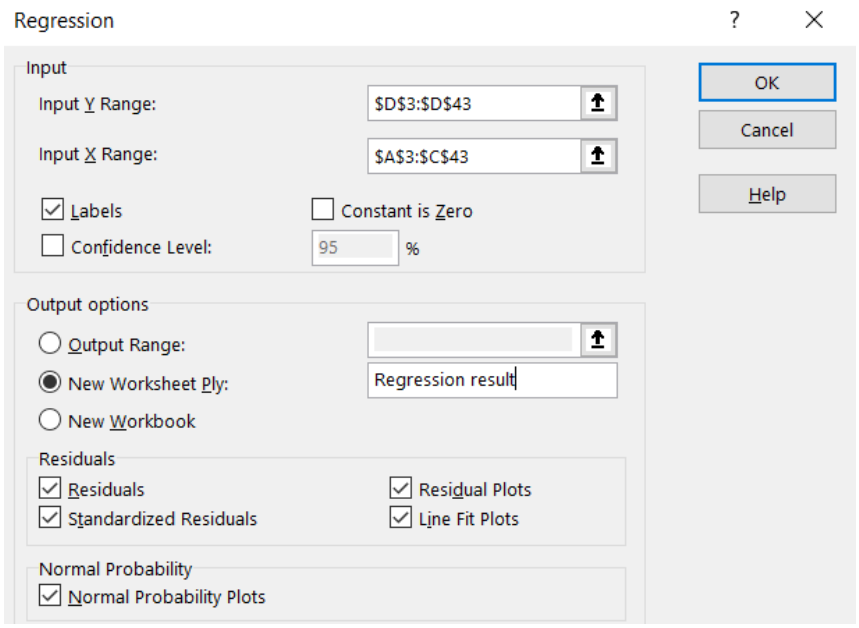
b. Model testing

The regression model is tested under different thickness, perimeter and weights to predict the costs of anodizing the extruded aluminum profiles.

c. Modeling results

1. For the regression analysis, analyze the performance of the model based on the Excel Regression report KPIs. Then, make conclusions based on the regression analysis with standardized data

These are the configurations in fitting the regression model in Excel



Labels box is ticked to ensure that the column names are counted and the true data starts from the second row. All output data of residuals and normal probability are outputted in another sheet. The regression results are obtained from the standardized data.

	Coefficients	Standard Error	Lower 95%	Upper 95%
Intercept	3.13147E-18	0.118224134	-0.239769656	0.239769656
Thickness	0.27453502	0.122533365	0.026025838	0.523044202
Perimeter	-1.368670503	0.282513081	-1.941633587	-0.795707419
Weight	1.130921913	0.280169493	0.562711844	1.699131981

This table above presents the results from a regression analysis. It shows the estimated coefficients for each predictor variable, their standard errors, and the 95% confidence intervals for the coefficients. Now we can look at the interpretation for each term

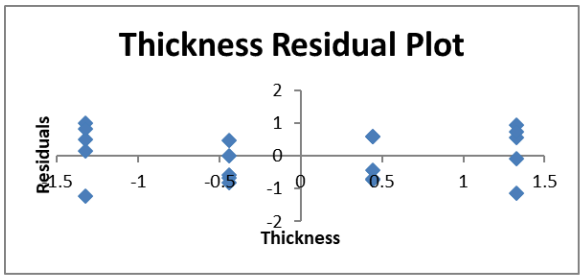
- **Intercept**: Since the confidence interval includes zero, this suggests that there is no significant intercept term for the model. It means that if there is no anodizing at all, then the predicted cost would be 0 euros

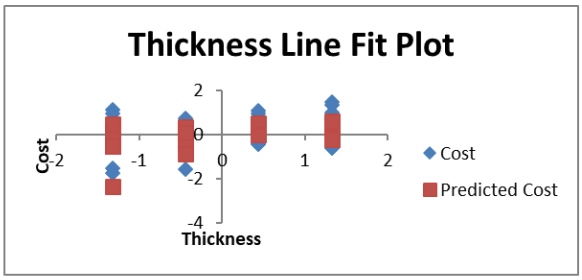
- **Thickness:** For each unit increase in thickness, the cost increases about 0.275 standardized euro unit. The 95% confidence interval ranges from 0.026 to 0.523, which suggests that thickness has a significant positive effect on the cost

