**Help Mr. K the Car Thief! Kaggle Competition**

Name – W. W. Navod Sachindra Perera

Student ID –

* Cardiff - st20284595
* ICBT - CL/BSCSD/26/10

Lecturer – Mr. Roy Ian

Contents

[1. Introduction to Deep Learning 3](#_Toc161152680)

[2. Literature Review 4](#_Toc161152681)

[3. Exploratory Data Analysis (EDA) 7](#_Toc161152682)

[4. System Architecture & What Makes XGBRegression. 9](#_Toc161152683)

[5. Full Model Evaluation & Implementation Details 12](#_Toc161152684)

[6. Conclusion 13](#_Toc161152685)

[7. References 14](#_Toc161152686)

[Figure 1: Correlation Matrix 9](#_Toc161152687)

[Figure 2: User Interface1 10](#_Toc161152688)

[Figure 3: User Interface2 10](#_Toc161152689)

[Figure 4: User Interface3 11](#_Toc161152690)

## Introduction to Deep Learning

Deep learning is an ingenious approach that teaches computers to learn from experience, using vast amounts of data and a special system called neural networks. Think of these neural networks as the computer's brain, made up of layers that work together to understand the world [1].

A neural network is like a complex stack of layers within the computer's brain, where each layer consists of tiny units called neurons. These are inspired by the neurons in human brains. When teaching the computer about something new, like recognizing animals or understanding languages, data passes through these layers in a fascinating journey. Initially, the early layers focus on simple aspects such as identifying edges and colors in images. As the data progresses to deeper layers, it starts recognizing more complex features, like shapes and patterns. For instance, in recognizing a cat, one layer might detect shapes that resemble ears, while deeper layers might combine various features to conclude, "This looks like a cat." [1]

The learning process in deep learning is quite remarkable. By feeding the computer a multitude of examples, such as thousands of pictures labeled as "cat," it begins to identify patterns and characteristics of what defines a cat. The computer makes initial guesses, learns from its mistakes, and gradually improves its ability to correctly identify new images of cats. This process of learning from a vast amount of labeled data allows the computer to understand and make decisions based on its learning, all by itself.

The capabilities of deep learning extend far beyond recognizing images. It has the power to transform speech into text, enabling phones and computers to understand spoken language and respond in kind. It can also comprehend and generate human-like text, which is useful for translating languages or summarizing information. Additionally, deep learning is essential to autonomous cars because it enables them to make safe driving judgements. It has even conquered challenging games, demonstrating its capacity for critical thought and strategy [2].

Computers' interactions with the outside world are being completely transformed by deep learning, which gives computers more intelligence and intuition by imitating human learning and thought processes. Future technologies might benefit greatly from this, including more intelligent personal assistants and robots that are capable of navigating and comprehending their surroundings like humans.

## Literature Review

Predicting the price of cars, especially used ones, is a critical task that combines knowledge from various fields and utilizes multiple data attributes. This process aids buyers and sellers in making informed decisions and understanding market trends. Various studies have been conducted to achieve this purpose using Machine Learning and Deep Learning, and to enhance the performance of existing models. Linear regression has been widely used in the existing literature.

One study focused on developing a robust price prediction system for used cars, recognizing the growing significance of the second-hand vehicle market [3]. This research emphasized the multifaceted nature of pricing determination, considering factors such as manufacturing year, original retail price, fuel type, seller type, transmission, and ownership history. Employing regression analysis, particularly linear regression, the study aimed to generate continuous price predictions tailored to the specific attributes of individual vehicles. By incorporating a user-friendly interface, the proposed system allows users to input relevant information and obtain accurate price estimates. This study underscores the utility of regression models in addressing the complexity of used car pricing, offering a practical solution for facilitating transactions in the automotive marketplace.

Meanwhile, another study focuses on predicting car prices in the U.S market using a linear regression model, aiming to aid newcomers in understanding crucial pricing factors in the automobile industry [4]. The research involved a thorough analysis encompassing data cleaning, exploration, visualization, feature selection, and model building. Data for the analysis were collected from the web portal fred.stlouisfed.org using a Python/Jupyter-based web scraper. The study adopted a problem-solving approach divided into five parts, including data understanding, cleaning, preparation, feature selection, and model building. Examination plots revealed symmetrically placed points along diagonal and horizontal lines, indicating alignment of observed against forecast values and residuals versus projected values, respectively. The study identified outliers and assessed model assumptions, providing insights into the predictive accuracy and potential improvements in the linear regression model.

