Mladá veda
Young Science
ISSN 1339-3189

Vol. 12 (2), pp. 46-58

PREDIKCIA A URČOVANIE CIEN CESTNÝCH VOZIDIEL: PRVOTNÝ KONCEPT

FORECASTING AND PRICING OF ROAD VEHICLES: AN INITIAL CONCEPT

Natália Kováčová, Patrik Veľký [1]

Natália Kováčová pôsobí ako interná doktorandka na Fakulte prevádzky a ekonomiky dopravy a spojov na Žilinskej univerzite v Žiline. Vo svojom výskume sa venuje analýze emisií cestných vozidiel. Patrik Veľký pôsobí ako interný doktorand na Fakulte prevádzky a ekonomiky dopravy a spojov na Žilinskej univerzite v Žiline. Vo svojom výskume sa venuje vývoji neurónových sietí v rôznych aplikáciách.

Natália Kováčová works as an internal PhD student at the Faculty of Operations and Economics of Transport and Communications at the University of Žilina. Her research focuses on the analysis of road vehicle emissions. Patrik Veľký works as an internal PhD student at the Faculty of Operations and Economics of Transport and Communications at the University of Žilina. His research is focused on the development of neural networks in various applications.

Abstract

This paper devotes its attention to the increasingly advanced development of machine learning and the possibility of applying such learning in the field of road transportation. Sales of road vehicles form a significant part of a country's economy and the work of pricing them has a major impact on the purchase or sale. Used vehicles are still the first choice for a large part of the population, as the purchase of a new vehicle results in an immediate drop in price. Determining the right price for a used car is crucial for a proper purchase and a quick sale. Some vehicle parameters affect the price of a vehicle more and some less. This paper highlights the need for selection of such parameters and describes the development of a system for determining the price of vehicles based on these parameters. The system has the potential to serve as a determinant of whether the purchase or sale of a vehicle is advantageous in terms of vehicle price. Also, such a system serves as a predictor of price development, which helps to regulate the economy to ensure market sustainability, proper economics and the possibility of early action if necessary.

Key words: prediction, pricing, vehicles, neural network, data analysis

[1] Adresa pracoviska: Ing. Natália Kováčová, Ing. Patrik Veľký, Žilinská univerzita v Žiline, Fakulta prevádzky a ekonomiky dopravy a spojov, Univerzitná 8215/1, 01026 Žilina
E-mail: natalia.kovacova@stud.uniza.sk, velky2@stud.uniza.sk

Mladá veda
Young Science
ISSN 1339-3189

Vol. 12 (2), pp. 46-58

Abstrakt

Tento článok venuje svoju pozornosť stále pokročilejšiemu vývoju strojového učenia a možnosti aplikácie takéhoto učenia v oblasti cestnej dopravy. Predaj cestných vozidiel tvorí značnú časť ekonomiky krajín a práce tvorba ich cien má zásadný vplyv na kúpu resp. predaj. Ojazdené vozidlá sú stále pre veľkú časť obyvateľstva prvou voľbou, nakoľko kúpou nového vozidla dochádza k okamžitému poklesu ceny. A práve určenie správnej ceny ojazdeného vozidla je kľúčové pre správnu kúpu a rýchly predaj. Niektoré parametre vozidla ovplyvňujú cenu vozidla viac a niektoré menej. Tento článok poukazuje na potrebu selekcie takýchto parametrov a popisuje vývoj systému pre určovanie ceny vozidiel na základe týchto parametrov. Systém má potenciál slúžiť ako prvok určujúci, či kúpa alebo predaj vozidla je výhodná z hľadiska ceny vozidla. Takisto, takýto systém slúži ako prediktor vývoju cien, čo pomáha k regulácii ekonomiky k zabezpečeniu udržateľnosti trhu, správnej hospodárnosti a k možnosti predčasného konania v prípade potreby.

Kľúčové slová: predikcia, určovanie cien, vozidlá, neurónová sieť, analýza dát

Introduction

In today's digital age, where an abundance of data is available online and technological advances in machine learning are constantly improving, vehicle price prediction based on analytical models is an increasingly desirable field in the automotive industry. Accurate estimates of vehicle prices are crucial for consumers when deciding whether to buy or sell a vehicle, as well as for automotive dealers and analysts when determining optimal pricing and making market forecasts.

