Predicting Vehicle Fuel Efficiency: A Comparative Analysis of Machine Learning Models on the Auto MPG Dataset

Alpay Doruk[1], Muhammed Ali Bayram[2]
[1]*Bandırma Onyedi Eylül University, Dept. of Computer Engineering, Bandırma, Balıkesir, Turkey, adoruk@bandirma.edu.tr, ORCID:*
[2]*Bandırma Onyedi Eylül University, Dept. of Computer Engineering, Bandırma, Balıkesir, Turkey, mabayram@gmail.com, ORCID: 0000-0001-7784-2992*

This study explores the application of various machine learning models to predict vehicle fuel consumption using the Auto MPG dataset. It examines the effectiveness of algorithms such as Decision Tree Regressors, Random Forests, Support Vector Regressors, and neural network-based models like LSTM and GRU. The study aims to enhance fuel efficiency prediction by analyzing factors like engine specifications, driving habits, and vehicle design. The models' performance is evaluated using metrics such as R-squared (R2), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to ensure accuracy and minimize error.

Keywords: *MPG, Regression, Machine Learning*

1. Introduction

The automotive sector is at the forefront of considerable change in an era when environmental sustainability meets with technical innovation[1,2]. With the intensifying effects of climate change and the ever-increasing global demand for energy efficiency, the capacity to precisely predicting vehicle fuel use has emerged not only as a critical challenge but also as a massive potential. Enter the world of machine learning, a dynamic and powerful tool that is changing the way we think about automobile fuel efficiency [3,4].

This study evaluates the performance of different machine learning models to predict car fuel usage with high accuracy. By examining the complexities of algorithms such as Decision Tree Regressors, Random Forests, Support Vector Regressors and neural network-based models such as LSTM and GRU, their success in modeling and analyzing complex factors affecting fuel consumption is discussed.

Machine learning provides a diverse prism through which fuel consumption may be precisely projected, from the data-driven complexities of engine specifications and driving habits to the broader implications of vehicle design and climatic conditions. This not only encourages the creation of more fuel-efficient automobiles, but also prepares the way for consumers and manufacturers to make wiser, data-informed decisions. Machine learning models are used in many areas such as communication systems[5-7], sound processing [8-10], natural language processing [11] and biomedical studies [12-13].

Auto MPG dataset was used in this study. The Auto MPG dataset is a classic collection of data originally used in the 1983 American Statistical Association Exposition. It contains information about 398 automobiles, including attributes such as miles per gallon (MPG), number of cylinders, displacement, horsepower, weight, acceleration, model year, origin, and the car name [14]. This dataset has become a mainstay for regression analysis, serving as a benchmark for algorithms aiming to predict fuel efficiency based on various vehicle characteristics.

The importance of accurately predicting a vehicle's MPG cannot be overstated, especially in a world increasingly concerned with energy efficiency and environmental sustainability. Fuel economy is a critical factor for both consumers and manufacturers. For consumers, it affects the long-term cost of vehicle ownership and day-to-day expenses, as fuel is a significant operational cost. For manufacturers, MPG figures are essential for meeting regulatory

standards and for competitive marketing. Accurate MPG predictions can lead to better-informed decisions regarding vehicle design and can highlight potential areas for improvement in fuel efficiency.

In addition to economic and regulatory motivations, the accurate prediction of MPG plays a vital role in environmental stewardship. Better fuel efficiency translates to lower greenhouse gas emissions, which is crucial for combating climate change. As such, the Auto MPG dataset not only provides a foundation for developing predictive models but also serves as a tool for progressing towards more sustainable automotive technology.

In fact, various studies have been carried out in this field. Srivatsa Bindingnolle Narasimha [15], the process is carried out to make adjustments in the company's immediate demands. To anticipate fuel usage, they primarily used K - means clustering models. As a result, the average overall inaccuracy was 10%. They also created the network architecture in order to improve MPG estimates. This model was created to forecast the average change in MPG during rerouting. The fundamental method in this article by Hyunkun kim et al [16] was to maximize fuel efficiency via reinforcement learning employing neural networks, where their models are generally trained to enhance and forecast fuel economy. Even on hidden routes. This model boosted fuel savings by 8%. Sandareka et al. [17] estimated a bus's fuel usage utilizing numerous characteristics such as road, vehicle, driver, and weather conditions, as well as machine learning approaches such as random forest and neural networks. They discovered that random forest produced the greatest outcomes of the two. Shalini et al. [18] examines the various factors affecting fuel consumption in vehicles, such as usage patterns, car models, and fuel prices, which have significant implications for both environmental impact and economic considerations. It explores the application of machine learning techniques, like linear regression, and optimization methods, such as gradient descent, to develop predictive models that aim to enhance fuel efficiency with high accuracy and minimal error.

In this study, firstly, an analytical evaluation of the features in the Auto MPG dataset is presented. Then, fuel consumption is estimated using 5 different machine learning models. To evaluate model performances, the 10-fold cross-validation method was used and comparisons were made with 4 different metrics.