In another study it has been investigated the effectiveness of three regression algorithms—linear regression, lasso regression, and ridge regression—for predicting the prices of used cars [5]. The research underscores the significance of considering various attributes to ensure reliable and accurate predictions. Utilizing Python libraries and machine learning tools, the study compared the accuracies of the three algorithms. Results revealed accuracies of 83.65%, 87.09%, and 84.00% for linear regression, lasso regression, and ridge regression, respectively. The lasso regression algorithm demonstrated the highest accuracy, leading to the selection of this method for predicting car prices. The study contributes to the understanding of regression techniques in the context of car price prediction and emphasizes the importance of algorithm selection in enhancing prediction accuracy.

Random forest is another frequently used algorithm in this regard. These are tree-based algorithms. One research project explored the impact of renewable energy cars on the used car market, noting the increased interest in electric cars supported by governmental policies [6]. This shift necessitates accurate price prediction models to assist consumers in making cost-effective decisions regarding buying or selling used cars. The study compared the accuracy of the Random Forest and Decision Tree techniques for automobile price prediction. The outcomes showed that the Random Forest approach outperformed the Decision Tree method in terms of accuracy, coming in at 72.13% as opposed to 67.21% for the latter. According to this research, the Random Forest approach could provide a more trustworthy instrument for predicting automobile prices in the dynamic automotive industry.

In some other studies, the authors have developed many models for the same purpose & compared their performance to derive the best model for that scenario. One research employed three separate machine learning techniques—Random Forest, Support Vector Machine, and Artificial Neural Networks—to create a prediction model for secondhand automobiles in Bosnia and Herzegovina [7]. Together, these strategies were applied to maximize the advantages of each approach. The study used data scraped from the web portal autopijaca.ba to train the model. By comparing the performance of these algorithms, the research aimed to identify the most suitable approach for the dataset at hand. The ensemble model was integrated into a Java application and, upon evaluation with test data, achieved an accuracy of 87.38%. This study highlights the importance of selecting appropriate machine learning techniques and the benefits of using ensemble methods to improve prediction accuracy.

In another study it has been investigated the effectiveness of three regression algorithms—linear regression, lasso regression, and ridge regression—for predicting the prices of used cars [5]. The research underscores the significance of considering various attributes to ensure reliable and accurate predictions. Utilizing Python libraries and machine learning tools, the study compared the accuracies of the three algorithms. Results revealed accuracies of 83.65%, 87.09%, and 84.00% for linear regression, lasso regression, and ridge regression, respectively. The lasso regression algorithm demonstrated the highest accuracy, leading to the selection of this method for predicting car prices. The study contributes to the understanding of regression techniques in the context of car price prediction and emphasizes the importance of algorithm selection in enhancing prediction accuracy.

The global auto industry saw a decline in vehicle production after the COVID-19 epidemic in 2020–2021 and it led to an increase in the used car market. Since then, there has been an increase in interest in vehicle price prediction, which has been facilitated by new advancements in online platforms that provide an enormous amount of data on the factors influencing used car values. In another study the aim of this work is to develop a machine learning model that can predict the approximate cost of a used car. Since the model uses the "Saudi Arabia Used Cars" Dataset, which is taken from the Syarah platform and made available on Kaggle, it is a helpful tool for both buyers and sellers in evaluating the market worth of used autos [8]. Three distinct machine learning techniques—Linear Regression, Random Forest, and XGBoost—were applied, yielding MSE scores of 0.15, 0.10, and 0.19, respectively. Amazingly, the Random Forest Regressor method outperformed all other algorithms, demonstrating superior performance across all evaluated measures, including R-squared, MSE, and RMSE. This work shows how machine learning can be used to estimate used car prices and highlights the potential of Random Forest as the optimal method for these types of tasks.

In a study focusing on the price prediction of used BMW cars in Taiwan, significant attention was given to the expanding market of used cars within the automotive industry [9]. Nine key parameters, including model, registration year, and transmission style, were analyzed to develop a price prediction system. The data were divided into three subsets: one for comparing algorithm results, another for model optimization using an optimization algorithm, and a third for validation during the prediction process. Through k-fold cross-validation to mitigate overfitting and selection bias, a model combining random forest and k-nearest neighbors’ algorithms with optimization algorithms such as gray wolf optimizer, multilayer perceptron, and membership function optimization was successfully established. Notably, the prediction results showcased promising performance metrics, including a mean square error of 0.0978, root-mean-square error of 0.3128, mean absolute error of 0.1903, and a coefficient of determination of 0.9249. This study underscores the effectiveness of employing multiple algorithms and optimization techniques for enhancing car price prediction accuracy.