There is currently a trend in buying and selling used vehicles, but how do you make the right decision on the price of a car? Will the purchase be profitable? How much will the price of the car drop in 2 years? Which parameter of the car affects the price the most? All these questions can be answered with a well-trained neural network.

In their paper ((Prashant Gajera, et al., 2021) they present 5 basic models using machine learning. However, in this case it is not deep learning and hence neural network formation. Nevertheless, they used machine learning tools and the results were interesting. The highest success rate was when a model called Random Forest Regressor was used, where the accuracy reached 93%. On the other hand, the lowest accuracy was achieved by using the so-called KNN-Regressor at 69%. They used a database of over 90,000 vehicles for their models.

An expert text (Enis Gegic, et al., 2019) published in this direction used straight machine learning techniques, but not using neural networks. Their most accurate model achieved an accuracy of 87%. Another paper (Samruddhi and Ashok Kumar, 2020) reported similar techniques achieving accuracy of 71% and 85% respectively.

The use of deep learning technique appears to be beneficial in the work of (Pillai, 2022) as the neural network model developed by them achieves 96% accuracy on 140,000 vehicle samples. Another feat in the world of vehicle price prediction using neural networks is the work of (Al-Turjman et al., 2022), which reports an accuracy of 90%. The paper (Varshitha et al., 2022) compares different machine learning techniques and specifically both

random forest and neural network. The accuracy results speak in favour of random forest in the ratio of 91% to 82%.

In the book (Chollet, 2021), the author, who is the top representative of one of the most famous libraries for deep learning, states that random forests are great models until the amount of data starts to increase rapidly. The moment you cross an undefined threshold in the amount of data, neural networks start to dominate.

In this article, the work focuses specifically on neural networks and their ability to find very complex patterns between values. They also provide the potential use of robust databases, complex analysis of results, robustness to noise, and more.

The goal is to create a basic model that can predict the price of vehicles based on input parameters, and do so in a short amount of time and with a relatively small database. This basic framework will serve as a basis for further optimization in building a system capable of predicting the price of a vehicle to the greatest extent possible. Such a process involves a large amount of work such as data analysis, data collection, network training and much more. Therefore, it is necessary to start with a small network and gradually expand its capabilities. The resulting baseline network should achieve accuracy based on the literature review, where a level of 80% was required in a large body of professional work. The goal is to achieve this level with a minimal amount of data and a small scale neural network. Another goal is to highlight not only the need for an efficient neural network, but also the need for activities such as understanding how the network works, data analysis, experimentation, and experience in this kind of science.

Methods and Methodology

The creation of a system that can predict the price of vehicles is subject to several steps, which correspond to the selection of the database, identification of relevant data and its processing, creation of a neural network, training of the neural network and evaluation of the results. The last step is the optimization of the system. Each of these steps is important for the creation of a functional system, so it is necessary to analyse all these aspects in detail. To understand the issues, extensive knowledge in machine learning is required, which includes statistics, mathematics, programming, data analysis but also creativity. The whole process of creation is time consuming, as in-depth analysis or training a neural network takes hours in terms of time. However, the goal is to create a basic system that can evaluate the price of a vehicle with an accuracy of more than 80%. Based on this model, an optimization process will later be created that will gradually increase this accuracy figure.

Database

In order to build a model that can predict vehicle prices, it is necessary to obtain a database that contains information about vehicles and their prices. This vehicle information may be of different nature. The most common influences on the price of a vehicle are its age, performance, make, technical condition and others. However, there are also other data in the databases such as torque, mileage and many others. The deeper the database, the more information can be fed into the neural network and the more accurate the results can be. However, some of the data may cause the amount of data to greatly increase the training time

and the accuracy result of the neural network may be the same or lower than the result without the data. Therefore, it is necessary to analyse the database and select the relevant data, or use the "grid search" or "random search" method. These methods are described in a later section. The database for this purpose contains 1000 data on vehicle parameters which include:
on road old, on road now, years, km, rating, condition, economy, top speed, hp, torque, current price.