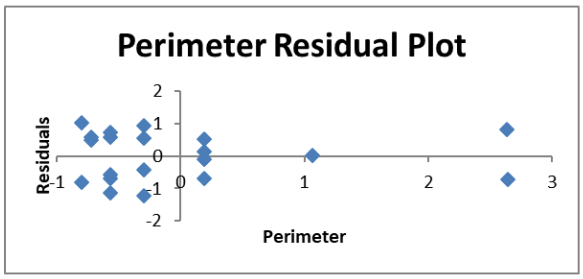
- **Perimeter:** For each unit increase in perimeter, the cost increases about 1.369 standardized euro unit. The 95% confidence interval ranges from -1.941 to -0.796, which suggests that perimeter has a significant negative effect on the cost

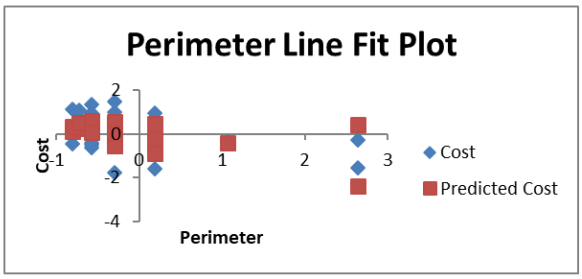
- **Weight:** For each unit increase in weight, the cost increases about 1.131 standardized euro unit. The 95% confidence interval ranges from 0.563 to 1.699, which suggests that weight has a significant positive effect on the cost, holding all else constant.

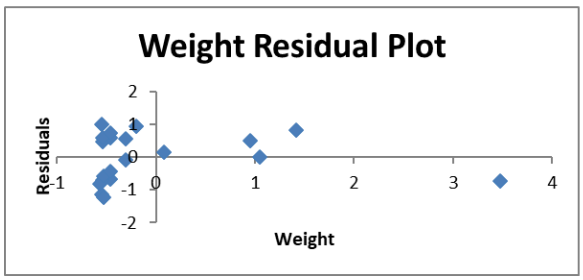
We can also look at the graphs generated by Excel KPI to analyze the regression model

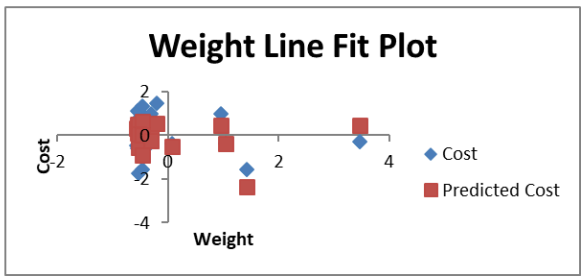












Above are the residual plots and line fit plots for thickness, perimeter, and weight w.r.t cost.

- **Residual plots:**

These plots record the residuals, which are the differences between observed and predicted values of the regression model. If the regression model is appropriate, the residuals should be randomly scattered around the horizontal axis, indicating that the model's predictions are unbiased at different levels of the independent variables.

For all three residual plots, we can observe that the residuals appear to be randomly distributed across different values, with no clear pattern. This suggests that the model's predictions for all features are generally unbiased.

- **Line fit plots:**

These plots compare the actual costs to the predicted costs from the regression model. The closer the points are to the line, the better the model fits the data.

From all the line fit plots, we can see all three features usually underpredict or overpredict the costs, as the range of cost prediction is usually contained inside the range of actual costs, suggesting the regression model does not manage to capture the variance in actual costs.

Finally, we have the regression statistics

Regression Statistics	
Multiple R	0.695649449
R Square	0.483928156
Adjusted R Square	0.440922169
Standard Error	0.747715073
Observations	40

Multiple R (Correlation Coefficient) is sqrt(R Square), which is the correlation between the observed and predicted values of the dependent variable. Multiple R of 0.696 means a moderate to strong positive correlation.

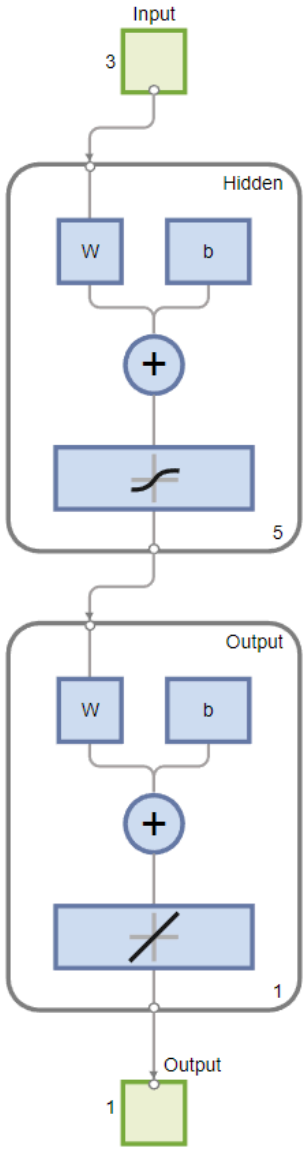
R Square of 48.39% means the variability in 3 features can be explained by the cost in this regression model. The closer this value is to 1, the more variability is explained by the model.

