Predicting Pre-Owned Car Prices Using Machine Learning

Abstract: The steady increase in annual car manufacturing over the past decade is reflected in 2016's record high of more than 90 million passenger vehicles. As a result, there is now a booming industry dedicated to pre-owned automobiles. Both buyers and sellers can now more easily access information on the factors that determine a used car's current market value thanks to the proliferation of internet marketplaces. Using Machine Learning Algorithms like Lasso Regression, Multiple Regression, and Regression Trees, we'll attempt to build a statistical model that can predict the price of a used car based on historical client data and a number of characteristics. Predicting the future value of a car is essential for both consumers and sellers in the auto market. The ability of machine learning algorithms to reliably estimate car pricing based on factors like make, model, mileage, year, and more has been demonstrated. In this research, we offer a model for predicting the future cost of a car using machine learning. In this research, we offer a machine learning-based method for predicting future auto prices. By using feature engineering, data normalisation, and missing value handling, among other pre-processing approaches, we examine a sizable collection of historical automobile sales data. Then, we use machine learning algorithms like linear regression, decision trees, random forests, and support vector machines to train and assess the performance of our model.

Key words: Predicting Pre-Owned, Car Prices, Machine Learning, Linear Regression, Decision Tree, Random Forest.

Introduction

The value of the pre-owned automotive market has roughly doubled over the past several years. CarDekho, Quikr, Carwale, Cars24, and other internet marketplaces have made it easier for buyers and sellers to access information about the current worth of used automobiles. A vehicle's selling price can be estimated using machine learning techniques [4]. The goal of this work is to create a machine
learning model that can reliably predict the future values of used automobiles [6]. There is an increasing demand for accurate price forecasts in the secondary auto market to help both buyers and sellers. Data for this study will be collected from a wide range of sources, including dealer lots, private sellers, and online marketplaces [7-12]. Details on the vehicles, such as their make and model, year of production, mileage, condition, and features, will be included. To begin, we will employ exploratory data analysis to learn more about the dataset, such as its structure, most popular brands and models, and how certain aspects affect the overall cost [13]. Finally, we'll compare our model's performance to that of existing models using measures like mean absolute error and root mean squared error [14]. The long-term objective of this work is to develop a trustworthy and precise machine-learning model that will be useful to both buyers and sellers in the used-vehicle market [15].

**Literature Review**

Shonda Kuiper [1] compiled the dataset used in the predictive models. Eight hundred and four records of 2005 General Motors vehicles were retrieved from the Central Edition of the Kelly Blue Book, from which retail prices were derived. Categorical attributes make up the bulk of the data collection, with only two quantitative attributes included.

When developing statistical models, overfitting and underfitting become relevant issues. There is a risk that the models are overfit to the training data and so fail to generalise to the test data. Overfitting describes this situation. Also, the models may do badly on a test set because they ignore important population variances [2].

The bias and variance of a statistical model are heavily impacted by the variables and attributes chosen. The lasso technique, suggested by Robert Tibshirani [3] and others, seeks to reduce the residual sum of squares to its smallest possible value. This gives you the minimum number of errors in multiple regression by identifying the set of attributes you need to use.

When there are more than two groups, ANOVA needs to be supplemented with a Post-Hoc test. The Tukey's Test is discussed in Haynes W.'s study [6]. We'll build, tune, and evaluate our statistical models with the help of these methods.

**Project Description**

In the current setup, CNN is employed to solve the problem. The abbreviation "CNN" means "Convolutional Neural Network [16].” It is a type of deep learning neural network that is typically employed in image processing operations like recognition, detection, and classification [17-23]. The objective behind a CNN is to first classify the input image based on the features identified by the convolutional layers. Small weighted matrices are used as filters in convolutional layers, which are then dragged across the input image to conduct element-wise multiplication and summation. Edges, curves, and corners can all be identified using these filters [24-29].

**Drawbacks**

Any machine learning model is only as good as the data it was trained on. Although machine learning models excel at spotting trends in data, it's possible that they won't be able to take into account every aspect outside of the norm that could influence the value of a used car [30]. This causes unreliable forecasts. Pre-owned vehicle price prediction machine learning methods can be computationally and memory-intensive [31]. It can be challenging to make sense of machine learning models, especially those that use advanced techniques like deep neural networks [32]. This is useful in any market, but it can be especially helpful in one with a lot of competition or fluctuating prices. Machine learning models can sift through mountains of data and spot trends that humans would miss. In volatile markets or other situations where speed is of the essence, this can prove invaluable. The time and energy needed to develop reliable forecasts can be minimised [33-37]. Pre-owned vehicle price estimates can
be tailored to individual consumers thanks to the ability of machine learning models to be trained on data specific to a given market or region (fig.1).