Some of these parameters are directly correlated with the current price of the vehicle and some have no effect on the price of the vehicle. The parameters on road old and on road now are not explained in the database, but their exact meaning is not necessary since the neural network does not work with the meaning of the terms but with their values. The term years denotes the age of the vehicle, km denotes the number of kilometres driven, rating expresses numerically from 0 to 5 the rating of the vehicle, where 0 is the worst and 5 is the best, condition expresses the technical condition of the vehicle from 0 to 10, where 10 is the best, economy expresses the consumption of the vehicle from 7.5 to 15. Units of consumption are not needed in this case. The Top Speed parameter expresses the maximum possible speed of the vehicle, hp expresses the power of the vehicle, torque is the torque of the vehicle and current price expresses the current price of the vehicle based on all defined vehicle parameters. Any units for the given parameters are not necessary, since the neural network looks for patterns between the values, without regard to the units.

The given database contains sufficient data to create a basic model for vehicle price prediction. For their further processing, an analysis of the individual data is necessary for the correct selection of the input parameters to the neural network.

Identification of relevant data and their processing

The identification of relevant data consists in the analysis of vehicle parameters, their combinations and the vehicle price parameter. Such an analysis will help to select data that will allow the network to learn relatively quickly and accurately how to identify the price of a vehicle based on this data. For data analysis, Python programming language is used along with Seaborn library to create the diagrams between each parameter. For such visualization a grid of scatterplots for pairwise combinations of variables along with histograms or kernel density estimates along the diagonal is used. Each scatterplot in the grid shows the relationship between two variables, with one variable plotted on the x-axis and the other variable plotted on the y-axis. Kernel density estimation (KDE) is a non-parametric method for estimating the probability density function of a random variable. In the context of pairplots, KDE plots provide a smooth estimate of the distribution of values for each variable along the diagonal.

After performing a comprehensive analysis, Figure 1 shows a diagram of the correlations between the current price of a vehicle and such parameters that have been evaluated as unsuitable for a neural network. Using such an interpretation, it can be seen that there is no direct correlation between the vehicle price and the parameters on road old, on road now, top speed, torque, hp. Such a result is surprising since the expectation of top speed and power was that they directly affect the price of the vehicle. However, according to this database, this is not the case as can be seen in Figure 1. However, it should be mentioned that

such an analysis is not binding for all vehicle databases, it only shows the way of data selection and must be done individually for each database.