Meanwhile there are some efforts that have made some innovative approaches such as combining existing models which are called ‘Ensemble Models’. An iterative framework combining XGBoost and LightGBM to address the challenge of low prediction accuracy in used car prices, especially when dealing with numerous features and large datasets is introduced in another study [10]. The framework involves data processing, training a deep residual network, and incorporating predicted results as new features for further training. Experimental results demonstrate that the iterative framework outperforms random forest and deep residual network models in terms of prediction accuracy. Additionally, combining the iterative framework with existing methods significantly improves prediction performance across different models. The study offers a novel approach to enhancing prediction accuracy in the used car market, providing valuable insights for future research in predictive modeling.

The traditional offline used automobile selling models have had difficulties in adapting to the changing requirements of consumers due to the rapid expansion of the mobile Internet. As a result, used automobile trading websites have become popular as a suitable response to the current circumstances. The evaluation of used vehicle prices is a crucial step in the second-hand automobile trading process as it provides a fair and accurate pricing system that accurately reflects the genuine nature of the market. To address this, a comprehensive investigation was conducted into the linear correlation between various vehicle parameters, conditions, transaction factors, and used car prices [11]. Grey relational analysis was employed to filter feature variables affecting used car prices, while the traditional BP neural network was optimized through the integration of the particle swarm optimization algorithm. Subsequently, a novel used car price prediction method, termed PSO-GRA-BPNN, was proposed. Results revealed a significant linear correlation between new car price, engine power, and used car price, as well as between new car price, displacement, mileage, gearbox type, fuel consumption, and registration time. Comparative analysis against traditional BPNN, multiple linear regression, random forest, and support vector machine regression models demonstrated superior performance of the PSO-GRA-BPNN model, with a Mean Absolute Percentage Error (MAPE) of 3.936%, 30.041% lower than the error of the other three models. Additionally, the Mean Absolute Error (MAE) of the PSO-GRA-BPNN model exhibited a maximum reduction of 0.622 compared to the other models, with R and R-squared values reaching up to 0.998 and 0.984 respectively. Despite the longest training time of 94.153 seconds, the overall prediction efficacy of the PSO-GRA-BPNN model outperformed other used car price prediction models, offering a novel approach for enhancing the accuracy of used car evaluations.

## Exploratory Data Analysis (EDA)

A critical phase in the data modelling process is exploratory data analysis, or EDA. In order to influence future modelling techniques, it enables data scientists and analysts to comprehend the structure, trends, and anomalies within their data. Making educated judgements regarding feature selection, data cleaning, and the kind of model to deploy requires having a thorough understanding of the distribution, variability, and interaction between variables—all of which are facilitated by exploratory data analysis (EDA). Visualising data, spotting outliers, dealing with missing information, and comprehending the association between variables are frequently included in this process. One may make sure the model is strong, dependable, and accurate by carefully examining the dataset.

* 1. **Handling Missing Values**

The initial step involved checking for null values across the dataset. Null values can distort the predictive modeling process, leading to inaccurate predictions. However, since XGBRegressor is good in handling null values, they were kept as it was.

* 1. **Feature Engineering**

New features were introduced to enhance the model's predictive capability. These features include Clean Title, HP (Horsepower), Turbo, EngineType, Cylinder, and V (engine configuration). The extraction of these features was performed using custom Python functions applied to the existing data columns.

**getCylinder(engineStr)**: This function searches a string for the word "Cylinder" and returns the word immediately preceding it, which is assumed to represent the number of cylinders in an engine. If "Cylinder" is not found, it returns "0".

**getV(engineStr)**: This function parses the input string engineStr to find and return the numeric value associated with an engine configuration that ends with the letter "V" (typically indicating a V-type engine). It searches through each word in the input string for a word that ends with "V" and returns the numeric part of that word, stripping the "V" character. If no such pattern is found, it returns "0".

**getHP(engineStr):** This function looks for and extracts the horsepower (HP) value from the input string engineStr. It iterates through each word in the string, looking for a word that contains "HP". Upon finding such a word, it returns the numeric part by removing the "HP" text. If no horsepower information is found, it returns "0".

**engineType(engineStr):** This function identifies the engine type based on specific keywords present in the input string engineStr. It checks for the presence of keywords such as "Gasoline", "Diesel", "Electric", "DOHC", and "Flex Fuel" in the string, assigning a unique numeric code (1 through 5) to each engine type. If none of these keywords are found, it defaults to "0".