2. Material and Methods

2.1 Long Short-Term Memory (LSTM) Networks

LSTM networks are a type of recurrent neural network (RNN) capable of learning long-term dependencies, introduced by Sepp Hochreiter and Jürgen Schmidhuber in 1997. They were created to remedy the vanishing gradient problem that can be encountered when training traditional RNNs. Vanishing gradients make it difficult for the RNN to learn and retain information over many time steps, hence the introduction of LSTMs that use a gating mechanism to control the memorization process.
An LSTM unit includes three types of gates: input gate, forget gate, and output gate, which regulate the flow of information. Each gate can be thought of as a 'conventional' artificial neuron in a neural network that provides a way to optionally let information through.
The key equations governing the behavior of an LSTM unit for a single time step are as follows:
Forget gate decides what information should be thrown away from the cell state.

$$ft = \sigma(Wf \cdot [h_{t-1}, x_t] + bf)$$

Here, σ denotes the sigmoid function, W_f represents the weight matrix for the forget gate, b_f is the bias term, h_{t-1} is the previous hidden state, and x_t is the current input.
Input Gate updates the cell state with new information.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

i_t is the input gate's activation vector, $\tilde{C}$ is the candidate values vector, which is combined with the old state to create the new state.
Cell State Update is the combination of forgetting the old state and adding the new state.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

C_t is the new cell state, $C_{t-1}$is the old cell state, and $*$ denotes element-wise multiplication.
Output gate decides what the next hidden state should be.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * tanh(C_t)$$

o_t is the output gate's activation vector, and h_t is the output hidden state.
The functions σ and $tanh$ provide non-linearities necessary for the network to learn complex data patterns. The weight matrices and bias vectors are parameters that are learned during training.
The LSTM's ability to maintain a cell state over time, along with its gated architecture, makes it highly effective for tasks involving sequential data, such as time series prediction, natural language processing, and speech recognition.

2.2 Gated Recurrent Units (GRU)

GRUs are a type of RNN architecture that, similar to LSTMs, are designed to model temporal sequences and their long-range dependencies more effectively than standard RNNs. GRUs achieve this by using gating units to regulate the flow of information. These gates are analogous to the memory cells in LSTMs and help to capture dependencies for different time scales of sequential data.

The architecture of a GRU is characterized by two gates:

Update gate decides to what extent the unit updates its activation, or content. It helps the model to determine how much of the past information needs to be passed along to the future. The update gate is computed using the following formula:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$

Here, z_t represents the update gate's activation vector at time *t*, σ denotes the sigmoid function, W_z is the weight matrix for the update gate, $b_z$is the bias term, h_{t-1} is the previous hidden state, x_t and is the current input vector.

Reset gate determines how much of the past information to forget. It is used to decide how much of the past information to let through and is crucial for capturing time dependencies of varying lengths. The reset gate is computed as:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$

$r_t$is the reset gate's activation vector, and W_r, b_r are the weight matrix and bias for the reset gate, respectively.

The actual hidden state h_t at time *t* is then updated using the following formula:

$$h_t = z_t * h_{t-1} + (1 - z_t) * \tilde{h}_t$$

Where $\tilde{h}_t$ is the candidate hidden state, calculated using:

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t] + b)$$

Here, *W* and *b* are the weight matrix and bias term for the candidate hidden state.

The operations within a GRU allow it to keep relevant information in the hidden state over long sequences, discard irrelevant information, and make updates to the hidden state. This mechanism helps to mitigate the vanishing gradient problem that plagues standard RNNs.

2.3 Decision Tree Regressor

A Decision Tree Regressor is a model used in the context of supervised learning where the goal is to predict continuous outcomes. It is a non-parametric method, implying it makes no assumptions about the form of the mapping function from input variables to the target variable. The model operates by partitioning the input space into distinct regions,

where the prediction for a given observation is the mean of the dependent variable for training observations in the region to which the new observation belongs.

Formally, the construction of a decision tree involves recursive binary splitting of the data. At each node of the tree, the algorithm selects the predictor and a corresponding threshold that produces the most homogeneous sub-nodes, according to a predetermined metric. For regression problems, this metric is often the mean squared error (MSE), and the algorithm seeks to minimize the following objective function at each split:

$$min_{j,t}\left[min_{c_1}\sum i:xij < t(y_i - c_1)^2 + min_{c_2}\sum i:xij \geq t(y_i - c_2)^2\right]$$

Here, x_{ij} represents the value of the *j*-th predictor for the *i*-th observation, *t* is the threshold, c_1 and $c_2$are the mean responses for the observations in the resulting two regions defined by *t*, and the summations are over those observations.

The process of recursive splitting continues until a stopping criterion is reached, which could be a maximum depth of the tree, a minimum number of observations required in a node to permit a split, or a minimal improvement to the fit of the model required to justify a split. To prevent overfitting, which decision trees are particularly prone to, the tree may be pruned back with techniques like cost-complexity pruning.