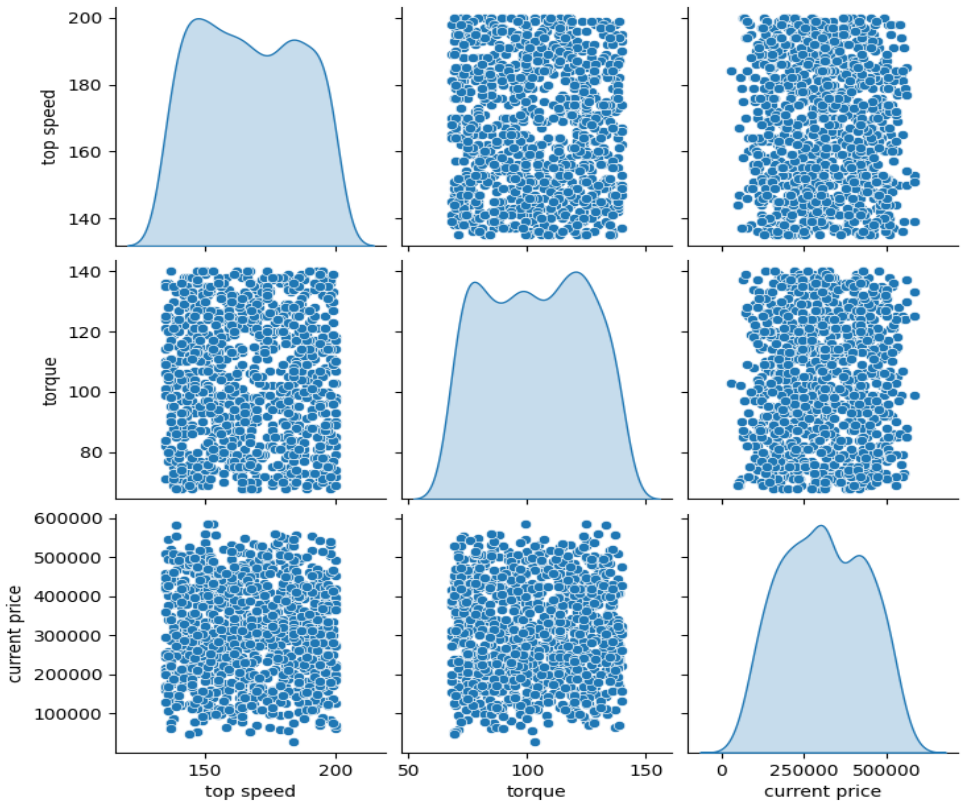

Figure 1 - Data analysis: Visualisation of irrelevant data
Source: authors

For Figure 2, it is precisely those parameters that influence the price of the vehicle to a significant extent that are shown. These parameters are years, km, rating, condition and economy. This can be seen in the scatterplots, where there is a clear relationship between the price of the vehicle and the given parameters. Unlike Figure 1, these correlations are not chaotic and are correlated with each other. However, it should also be noted that even this method of data selection is not perfect and it is possible to apply the neural network to the parameters in Figure 1 as well, as the network may be able to find patterns between parameters that are not visible to humans. Therefore, such an option is also included in the results evaluation section and confirms the correctness of the selection of the vehicle parameters.

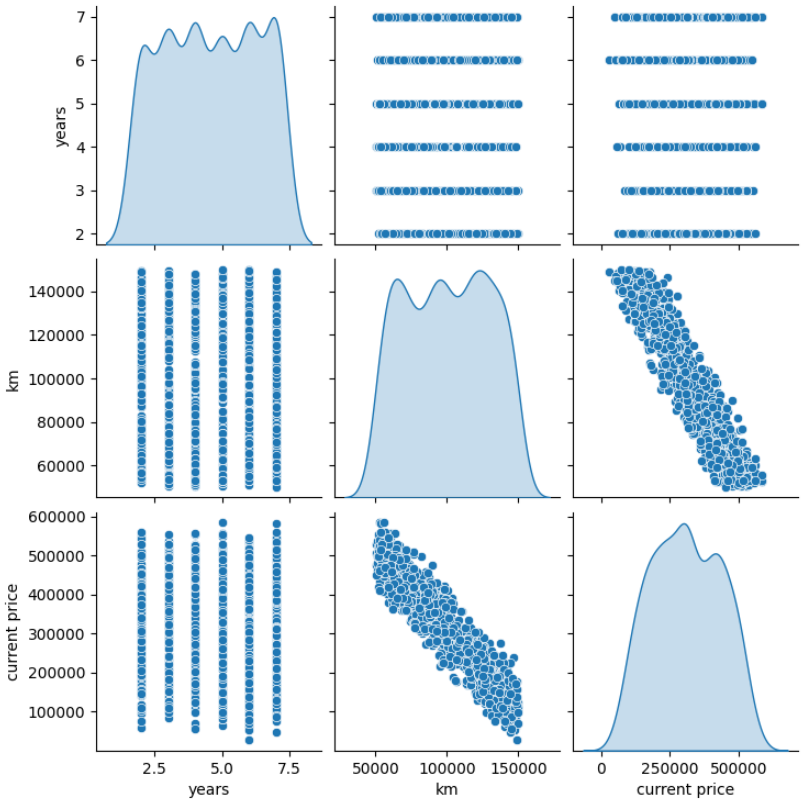

Figure 2 - Data analysis: visualisation of relevant data
Source: authors

Once these data are identified, they need to be processed as the values for each parameter take on different magnitudes. The normalization process will process this data, simplify it and allow the neural network to train faster with more accurate results. The normalization process consists to define minimum and maximum bounds for the values of the vehicle parameters. For this purpose, normalization in the form of formula (1) is used:

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

where x' represents the normalized value, x is for the original value, μ is the mean of all values of the parameter, σ is the standard deviation of the values of the parameter.
For μ, formula (2) then holds:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{2}$$

Where x_i is the parameter value for the i-th value in the dataset, n is the total number of values in the dataset.
The standard deviation is then expressed as formula (3):

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}} \tag{3}$$

Once all the values in all the vehicle parameters have been normalized, these values can be used as input to the neural network. However, the output value, i.e. the current price of the vehicle, is not normalized. The advantage is that after training the network, we get the instantaneous value of the vehicle without the need for an inverse function.