Adjusted R Square of 44.09% means that after accounting for the number of variables, around 44.09% of the variability is explained, which is less than the R square value but still significant

Standard error is standard deviation of the regression's residuals. On average, the actual costs are about 0.7477 euros away from the predicted costs.

Observations: 40 datapoints as we discussed earlier

2. For neural network results, report R2 and Standard Error. You can calculate these the same way as we did “manually” using Excel in the exercise – see RegressionAnalysisExample.xls in the course material



Neural Network Training (09-Feb-2024 15:28:59)

Network Diagram

Training Results

Training finished: Met validation criterion

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	9	1000
Elapsed Time	-	00:00:00	-
Performance	1.22	0.334	0
Gradient	2.89	0.104	1e-07
Mu	0.001	0.01	1e+10
Validation Checks	0	6	6

Training Algorithms

Data Division: Random dividerand
Training: Levenberg-Marquardt trainlm
Performance: Mean Squared Error mse
Calculations: MEX

Training Plots

Performance | Training State
Error Histogram | Regression
Fit

Here is the printed message from MATLAB, which can be used to debug the neural network

```
Sum of Squared Errors (SSE) for all dataset:
    4.8625
Preprocessing function
    {'mapstd'}
Postprocessing function
    {'mapstd'}
Weights for the input - hidden layer
   -0.8854   -1.7352    7.4934
    2.0113   -3.1197    2.3130
   -2.5053    1.6550   -1.6085
   -0.6865    0.9253   -0.4141
    2.1619   -3.1939    3.4794
Bias for the input - hidden layer
    4.8089
   -1.1485
   -1.7450
   -3.0157
    3.2808
Weights for the hidden - output layer
    3.8282    1.3606    1.4988    2.1125    2.5167
Bias for the hidden - output layer
   -2.4777
inputStandardized = 3×40
     -1.3248   -0.4416    0.4416    1.3248   -1.3248   -0.4416    0.4416    1.3248 ...
     -0.8032   -0.7198   -0.5697   -0.2994    0.1872    1.0629    2.6393   -0.5697
     -0.5510   -0.5328   -0.4625   -0.3075    0.0802    1.0494    3.4723   -0.5494
```

The hidden layer values are

119	2.6384	12.8191	2.3841	1.6743	1.1586	12.0373	2.2577	1.4715	1.0073
812	1.9769	-2.1868	-0.8301	0.7529	2.2234	-8.7607	-1.4728	-0.3962	0.2210
354	-5.2314	0.3466	-1.0645	-3.1856	-5.2628	3.6574	-0.7378	-2.6028	-4.2597
243	-4.1167	-2.3287	-3.2232	-3.7642	-4.2606	-0.2530	-3.0224	-3.4043	-3.6246
977	6.3921	3.1439	2.9370	4.6806	6.3550	-3.0714	2.3204	3.5823	4.4773

The hidden layer values after Tanh are

879	0.9898	1.0000	0.9832	0.9321	0.8206	1.0000	0.9784	0.8999	0.7646
740	0.9624	-0.9751	-0.6805	0.6369	0.9768	-1.0000	-0.9001	-0.3767	0.2175
944	-0.9999	0.3334	-0.7874	-0.9966	-0.9999	0.9987	-0.6278	-0.9891	-0.9996
986	-0.9995	-0.9812	-0.9968	-0.9989	-0.9996	-0.2477	-0.9953	-0.9978	-0.9986
995	1.0000	0.9963	0.9944	0.9998	1.0000	-0.9957	0.9809	0.9985	0.9997

The output layer (standardized) values are

1.1016	-0.4867	0.6806	1.3520	-0.3829	-0.3413	-0.2904	-0.4651	-1.7418

Outputs of neural network prediction (De-standardized) - Manual

5.5619	3.9733	5.1409	5.8123	4.0772	4.1187	4.1697	3.9950	2.7180

Outputs of neural network prediction (De-standardized) - Automatic

5.5616	3.9735	5.1407	5.8120	4.0773	4.1188	4.1698	3.9951	2.7185

We can see that the manual calculation of feed forward matrices and the automatic solution by calling the neural network prediction are the same. Additionally, I have fixed the preprocessing and postprocessing function of MATLAB neural network toolbox as standardization (mapstd), instead of normalization (mapminmax) to match with our current study case.