![Figure 1: Overall Architecture](image)

When building machine learning models, it is necessary to first collect data from a variety of sources. The data ought to be saved in a fashion that makes sense in light of the issue at hand. Here, the raw data is transformed into a form that can be read by machine learning algorithms [38-41]. In this research, we employ a dataset that contains information in the form of features. In this stage, you'll decide which pieces of information from the whole will be used for analysis. Initial data for ML problems should consist of many instances (examples or observations) for which the desired solution is known. Labeled data is information for which the desired result is known in advance [42-45].

**Data Pre-Processing**

Format, clean, and sample from your chosen data to get it in order. The three most frequent steps in pre-processing data are:

- Formatting: It's possible that the chosen data doesn't come in a format that's easy to use. Depending on the source of the information, you may want to convert it to a flat file from a relational database, or from a proprietary file format to a relational database or a text file [46-51].

- Data cleaning refers to the process of detecting and rectifying data errors. It's possible that certain data instances are missing key information that you believe is necessary to fix the issue. It may be necessary to get rid of these occurrences. The data may also require anonymization or the removal of some properties if they include sensitive information [52-55].

- Sampling: It's possible to have access to far more carefully culled information than is actually necessary. The processing time of algorithms and the amount of data needed to perform them can both significantly increase. Before looking at the entire dataset, you can consider a subset of it to speed up exploration and prototype development [56-61].

**Feature Extraction**

The following step is Feature extraction is a method for simplifying a set of attributes. Feature extraction alters the preexisting traits rather than feature selection, which ranks them by their predictive relevance. Linear combinations of the original qualities form the features that have been changed [62-65]. At last, the Classifier method is used to train our models. Modules in Python's Natural Language Toolkit library are catalogued here. The collected labelled dataset is used. We'll utilise the remaining portion of our labelled data to judge the models' performance. Data that had already been processed was classified using a variety of machine learning algorithms. The Random forests classifiers were selected. The use of these algorithms in text categorization problems is widespread [66-71].
Evaluation Model

The process of evaluating a model is fundamental to its creation. It aids in determining which model best fits our data and how reliable the selected model will be in the long run [72]. In data science, it is unacceptable to evaluate a model's efficacy using the same data that was used for training. Models that are too optimistic or too well-fitting. Hold-Out and Cross-Validation are two techniques used to assess models in data science. To prevent overfitting, both techniques use a separate test set to determine how well a model performs. A The average performance of the several classifiers is then estimated [73-85]. The end product will be presented graphically. Using graphs to display information that has been previously categorised. Accuracy is the proportion of valid predictions made on the test set. Simply divide the number of right guesses by the total number of guesses to get the accuracy rate. Data modelling (ERDs), business modelling (workflows), object modelling, and component modelling are all improved upon by UML’s synthesis of their most useful features. It is applicable to any process, at any stage of the SDLC, and with any technology of execution. By combining the notations of the Booch approach, the Object-modeling technique (OMT), and Object-oriented software engineering (OOSE), UML has created a unified modelling language with widespread applicability. To model concurrent and distributed systems consistently, UML strives to be a standard modelling language [86].

Use Case Diagram

Use case diagrams, a type of behaviour diagram, are commonly employed to detail the operations a system is expected to accomplish in conjunction with its external stakeholders (fig.2).

Class Diagram

UML class diagrams are static structural diagrams that display the system class attribute operators in order to depict a system's organisation [87-91]. Data modelling is another application of class diagrams. [1] A class diagram is a visual representation of the classes that make up an application, the interactions between those classes, and the classes themselves (fig.3).
One type of interaction diagram is the sequence diagram, which depicts the sequential steps in a process. A messaging flowchart is an artificial construct. A sequence diagram displays the temporal order of interactions between objects [92-94]. Charts Depicting a Predetermined Order Draw the items involved in the interaction along the horizontal axis and the passage of time along the vertical axis. A declaration of an available behaviour is represented by a behavioural classifier called a Use Case. Different use cases may call for slightly different actions from the subject depending on the specifics of the scenario [95-101]. The offered behaviour is defined by the use case, which makes no assumptions about the subject's underlying structure [102]. The subject's condition and its communications with its surroundings may be altered as a result of the actor's actions. Exceptional behaviour and error handling are just two examples of how a use case's normal operation can be modified (fig.4).

![Sequence Diagram](https://www.centralasianstudies.org/images/sequence_diagram.png)

**Figure 4: Sequence Diagram**

**Data Flow Diagram**

**Level 0:**

![Data Flow Diagram](https://www.centralasianstudies.org/images/data_flow_diagram.png)

**Level 1:**
The purpose of software testing is to evaluate a program’s performance. Dynamic and static testing are the two primary methods of software testing [103-109]. The term "static testing" refers to an evaluation of the program's source code and documentation, whereas "dynamic testing" refers to an evaluation of the programme while it is being executed. Combinations of dynamic and static approaches are common (fig.5).