The next step is to split the data into three sets. The first set is the training set, which serves as the database of the data on which the network is trained. This data enters the network at each new epoch. The second set is the validation set. This set is used during training to verify whether the network is training in the correct way. At the end of each epoch, when all the training data is entered into the network, the verification data is entered into the network at the end. This data serves as data that the network did not see during training, so it compares its results with the new values. Based on the difference result from the network and the expected result from the validation set, a value is set that determines the need for the network to reconfigure its state so that the result from the next epoch has a lower difference output from the network with the expected value. The last set is the test set. This is the set that the network does not see during the entire training. After the network is trained, this data is fed into the network and the network result is compared with the expected result to determine the overall accuracy of the neural network.

For these purposes, the database of 1000 values was divided into three sections as follows:

- The training set contains 80% of all data,
- The verification set contains 17.5% of all data,
- Test set contains 2.5% of all data.

These data are partitioned to avoid repetition between sets.

If the training data contained 80%, or 800 values, each value would have to enter the network, it would have to be mathematically processed, and the gradient of change needed to adjust the network based on a comparison of the network output to the expected result would have to be calculated. Depending on the type of network, this gradient is calculated as the partial derivative of a specific network element, expressed as a formula (4):

$$[\nabla F(x)]_j = \frac{\partial F(x)}{\partial x_j} \tag{4}$$

Where *F(x)* expresses the function *F* for the parameter *x* in neuron *j*, where the parameter *x* is a vector expressed as:

$$x^T = [x_1 \quad x_2 \quad \cdots \quad x_n]$$

Where x^T is a vector of network elements consisting of the number of elements corresponding to the number of neurons and the number of layers in the network. For the total x^T it is possible to write:

$$x^T = \left[w_{1,1}^1 \quad w_{1,2}^1 \quad \cdots \quad w_{S^1,R}^1 \quad b_1^1 \quad \cdots \quad b_{S^1}^1 \quad w_{1,1}^2 \quad \ldots \quad b_{S^M}^M\right]$$

Where *w* denotes the weight and *b* the bias. These parameters are training parameters, i.e., variables that the network adjusts so that the result of this network is as close as possible to the expected result.

The difference of x^T as the result from the network is compared with the target value *v*. This difference is amplified for simpler calculation of the derivative. The result for each neuron in each layer is a matrix, called the Jacobian matrix, in which the gradients are recalculated according to the formula (5):

$$J(x) = \begin{bmatrix} \frac{\partial v_1(x)}{\partial x_1} & \frac{\partial v_1(x)}{\partial x_2} & \cdots & \frac{\partial v_1(x)}{\partial x_n} \\ \frac{\partial v_2(x)}{\partial x_1} & \frac{\partial v_2(x)}{\partial x_2} & \cdots & \frac{\partial v_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_N(x)}{\partial x_1} & \frac{\partial v_N(x)}{\partial x_2} & \cdots & \frac{\partial v_N(x)}{\partial x_n} \end{bmatrix} \tag{5}$$

For this reason, it is not only necessary that the network contains as few neurons and layers as possible, but also that the amount of resulting *v* is as low as possible. Therefore, the input data is processed in the form of batches. This batch contains several input parameters at a time, and therefore the gradient recalculation does not occur after each input value, but only after the batch value is reached. In this case, the number of 32 has been chosen, so only after 32 values have been input does the gradient recalculation occur. This method speeds up the training process. However, it may cause the accuracy of the network not to reach its maximum.

Creation of a neural network

Creating a neural network consists of several steps. Due to the relatively small amount of data, and the fact that such a network has to be trained in a few minutes, several initial parameters need to be set. With a database containing little data, overfitting is a common problem. Due to the small amount of data, the training process often occurs with high test loss or low accuracy. Typical steps are to reduce the training ratio, increase the training steps or increase the number of neurons or layers in the network. This can result in so-called overfitting, where the network is trained too well on the training set and if data other than the training data was input, the network does not show sufficient accuracy. Hence, it is necessary to introduce techniques into the network that will serve as a prevention against overfitting. One of these techniques is called Dropout. This is the probability with which the connection between neurons is "disconnected" after each epoch. These neurons then lose the ability to learn accurately based on the training data.