**isTurbo(engineStr):** This function finds whether turbocharging is included in the engine specified in engineStr. It looks for the word "Turbo" in the string. It returns "0" if "Turbo" cannot be located. In order to differentiate between engines that are single-turbocharged ("1") and twin-turbocharged ("2"), if "Turbo" is detected, it then looks for "Twin Turbo."

These functions were applied to both training and test datasets to create new columns, enhancing the dataset with information crucial for predicting car prices.

* 1. **Conversion of Categorical Fields**

The categorical fields were converted into numeric formats, enabling the machine learning algorithms to process them. This conversion was achieved by first encoding engine and transmission types as binary features, indicating the presence or absence of each type within the data. Following this, a LabelEncoder was applied to several categorical columns, transforming them into a numerical format. This step is essential for feeding categorical data into machine learning models, which typically require numerical input.

* 1. **Correlation Analysis**

Finally, a correlation analysis was conducted to identify the relationship between the newly created features (and existing features) and the target variable, the price of the car. This analysis is crucial to understanding how each variable impacts the price, helping in feature selection for the predictive model. However, it was observed that there were no highly correlated features with the price, indicating that a more complex model or additional features might be necessary to accurately predict car prices.

Through these steps of EDA, including handling missing values, feature engineering, conversion of categorical fields, and correlation analysis, the dataset was thoroughly prepared for building a predictive model. This meticulous preparation ensures that the model is built on a solid foundation of clean, relevant, and appropriately encoded data, setting the stage for more accurate and reliable predictions.

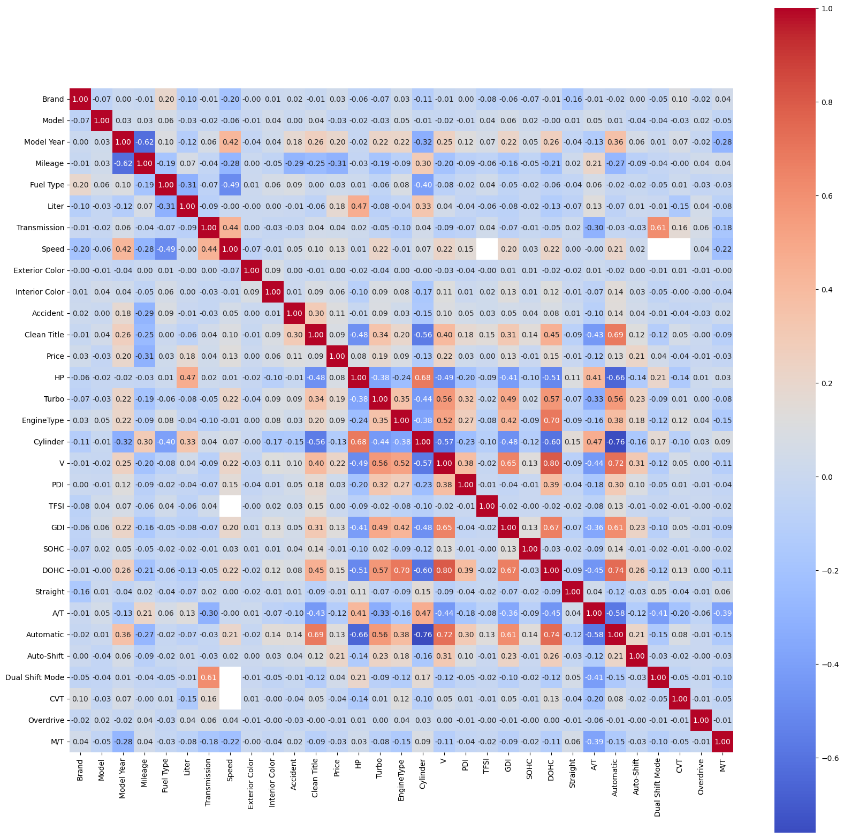


Figure 1: Correlation Matrix

## System Architecture & What Makes XGBRegression.

* 1. **System Architecture**

Working with an XGBoost regressor model requires a system architecture made up of a number of interconnected parts and procedures that work together to transform raw data inputs into predictions that can be used or further examined. The design begins with a data preparation layer that gathers data from several sources and performs preprocessing, cleaning, and feature engineering on it. This step is crucial because it checks that the data the model will use to train is relevant and of high quality. It also fixes problems like missing values, encodes categorical variables, and scales numerical values to increase the model's learning effectiveness.