Once the tree is constructed, it can be used to make predictions on new data. An observation is filtered through the tree, following the splits at each node according to its attribute values until it reaches a leaf node. The predicted outcome for the observation is then the mean of the target variable for the training observations in the leaf node.

The interpretability of decision trees is a notable advantage. The model's decisions are transparent and can be visualized, allowing for easy understanding of how the input features are used to make predictions. Despite their simplicity, decision trees can capture complex non-linear relationships between features and the target variable, making them powerful tools for regression tasks.

2.4 Support Vector Regressor

Support Vector Regressor (SVR), a variant of Support Vector Machines (SVM), is a powerful algorithm used in machine learning for regression tasks. It is based on the principle of Structural Risk Minimization (SRM) from statistical learning theory, which seeks to minimize an upper bound of the generalization error, rather than minimizing the training error. This approach is fundamentally different from techniques that aim to minimize empirical risk, such as Ordinary Least Squares (OLS) regression.

The core idea behind SVR is to find a function $f(x)$ that deviates from the actual observed targets y_i by a value no greater than ε for each training point x_I and at the same time is as flat as possible. Flatness in this context typically means seeking small values for the weights in the linear combination of the inputs that make up $f(x)$. Mathematically, SVR solves the following optimization problem:

Minimize:

$$\frac{1}{2} \parallel w \parallel^2 + C\sum_{i=1}^{n}(\xi i + \xi i^*)$$

Subject to:

$$yi - \langle w, xi\rangle - b \leq \epsilon + \xi i,$$

$$\langle w, xi\rangle + b - yi \leq \epsilon + \xi i^*,$$

$$\xi i, \xi i^* \geq 0 \text{ for all i} = 1, \ldots, \text{n}.$$

Here, $\langle w,x_i \rangle$ represents the dot product between the weight vector w and a feature vector x_i, b is the bias term, ξi and ξi^* are slack variables introduced to soften the margin, C is the regularization parameter, and ϵ is the epsilon-insensitive loss function which defines the margin width that is tolerated in the training data.

The dual form of the SVR optimization problem introduces Lagrange multipliers, which transform the problem into a quadratic programming problem. The solution involves only a subset of the training data, known as support vectors, which lie on the boundary of the ε tube or violate the ε constraint.

In cases where the relationship between the independent variables and the dependent variable is non-linear, SVR can be extended to model non-linear relationships by applying a kernel function. The kernel function implicitly maps the input space into a high-dimensional feature space, enabling the application of linear regression techniques in this transformed space. Common choices of kernel functions include polynomial, radial basis function (RBF), and sigmoid kernels.

SVR is particularly noted for its robustness, especially in situations where the data has a lot of noise or when the number of features is greater than the number of observations. However, its performance is heavily dependent on the appropriate setting of hyperparameters like C, ϵ, and the parameters of the chosen kernel

2.5 Random Forest Regressor

A Random Forest Regressor is an advanced machine learning model predominantly used for regression tasks, which enhances prediction accuracy and overcomes overfitting issues inherent in decision trees through the ensemble learning approach. This method involves constructing a multitude of decision trees during the training phase and producing the output as the mean prediction of these individual trees. The methodology of Random Forest Regressor begins with bootstrap sampling, where each tree in the forest is trained on a distinct bootstrap sample drawn from the original dataset. This step ensures that each tree encounters slightly varied data, fostering diversity within the forest and subsequently reducing variance in the overall model.

An additional layer of randomness is introduced in feature selection for each split in the tree. Rather than examining all available features, a random subset of features is considered, which further contributes to de-correlating the trees and enhances the model's robustness against noise. Each tree is grown to its maximum depth, which allows the trees to maintain low bias, while the averaging of predictions across trees reduces the variance. The result is a model that effectively balances bias and variance, capable of handling high-dimensional data and large datasets.

The Random Forest Regressor is adept at modeling complex, non-linear relationships, and it provides insights into feature importance based on their contribution to reducing impurity in the trees. However, this robustness and versatility come with the cost of increased computational resources and reduced interpretability compared to single decision trees. The performance of the model is highly dependent on the tuning of hyperparameters such as the number of trees and the number of features considered for splits. Despite these considerations, Random Forest Regressors are widely favored for their general applicability to various regression problems, offering a blend of simplicity and effectiveness.

3. Performance Evaluation Metrics

In statistical modeling and machine learning, evaluating the performance of regression models is crucial, and this is where metrics like R2, RMSE, MAE, and MAPE come into play. R2, or the coefficient of determination, serves as a key indicator of the proportion of variance in the dependent variable that can be predicted from the independent variables, offering a scale from 0 to 1 where higher values denote a better fit. The Root Mean Square Error (RMSE) takes a different approach, providing a quadratic scoring rule that captures the average magnitude of the error; it's particularly sensitive to larger errors due to its squaring of differences. Mean Absolute Error (MAE), in contrast, offers a linear measure by averaging the absolute differences between predictions and actual observations, making it more robust against outliers. Lastly, the Mean Absolute Percentage Error (MAPE) presents accuracy as a percentage, thus offering an intuitive and

relative measure of error across different scales or datasets. Each metric sheds light on different aspects of model accuracy and error, making them collectively valuable for a comprehensive evaluation of a regression model's performance.