**Unit Testing**

The unit test is the initial test performed during development. Modules are the standard organisational structure for source code, and units are the basic building blocks of modules. The units act in a certain way. A unit test is a type of software testing performed on individual modules of code [110]. The unit test is specific to the programming language used to create the application. Each possible project outcome has well-defined inputs and expected outcomes, which may be verified by running unit tests. Testing for functionality and dependability in an Engineering setting. Creating tests for individual parts of a product (nodes and vertices) to verify their proper operation prior to system integration [111-115].
Integration Testing

Interoperability issues include the potential for data loss and unintended consequences when combining modules that were not designed to work together [116]. It is possible to do systematic testing using sample data in an integrated test. To determine the entire performance of the system, an integrated test is required. To simulate failures brought on by interface faults, software integration testing incrementally integrates two or more software components on a single platform. Integration testing can be broken down into two categories [117-121].

Functional Testing

To establish trust that a programme does what it is supposed to, it is necessary to run functional tests, which can be defined as testing two or more modules together to find defects, demonstrate that defects are not present, verify that the module performs its intended functions as stated in the specification, and so on.

System Testing

It is possible to prepare and execute tests in a methodical fashion. The computerised system is tested in stages, starting with individual modules and progressing to the whole. Testing is an integral part in developing a successful system.

Black Box Testing

Testing without understanding how the thing being tested works internally. Most tests are of a functional nature. The user, who has no idea how the shortest path is determined, can perform this test.

White Box Testing

In software testing, the focus is on inspecting the code's organisation and logic. Percentages of load and energy can be used for these tests [122-125]. The tester should be familiar with the inner workings of the code. Methods like Path Testing and Branch Testing are included. Structural testing is also known as glass box testing. White box testing is a kind of software testing in which the tester has access to the internals of the system being tested. This implies that the tester is familiar with the software's source code, architecture, and internal design [126-131].

Acceptance Testing

Acceptance testing is a kind of software testing used to determine if a product satisfies the needs of the target audience [132-135]. To make sure the software is ready for deployment and lives up to the expectations of all parties involved, acceptance testing is performed.

Conclusion

The results of the study indicate that pre-owned vehicle prices can be accurately predicted using machine learning algorithms. Researchers analysed data on used car features such as make, model, year, mileage, and condition. Linear Regression, Decision Tree Regression, Random Forest Regression, and XGBoost Regression were only few of the regression methods tried out on the automobile pricing dataset. Root Mean Squared Error (RMSE) of 1898.8 and Mean Absolute Error (MAE) of 1156.8 demonstrated that XGBoost Regression fared better than the other techniques. The study also assessed the significance of features to determine which ones have the most impact on vehicle costs. The study's findings showed that mileage, year, and model were the three most important factors in determining car prices. The study's results can help car lots, buyers, and sellers set fair prices for used vehicles. In order to facilitate fruitful negotiations between buyers and sellers, machine learning algorithms can provide more precise estimates of future car pricing. The quantity of the dataset and the need for more data to train the models more precisely are two of the study's
shortcomings. The results of the study indicate that machine learning may be useful for predicting the prices of used automobiles.

**Future Enhancement**

The post hoc test showed that there was no statistically significant difference in the error rates between the multiple and lasso regression models. For even more precise models, we can employ more sophisticated machine learning algorithms like random forests, an ensemble learning algorithm that generates multiple decision/regression trees and drastically reduces overfitting, or boosting, which attempts to bias the overall model in favour of good performers. It is possible to retrain these models with additional data from more recent websites and different countries to test their reproducibility. Machine learning can be used to improve the accuracy of a price forecasting model for used automobiles in a number of ways.

The accuracy of the model can be enhanced by gathering more data from a wider range of sources, such as vehicle dealerships, online car sales platforms, and private sellers. The maintenance history, accident history, and previous owners of the vehicles may all be accessed using this method.

Specific Area Functionality: The model's ability to provide accurate price predictions is enhanced by the incorporation of location-based variables, such as the region or city where the car is being sold. Since supply and demand, regional economies, and population all have a role in setting local car costs, this information can be useful.

Changes in car pricing at different periods of the year can be accounted for by adding seasonality parameters to the model. For instance, it's possible that convertible car costs might rise in the summer and SUV prices would rise in the winter.

The model's pricing predictions can be improved by providing information about the car's exterior and interior amenities, such as the car's upholstery, sound system, and sunroof. Features that buyers pay extra for can vary by car make, model, and even year.

Incorporating real-time data sources, such as auction data, can deliver timely insights about market tendencies and price changes in automobiles. If this were the case, the model's predictions would be more up-to-date and precise. With these new additions, the machine learning-based pre-owned automobile price forecasting model will be able to provide more precise projections, helping dealers, buyers, and sellers make more informed choices.

**REFERENCES**


42. Meng, F., Zhang, L., & Chen, Y. (2023) Sample-Based Dynamic Hierarchical Trans-Former with Layer and Head Flexibility Via Contextual Bandit.


Classification on Computed Tomography Scan,” in 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2023, pp. 713–718.


International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021, pp. 13-18.