The model training layer comes first, after data preparation. To aid in the model's learning and assessment process, the processed dataset is split into subsets for testing, validation, and training at this step. Next, a set of hyperparameters are initialized for the XGBoost regressor model. These parameters may be adjusted in later rounds based on performance metrics assessed using the validation set. Gradient boosting is the foundation of the XGBoost algorithm. In gradient boosting, decision trees are built one after the other with the goal of correcting the faults of the previous trees, hence improving the model's predictions over time.

Once the model is trained and fine-tuned to achieve satisfactory performance, the deployment and inference layer comes into play. This involves integrating the model into a production environment where it can receive new data inputs, preprocess them as required, and generate predictions. The deployment setup needs to be robust, ensuring that the model can handle new data efficiently and scale as required by the application demands. Additionally, maintaining model performance over time is crucial, necessitating mechanisms for continuous monitoring and periodic updates or retraining with new data to address concept drift or changes in data patterns.

The infrastructure supporting these activities is pivotal, comprising both the hardware and software necessary for model training, evaluation, and deployment. Given the computationally intensive nature of training complex models like XGBoost, especially on large datasets, access to adequate computational resources (such as GPUs or distributed computing environments) is essential. The software stack typically includes the XGBoost library along with other data science and machine learning tools for data manipulation, model training, and evaluation, often within a Python ecosystem. The choice of tools and technologies is influenced by the specific needs of the project, including considerations for scalability, performance, and ease of integration into existing systems.

A screenshot of a car price prediction

Description automatically generated

Figure 2: User Interface1

A screenshot of a computer

Description automatically generated

Figure 3: User Interface2

A screenshot of a computer

Description automatically generated

Figure 4: User Interface3

* 1. **What Makes XGBRegressor set apart from other tree-based algorithms.**

Extreme Gradient Boosting, or XGBoost, is a well-known tree-based machine learning technique. Its regression-specific counterpart, XGBRegressor, has a few unique characteristics. Its incorporation of regularization methods within the gradient boosting architecture is one of its noteworthy features. During model training, this comprises adding L1 (Lasso) and L2 (Ridge) regularization terms to the objective function. By reducing overfitting, this kind of regularization improves the model's capacity to generalize well to new data. In addition, XGBoost builds decision trees using an optimized splitting technique. Rather than systematically looking for the best split points, it makes use of a "greedy" algorithm in conjunction with methods such as quantile sketching and histogram-based algorithms. Because of this method's significant reduction in computing time, XGBoost is exceptionally efficient, especially when working with huge datasets.

The ability of XGBRegressor to manage missing values internally throughout both the training and prediction stages is another unique feature. When training a model, XGBoost can automatically determine the optimal method for managing missing data, in contrast to some other tree-based methods that need preprocessing steps to impute missing values. To further reduce overfitting and boost computational effectiveness, XGBoost includes early halting and tree trimming techniques. With the aid of early stopping criteria and controls over tree depth and complexity, XGBoost makes sure the model runs as efficiently as possible without using needless computation.

Another advantage of XGBoost is that it supports distributed and parallel computation. By utilizing distributed computing frameworks such as Apache Spark and Dask, together with multicore processors, XGBoost can effectively expand to manage enormous datasets and computationally demanding jobs. Furthermore, XGBoost provides versatility by enabling users to customize the algorithm to particular regression problems or optimization objectives through the use of customizable objective functions and assessment measures. This versatility further cements XGBoost's standing as the go-to option for data science and machine learning applications by making it appropriate for a wide range of regression issues across several domains.

## Full Model Evaluation & Implementation Details

Mean Squared Error (MSE) is the statistic used to assess the entire model. This measure aids in assessing how effectively the model predicts results. Given that the average price data presented in the dataset is around $44,553.19, the MSE value of 1890.216 for the training data is appropriate. The MSE value for the testing data, however, is significantly larger at 12,705.41. The model may be overfitting if the MSE for the testing data is so high, which would indicate that it fits the training data too well and does not generalize well to new, unknown data.

Although tree-based models such as XGBRegressor are usually good at preventing overfitting, this example shows that there could be more space for development. Tree-based models sometimes have overfitting tendencies, however XGBRegressor is usually resistant to these problems. In spite of this, the testing data observed high MSE suggests that the model may still need to be adjusted.