4. Results and Discussion

In this section, first, an analytical evaluation of the features in the Auto MPG dataset is carried out using the correlation matrix, scatter plot and tSNE graphics. Afterwards, the MPG value is predicted using 5 different machine learning models.

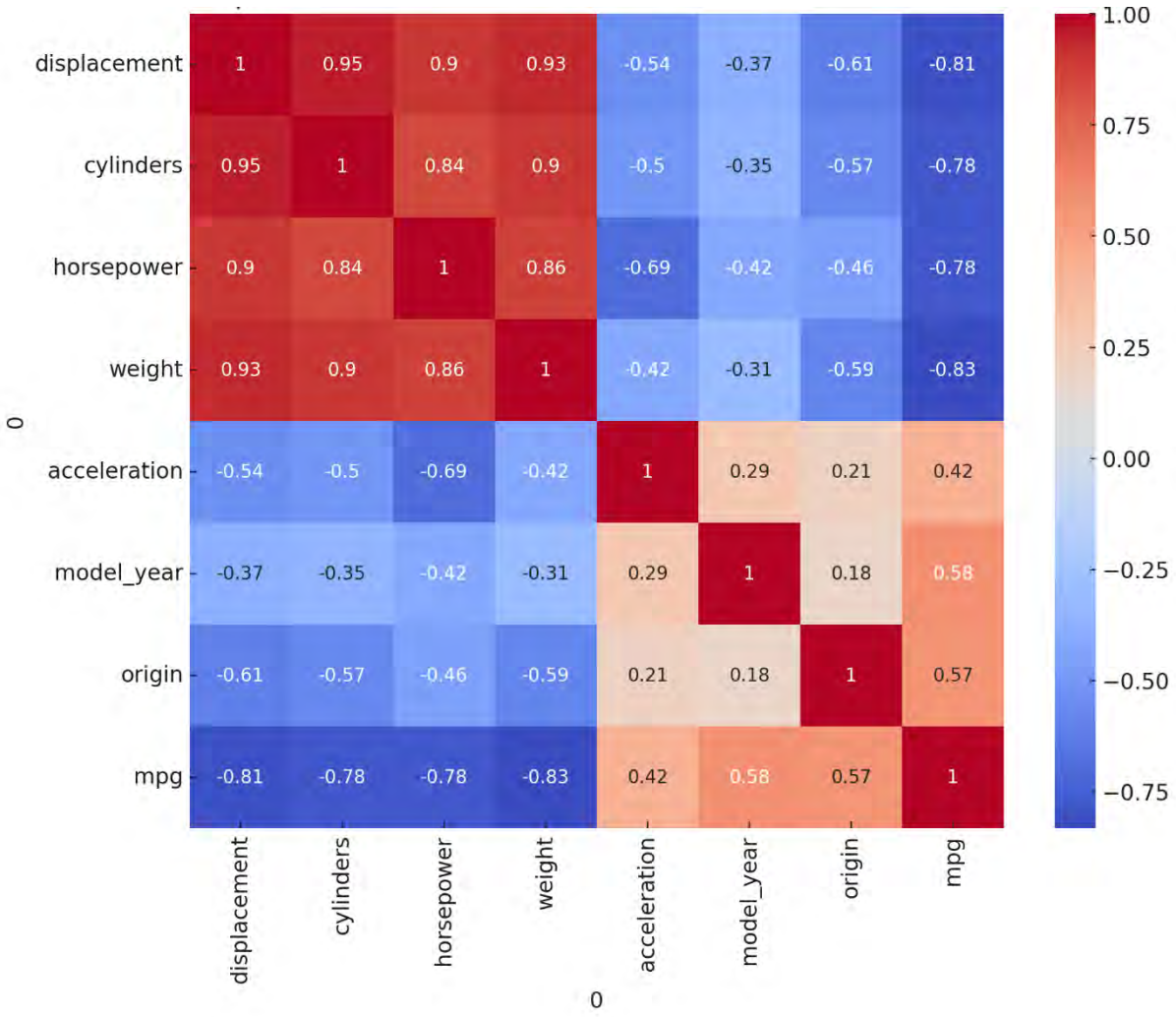

Figure 1. *Correlation Matrix of Auto MPG dataset.*

Figure 1 shows the correlation matrix for the Auto MPG data set. The matrix indicates how each pair of attributes is correlated. A positive correlation (closer to +1.00) suggests that as one attribute increases, the other attribute also increases. A negative correlation (closer to -1.00) indicates that as one attribute increases, the other decreases. Displacement, Cylinders, Horsepower, and Weight vs. MPG features show strong negative correlations with mpg, indicating that vehicles with higher displacement, more cylinders, more horsepower, and greater weight tend to have lower miles per gallon, which means they are less fuel-efficient.