In order to create a network that can look for patterns among the data, it is necessary to use a functor that processes the output values from the neurons. A popular function is the

ReLU function, which filters the data in such a way that if the output from a neuron is negative, it converts that value to 0. However, any positive values from the neuron remain the same.

For the relationship used in the neurons of the network, a linear transformation is applied according to the formula (6):

$$Z = X.W + b \tag{6}$$

Where Z is the output of the neuron and W is the weight matrix.
To calculate the error or accuracy of the network, the formula (7) is used:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{true,i} - y_{pred,i}| \tag{7}$$

Where MAE is the mean absolute error, $y_{true,i}$ is the expected value and $y_{pred,i}$ is the value of the network output for the parameter. This value is then the key to changing the network parameter weights according to the formula (8):

$$W_{new} = W_{old} - lr \times gradient \tag{8}$$

Where W_{new} is the new value of the weight matrix, W_{old} is the old value of the weight matrix, lr is the learning rate, and the gradient is the calculated value based on MAE as the input value to the formula (4) to calculate the gradient. A value of 0.1 was chosen for the learning rate.
The entire neural network was created using the TensorFlow and Keras libraries, and Figure 3 shows the overall form.

Layer (type)	Output Shape	Param #
normalization_26 (Normalization)	(None, 5)	11
dense_40 (Dense)	(None, 128)	768
dropout_18 (Dropout)	(None, 128)	0
dense_41 (Dense)	(None, 128)	16512
dropout_19 (Dropout)	(None, 128)	0
dense_42 (Dense)	(None, 128)	16512
dropout_20 (Dropout)	(None, 128)	0
dense_43 (Dense)	(None, 1)	129

Total params: 33932 (132.55 KB)
Trainable params: 33921 (132.50 KB)
Non-trainable params: 11 (48.00 Byte)

Figure 3 - Architecture of neural network
Source: authors

The network has a total of 33921 trainable parameters, which is a really low number in the machine learning world. Thus, the network meets the requirement of small size which means fast training process.

The training process was carried out in epochs, the maximum number of which reached 150.

Change input data

To obtain data on the effectiveness of the analytical data selection based on the correlations between the values expressed in the form of graphs in the previous sections, the input data travelling to the neural network is modified and instead of the data that have been selected as relevant, the values of top speed, torque and hp are added among them. These data if non-chaotic should have a positive effect on the neural network. However, the analysis has shown the opposite as described, and the way to verify this claim is to simulate the input of all relevant and the aforementioned non-relevant data to the network. The results of both trainings are summarized in the Results section.

Results

The total training time of the neural network for training 150 epochs took only 23 seconds, and the input data was classified as relevant. The overall accuracy of the model reached 82%, which meets the requirement of accuracy higher than 80%. The result of comparing the predicted price and the actual price from the test set is shown in Figure 4.