To address this overfitting concern, it may be necessary to refine the model by adjusting certain parameters or implementing additional regularization techniques. This could involve optimizing parameters related to tree complexity, learning rate, or regularization terms. By fine-tuning the model in this manner, it's possible to improve its generalization ability and reduce overfitting tendencies, ultimately leading to more accurate predictions on unseen data.

## Conclusion

In conclusion, this document has offered a comprehensive exploration of implementing and evaluating an XGBRegressor model for regression tasks, contextualized within existing literature on car price prediction using various models. At the outset, the literature review provided insights into prior research works, highlighting the diverse range of models employed for car price prediction, including linear regression, decision trees, random forests, and neural networks. Each model was scrutinized for its strengths and limitations, offering valuable context for the subsequent discussion on the suitability of XGBRegressor.

Following the literature review, the document delved into the system architecture of XGBRegressor, highlighting its distinctive features such as regularization techniques, optimized splitting strategy, and handling of missing values. Moreover, it detailed the data preprocessing tasks undertaken, including the utilization of specific functions to clean and preprocess the data, as well as the conversion of categorical features into numerical representations. These preprocessing steps were crucial for preparing the dataset for training the XGBRegressor model, ensuring optimal performance and accuracy.

Subsequently, the model evaluation process was detailed, employing the Mean Squared Error (MSE) metric to assess performance on training and testing datasets. While the model demonstrated reasonable accuracy on the training data, a higher MSE value for the testing data hinted at potential overfitting, prompting further refinement recommendations.

By integrating insights from the literature review with practical implementation and evaluation processes, this document offers a holistic perspective on the utilization of XGBRegressor for car price prediction. Moving forward, practitioners can leverage this synthesis of theory and practice to inform their decision-making processes, optimizing model performance and advancing the field of predictive analytics in automotive industries and beyond.

## References

[1] A. Banafa, “What is Deep Learning?,” in *Quantum Computing and Other Transformative Technologies*, 2023. doi: 10.1201/9781003339175-12.

[2] Kevin Murnane, “What Is Deep Learning And How Is It Useful?,” *Forbes*, 2016.

[3] S. MUTİ and K. YILDIZ, “Using Linear Regression For Used Car Price Prediction,” *International Journal of Computational and Experimental Science and Engineering*, vol. 9, no. 1, 2023, doi: 10.22399/ijcesen.1070505.

[4] H. Mammadov, “Car Price Prediction in the Usa By Using Linear Regression,” *Research Gate*, vol. 11, 2022.

[5] Prof. P. Bharambe, B. Bagul, S. Dandekar, and P. Ingle, “Used Car Price Prediction using Different Machine Learning Algorithms,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 4, 2022, doi: 10.22214/ijraset.2022.41300.

[6] P. H. Putra, A. Azanuddin, B. Purba, and Y. A. Dalimunthe, “Random forest and decision tree algorithms for car price prediction,” *Jurnal Matematika Dan Ilmu Pengetahuan Alam LLDikti Wilayah 1 (JUMPA)*, vol. 3, no. 2, 2023, doi: 10.54076/jumpa.v3i2.305.

[7] E. Gegic, B. Isakovic, D. Keco, Z. Masetic, and J. Kevric, “Car price prediction using machine learning techniques,” *TEM Journal*, vol. 8, no. 1, 2019, doi: 10.18421/TEM81-16.

[8] M. Gollapalli *et al.*, “Intelligent Modelling Techniques for Predicting Used Cars Prices in Saudi Arabia,” *Mathematical Modelling of Engineering Problems*, vol. 10, no. 1, 2023, doi: 10.18280/MMEP.100115.

[9] P. H. Kuo, S. Y. Chen, and H. T. Yau, “A Multimodel Transfer-Learning-Based Car Price Prediction Model with an Automatic Fuzzy Logic Parameter Optimizer,” *Computer Systems Science and Engineering*, vol. 46, no. 2, 2023, doi: 10.32604/csse.2023.036292.

[10] B. Cui, Z. Ye, H. Zhao, Z. Renqing, L. Meng, and Y. Yang, “Used Car Price Prediction Based on the Iterative Framework of XGBoost+LightGBM,” *Electronics (Switzerland)*, vol. 11, no. 18, 2022, doi: 10.3390/electronics11182932.

[11] E. Liu, J. Li, A. Zheng, H. Liu, and T. Jiang, “Research on the Prediction Model of the Used Car Price in View of the PSO-GRA-BP Neural Network,” *Sustainability (Switzerland)*, vol. 14, no. 15, 2022, doi: 10.3390/su14158993.