Model Year vs. MPG a moderate positive correlation between model_year and mpg, suggesting that newer models tend to be more fuel-efficient. Origin vs. MPG's also a moderate positive correlation between origin and mpg. The origin attribute likely codes the country of origin, which might suggest that cars from certain regions are more fuel-efficient than others.

The high positive correlations between displacement, cylinders, horsepower, and weight indicate that these features are closely related; typically, larger, more powerful cars are also heavier and have more cylinders. These insights can be particularly useful for automobile manufacturers looking to improve fuel efficiency and for consumers interested in the environmental impact and fuel economy of their vehicles.

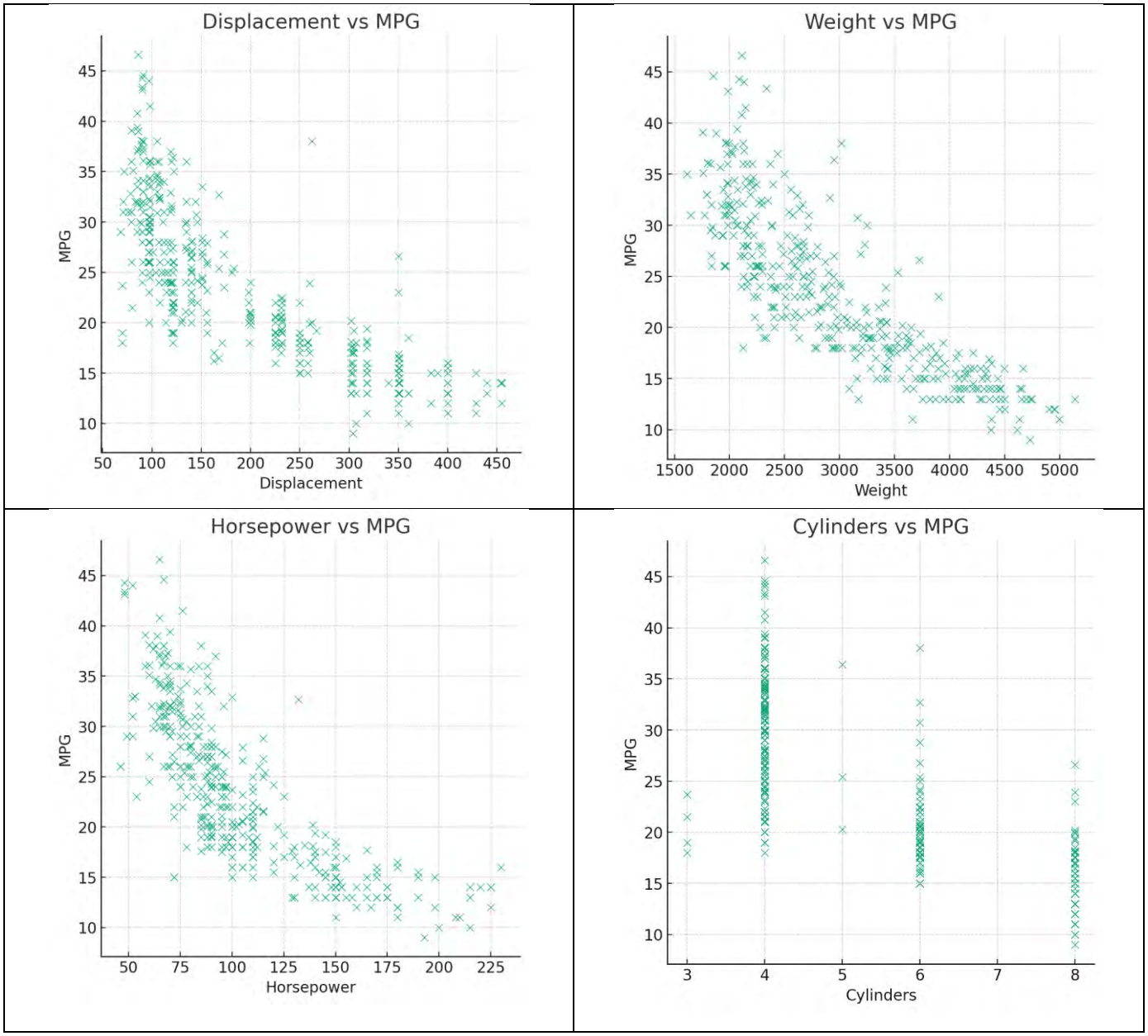

Figure 2. *Scatter plot representation of 4 features in the Auto MPG dataset*

Figure 2 shows scatter plot representations. The series of scatterplots created for the Auto MPG dataset each reveal distinct relationships between 'mpg' and various vehicle attributes. The first plot, showcasing the inverse relationship between displacement and 'mpg', indicates that vehicles equipped with larger engines typically exhibit lower fuel efficiency. This suggests that displacement is a significant factor influencing a vehicle's fuel consumption. Moving to the second plot, the negative correlation between the vehicle's weight and 'mpg' is evident. Heavier vehicles consistently demonstrate lower 'mpg', highlighting the impact of weight on fuel efficiency and emphasizing the potential benefits of lightweight materials in automotive design.