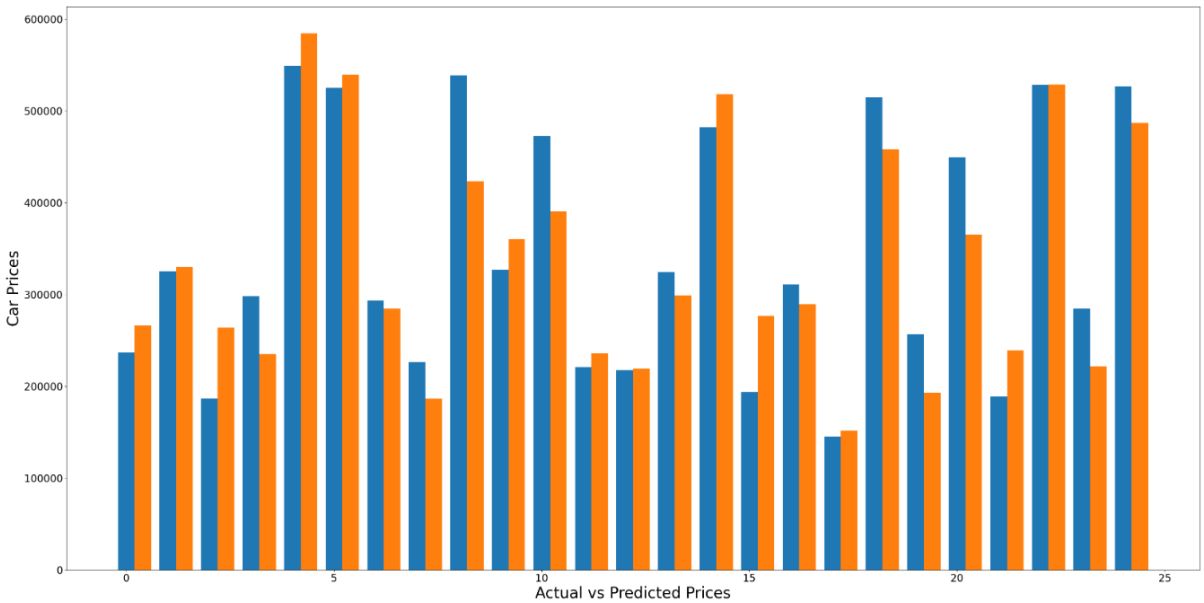

Figure 4 - Training results after input of relevant data
Source: authors

Figure 4 shows that the neural network can predict the price of vehicles in the bottom 50% of vehicle prices relatively accurately. For vehicle prices in the higher category, the network becomes relatively inaccurate for a few vehicles. However, the overall results are satisfactory.

The total training time of the neural network for training 150 epochs for both relevant and irrelevant data as input was 24 seconds, which is still a very low time. The overall accuracy of the network dropped to 57%. The result of comparing the predicted price and the actual price from the test set is shown in Figure 5.

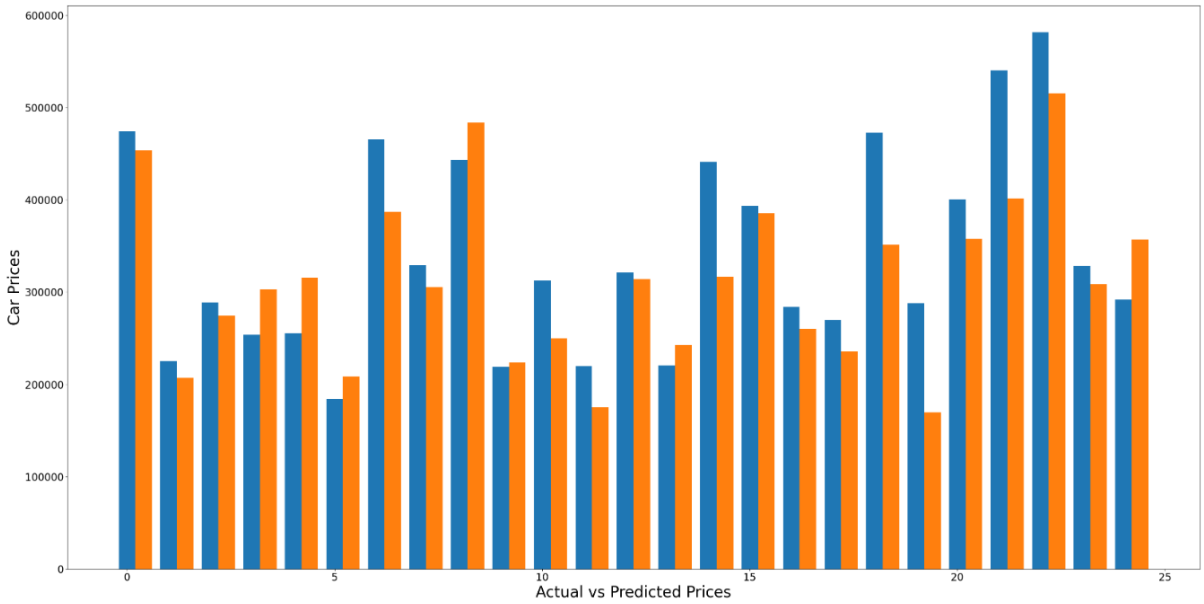

Figure 5 - Training results after input of irrelevant data
Source: authors

From the given figure, it is clear that data selection is indeed important, as the same model with the same parameters with only a change in the input data causes a significant decrease in the accuracy of the network.