For more information on the preprocessing and postprocessing functions of MATLAB:

https://se.mathworks.com/help/deeplearning/ug/choose-neural-network-input-output-processing-functions.html

The diagnostics of neural networks is as follows:

R square value: 0.8747

SSE: 4.886552, TSS: 39

Standard error: 0.152365

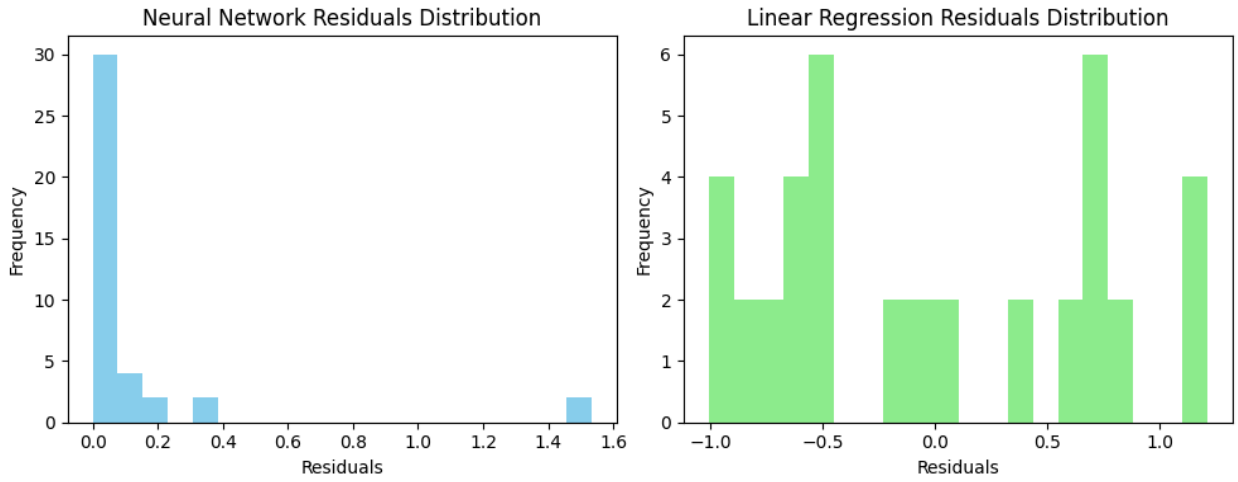
3. Compare the R2 and standard error for both models. You can even list the errors of the models' outputs and the target values and analyze their distribution, min, and max values

Comparison of R2 and standard error

	Linear regression	Neural networks
R2	0.4839	0.8747
Standard error	0.7477	0.1523
Min residual	0	0.0003502
Max residual	1.533972	1.4709506

From the comparison, we can conclude that neural networks far outperform linear regression.

Additionally, here are the distribution graph of the residuals of the two models



We can see that the error distribution of the neural network is tightly concentrated near 0 - 0.2, while the error distribution of linear regression is uniformly distributed along [-1, 1], suggesting that on average, the neural network regression has better prediction than linear regression.

4. ANALYSIS

Two particularly important findings are revealed by analyzing the modeling methodologies used to estimate the anodizing costs of extruded aluminum profiles. First off, it's clear that neural networks outperform linear regression in building relationships between the three features and the cost. The R2 value of 0.8747 for the neural network model was much higher than the R2 value of 0.4839 for the linear regression model. This difference emphasizes how the neural network can capture features' interactions and nonlinear patterns that linear regression is unable to account for, such as thickness, perimeter, and weight.

Secondly, the neural network's advantage is also proven by the standard error measure, which shows a noticeably lower value of 0.1523 compared to 0.7477 from linear regression. This metric shows that the neural network's predictions are more accurate and dependable because they are generally closer to the actual prices. The neural network's residual distribution, which is mainly centered around zero, is in contrast to linear regression's more widely distributed pattern, showing the neural network's better performance in this 40 point dataset. The performance difference between the two models emphasizes how crucial it is to select the correct hypothesis.

5. CONCLUSION

a. General and specific conclusions

When predicting anodizing expenses, neural networks outperform linear regression in terms of accuracy and flexibility to diverse data patterns. This result shows how limited linear regression is when addressing nonlinear interactions between variables.

b. Practical value of this assignment

This assignment demonstrates how neural networks can improve manufacturing's operational efficiency and budgetary planning. Increased cost prediction accuracy may result in substantial expense reductions and better decision-making.

c. Reliability of results

The neural network model's performance measures demonstrate its dependability, which implies it has the potential to be a reliable cost prediction tool. The total reliability is still dependent on the caliber of the training data, though.