The third scatterplot, which compares horsepower to 'mpg', also presents a clear inverse relationship. This plot implies that vehicles with more powerful engines tend to have lower 'mpg', a trend commonly seen in performance vehicles where power takes precedence over fuel economy. Lastly, the final scatterplot illustrates the relationship between the number of cylinders and 'mpg'. Here, we see that vehicles with a higher number of cylinders generally report lower 'mpg', reinforcing the concept that larger engines, often with more cylinders, lead to increased fuel consumption. Collectively, these scatterplots provide valuable insights into the characteristics that affect a vehicle's fuel efficiency, which can inform both manufacturers and consumers in their decisions related to vehicle design and selection.

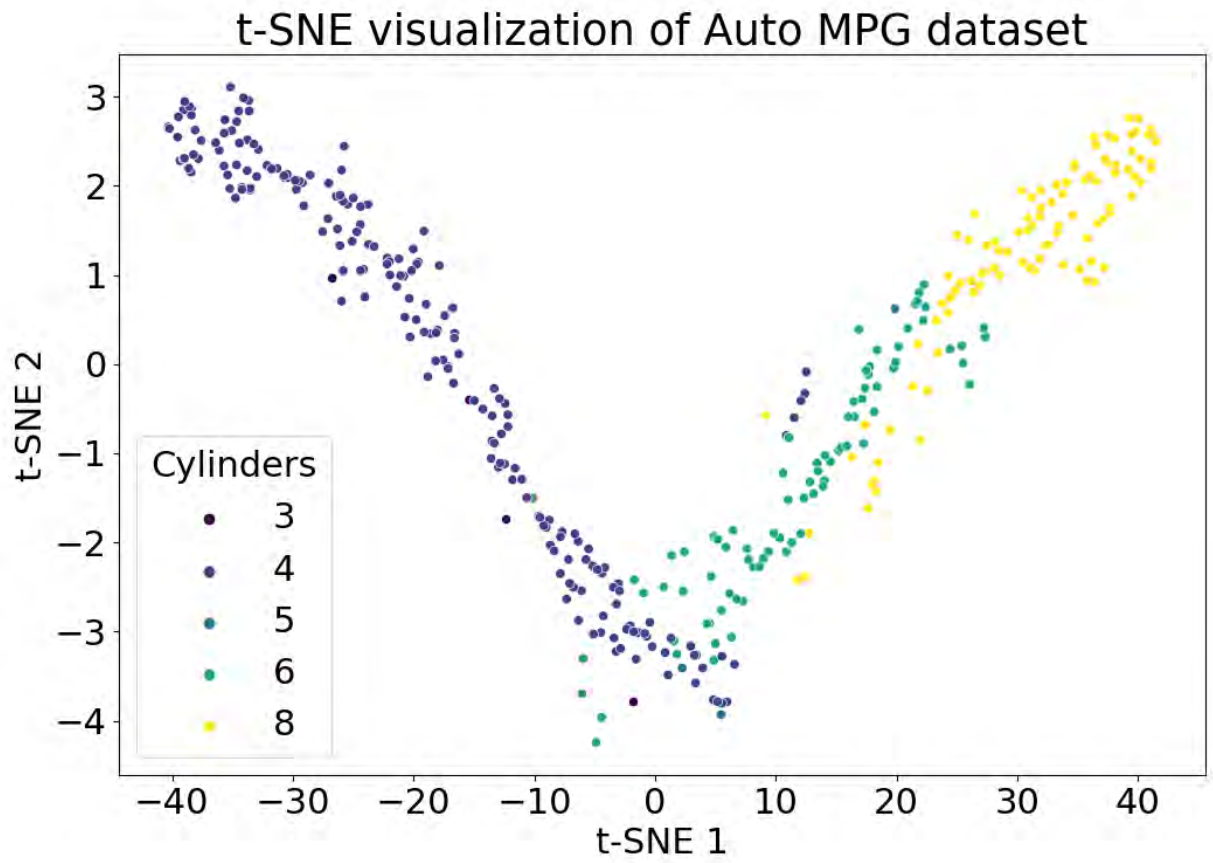

Figure 3. *t-SNE graph of the Auto MPG dataset*

Figure 3 also shows the t-SNE chart. The t-SNE chart provided depicts the Auto MPG dataset reduced to two dimensions, which allows us to visualize the high-dimensional data's inherent structure. The chart shows distinct clusters that likely represent cars with similar characteristics based on the features such as cylinders, displacement, horsepower, weight, and acceleration. Colours differentiate the number of cylinders in the cars, illustrating that the number of cylinders is a significant feature that influences the clustering, as cars with the same number of cylinders tend to be grouped together. There is a smooth transition between clusters, especially noticeable from cars with 4 cylinders to those with 6 and 8, suggesting gradual differences between these groups rather than abrupt changes. A few points stand apart from the main clusters, which could be indicative of cars with unique characteristics or potential anomalies in the data. The varying density of points within clusters can indicate the level of variance within each group, with the 4-cylinder cluster appearing quite dense, implying a lower variance within that group compared to others. This visualization is useful for understanding the grouping of cars based on their features, and it highlights the number of cylinders as a strong distinguishing factor within this dataset.