The following table briefly includes a comparison of such models and highlights the differences and the reality of the need for analytical data selection and the accuracy of the data analysis performed in the previous sections.

Model	**Input data**	**Training time**	**Epochs**	**Accuracy**
Relevant	years, km, rating, condition, economy	23 seconds	150	82%
Relevant + Irrelevant	years, km, rating, condition, economy, road old, on road now, top speed, torque, hp	24 seconds	150	57%

Table 1 - Training summary
Source: authors

All calculations were performed on the powerful Tesla T4 70W TPU, therefore the training time is also significantly low.

Conclusion

In this paper, the process of developing a system for vehicle price prediction using neural network has been investigated. The goal was to create a functional system with prediction accuracy greater than 80%. Initially, the need for a database was analysed and relevant data that affect the price of vehicles was identified. After processing the data, a neural network was created using TensorFlow and Keras libraries. The network contained different layers and used the ReLU function to process the outputs. The number of trainable parameters was optimized to achieve a fast training process and minimize the risk of overfitting. After training the network, its performance was evaluated using a test dataset. The overall accuracy of the model reached the desired value of more than 80%, confirming the success of the system. An important note from the analysis is the need to select relevant data for training the neural network. Proper analytical data selection was shown to have a critical impact on the accuracy of the model. Comparison of the results between relevant and irrelevant data showed a significant difference in the accuracy of the network. Overall, the system can be considered successful as it achieved the required prediction accuracy.

This paper provides insights and guidelines for the development and optimization of a system for vehicle price prediction using neural networks. At the same time, it highlights the importance of proper analytical data selection and thorough analysis in the model building process.

Further Research

In future research, a regression model will be used to determine the parameters that have the most significant effect on the neural network results. This will be part of the process of optimizing the model to improve its accuracy and efficiency. The regression model will analyse the input data and identify the key factors influencing the price of vehicles, providing important information to further improve the predictive ability of the network. Extending the database will be one of the main objectives of future research. Increasing the amount of data available will allow the model to extract more information and better learn patterns in the data. This expansion should focus on collecting additional data, with an emphasis on key parameters affecting vehicle price, thereby improving the accuracy and performance of the model. Increasing the complexity of the neural network will also be part of future research. A more complex network will be able to extract and interpret information from the input data more efficiently, which should lead to better predictive ability of the model. However, the increase in complexity may also result in longer training time and risk of overfitting, which will require additional attention and adjustments in the training process.

This article was recommended for publication in the scientific journal Young Science by:
doc. Ing. Pavol Pecho, PhD.

Literature

1. Al-Turjman, F., Hussain, A.A., Alturjman, S., Altrjman, C., 2022. *Vehicle Price Classification and Prediction Using Machine Learning in the IoT Smart Manufacturing Era*. Sustainability 14, 9147. https://doi.org/10.3390/su14159147.
2. Chollet, F., 2021. *Deep learning with Python, Second edition*. ed. Manning, Shelter Island, NY.
3. Enis Gegic, Becir Isakovic, Dino Kečo, Zerina Mašetić, Jasmin Kevrić, 2019. *Car Price Prediction using Machine Learning Techniques.* TEM J. 8, 113–118.
4. Pillai, A.S., 2022. A Deep Learning Approach for Used Car Price Prediction. J. Sci. Technol. 3, 31–50.
5. Prashant Gajera, Akshay Gondaliya, Jenish Kavathiya, 2021. *OLD CAR PRICE PREDICTION WITH MACHINE LEARNING*. Int. Res. J. Mod. Eng. Technol. Sci.
6. Samruddhi, K., Ashok Kumar, R., 2020. *Used Car Price Prediction using K-Nearest Neighbor Based Model*. Int. J. Innov. Res. Appl. Sci. Eng. 4, 629–632. https://doi.org/10.29027/IJIRASE.v4.i2.2020.629-632.
7. Varshitha, J., Jahnavi, K., Lakshmi, C., 2022. *Prediction Of Used Car Prices Using Artificial Neural Networks And Machine Learning*, in: *2022 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–4. https://doi.org/10.1109/ICCCI54379.2022.9740817.