Table 1. *Regression Results for Five Machine Learning Models.*

	LSTM	GRU	Decission Tree Regressor	Support Vector Regressor	Random Forest Regressor
R2	0.659	0.69	0.538	0.182	0.771
RMSE	4.533	4.324	5.28	7.025	3.714
MAE	3.197	3.178	3.691	5.468	2.688
MAPE	37.225	39.781	40.91	34.87	38.88

Finally, the MPG value is estimated with machine learning models. Table 1 shows the results obtained by 10 fold cross-validation of 5 different machine learning models. The table presents a comparative analysis of several predictive models evaluated on the Auto MPG dataset, which comprises various automobile attributes with the goal of predicting fuel efficiency measured in miles per gallon (MPG). The performance of each model is quantified using four metrics: R-squared (R2), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

In this context, R2 values reflect each model's ability to explain the variability of MPG among different automobiles. The Random Forest Regressor exhibits the highest R2 value (0.771), signifying it has the strongest explanatory power for the variance in MPG, while the Support Vector Regressor has the lowest at 0.182, indicating a weak explanatory capability.

RMSE values provide insights into the typical deviation of the predicted MPG from the actual MPG values. The Random Forest Regressor shows the lowest RMSE (3.714), indicating that its predictions are closest to the actual MPG values. On the other hand, the Support Vector Regressor, with the highest RMSE (7.025), demonstrates the largest average deviation in MPG predictions.

MAE values represent the average magnitude of the errors in the MPG predictions, without considering their direction. Again, the Random Forest Regressor has the lowest MAE (2.688), suggesting that it has the smallest average error in predicting MPG. Conversely, the Support Vector Regressor records the highest MAE (5.468), highlighting its lesser accuracy in MPG prediction.

MAPE provides a relative measure of the prediction error in terms of MPG, offering a perspective on the error proportionate to the true MPG values. The Support Vector Regressor achieves the lowest MAPE (34.87%), suggesting that while its predictions are on average further from the true values, when errors occur, they constitute a smaller percentage of the actual MPG. The Decision Tree Regressor has the highest MAPE (40.91%), indicating that its prediction errors are the largest relative to the actual MPG values. Figure 4 shows the graph of the results obtained from one fold of the Random Forest Regressor model.

In the context of the Auto MPG dataset, the Random Forest Regressor would likely be considered the most reliable model for predicting MPG, as indicated by its superior performance across most metrics. Accurate MPG prediction is essential for assessing vehicle fuel efficiency, a critical factor for cost-conscious consumers, environmentally aware stakeholders, and manufacturers striving to meet regulatory standards. The lower RMSE and MAE values are particularly important in this case, as they suggest that the Random Forest model can predict MPG with a high degree of precision and accuracy, which is valuable for both practical and environmental implications in the automotive industry.

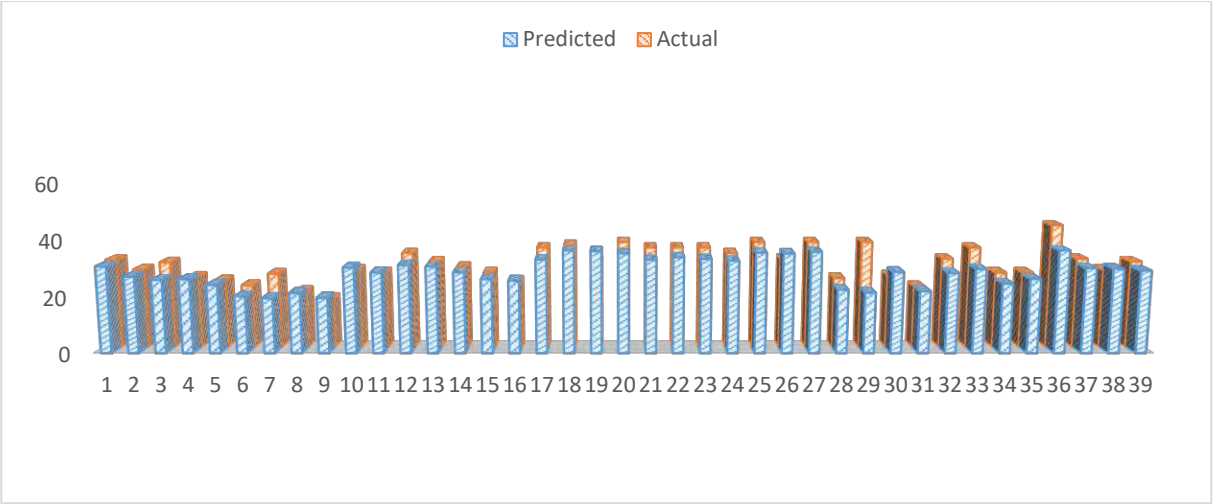

***Figure 4**. Graph of the one fold results for Random Forest Regressor model*

5. Conclusion

The study concludes that the Random Forest Regressor is the most effective model for predicting MPG in the Auto MPG dataset, demonstrating superior performance across most evaluation metrics. This model's success in accurately predicting fuel efficiency is crucial for developing cost-effective, environmentally friendly automotive technology. The findings highlight the potential of machine learning in advancing sustainable practices in the automotive industry, providing insights for manufacturers and consumers alike in improving fuel efficiency and reducing environmental impact.

REFERENCES

[1] Wells, Peter, and Paul Nieuwenhuis. "Transition failure: Understanding continuity in the automotive industry." Technological Forecasting and Social Change 79.9 (2012): 1681-1692.

[2] Kushwaha, Gyaneshwar Singh, and Nagendra Kumar Sharma. "Green initiatives: a step towards sustainable development and firm's performance in the automobile industry." Journal of cleaner production 121 (2016): 116-129.

[3] Moradi, Ehsan, and Luis Miranda-Moreno. "Vehicular fuel consumption estimation using real-world measures through cascaded machine learning modeling." Transportation Research Part D: Transport and Environment 88 (2020): 102576.

[4] Abediasl, Hamidreza, et al. "Real-time vehicular fuel consumption estimation using machine learning and on-board diagnostics data." Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering (2023): 09544070231185609.

[5] Seyman M. N., Taşpınar N., (2013), "Channel Estimation Based on Neural Network in Space Time Block Coded MIMO-OFDM System, Digital Signal Processing, Vol.23, No.1, pp. 275-280.

[6] Seyman M. N., Taşpınar N., (2013), "Radial Basis Function Neural Networks for Channel Estimation in MIMO-OFDM Systems", Arabian Journal for Science and Engineering, Vol.38, No. 8, pp. 2173-2178.

[7] Seyman M. N., (2023), "Convolutional Fuzzy Neural Network Based Symbol Detection in MIMO NOMA Systems", Journal of Electrical Engineering, Vol. 74, No. 1, pp. 60-64.

[8] Ozer, Ilyas, Zeynep Ozer, and Oguz Findik. "Noise robust sound event classification with convolutional neural network." Neurocomputing 272 (2018): 505-512.

[9] Ozer, Ilyas, Zeynep Ozer, and Oguz Findik. "Lanczos kernel based spectrogram image features for sound classification." Procedia computer science 111 (2017): 137-144.

[10] FADEL, Mariem Mine CHEİKH MOHAMED, and Ö. Z. E. R. Zeynep. "Trafikle İlgili Seslerin İşitsel Modeller ve Konvolüsyonel Sinir Ağları Kullanılarak Sınıflandırılması." *Mühendislik Bilimleri ve Araştırmaları Dergisi* 5.2 (2023): 233-242.

[11] Ozer, Zeynep, Ilyas Ozer, and Oguz Findik. "Diacritic restoration of Turkish tweets with word2vec." Engineering Science and Technology, an International Journal 21.6 (2018): 1120-1127.

[12] Bardak F. K., Seyman M. N., Temurtaş F., (2022), " EEG Based Emotion Prediction with Neural Network Models", Tehnički Glasnik, Vol. 16, No. 4, pp. 497-502.

[13] Gorur, K., Olmez, E., Ozer, Z., & Cetin, O. (2023). EEG-Driven Biometric Authentication for Investigation of Fourier Synchrosqueezed Transform-ICA Robust Framework. Arabian Journal for Science and Engineering, 1-23.

[14] Quinlan,R.. (1993). Auto MPG. UCI Machine Learning Repository. https://doi.org/10.24432/C5859H.

[15] Shalini, Lingampally, Soumyalatha Naveen, and U. M. Ashwinkumar. "Prediction of Automobile MPG using Optimization Techniques." 2021 IEEE Madras Section Conference (MASCON). IEEE, 2021.

[16] Kim, Hyunkun, et al. "Autonomous vehicle fuel economy optimization with deep reinforcement learning." Electronics 9.11 (2020): 1911.

[17] Wickramanayake, Sandareka, and HMN Dilum Bandara. "Fuel consumption prediction of fleet vehicles using machine learning: A comparative study." 2016 moratuwa engineering research conference (mercon). IEEE, 2016.

[18] Shalini, Lingampally, Soumyalatha Naveen, and U. M. Ashwinkumar. "Prediction of Automobile MPG using Optimization Techniques." 2021 IEEE Madras Section Conference (MASCON). IEEE, 2021.