MACHINE LEARNING MODEL TO PREDICT THE PRICES OF USED CARS

AWELE OKOLIE

SCHOOL OF COMPUTING AND DATA SCIENCE

WENTWORTH INSTITUTE OF TECHNOLOGY

BOSTON, MASSACHUSETTS, UNITED STATES

<mailto:okoliea@wit.edu>

ABSTRACT

The objective of this project is to predict the prices of used cars based on various vehicle features such as fuel type, transmission, mileage, engine size, and more. Multiple machine learning models including Random Forest, Gradient Boosting, Linear Regression, and Support Vector Regression (SVR) were evaluated using the [Used Car dataset](https://www.kaggle.com/datasets/jacksondivakarr/sample34?utm_source=chatgpt.com&select=processes2.csv) from Kaggle. Extensive experiments involving feature scaling, feature engineering, dimensionality reduction, and hyperparameter tuning were conducted to enhance model performance. The best results were achieved with a tuned Gradient Boosting model, which yielded a Mean Squared Error (MSE) of **4.4B** and an **R² score of 0.9064**, indicating strong predictive performance. These findings demonstrate the effectiveness of advanced machine learning techniques in providing accurate price estimations, aiding both consumers and automotive businesses in making informed decisions in the used car market.

KEYWORDS

Used Car Prices, Machine Learning, Gradient Boosting, Predictive Modeling.

1 INTRODUCTION

The aim of this project is to develop a predictive model for estimating the price of used cars based on various features such as fuel type, transmission type, mileage, engine size, and others. Predicting car prices accurately is essential for online car dealerships, used car marketplaces, and potential buyers to make informed decisions. With the rise of online platforms for buying and selling vehicles, understanding the factors influencing car pricing becomes crucial for optimizing sales strategies and improving customer satisfaction. The automotive market relies heavily on accurate pricing models to provide competitive prices and gain customer trust. Additionally, a reliable car price prediction model can aid businesses in maximizing profits, helping customers find the best deals, and streamlining the car-buying process. The challenge lies in capturing the nuances in car features that contribute to price variations, including factors such as condition, and market demand, all of which can interact in complex ways.

2 DATA

2.1 SOURCES OF DATA

The dataset used in this project is [Used Car dataset](https://www.kaggle.com/datasets/jacksondivakarr/sample34?utm_source=chatgpt.com&select=processes2.csv) on Kaggle. The dataset contains information about used cars listed for sale. The key features in the dataset are:

* **Car\_Name**: The name of the car model being listed for sale. (Maruti, Hyundai)
* **Year**: The year the car was manufactured.
* **Selling\_Price**: The price at which the owner wants to sell the car. (The target variable)
* **Km\_Driven**: The distance the car has been driven, measured in kilometers.
* **Fuel**: The type of fuel used by the car (petrol, diesel).
* **Seller\_Type**: Defines whether the seller is an individual or a dealer.
* **Transmission**: Defines whether the car has a manual or automatic transmission.
* **Owner**: The ownership status (e.g., First Owner, Second Owner, etc.)
* **seats**: The number of seats in the car.
* **max\_power (in bph):** Maximum power of the car’s engine in brake horsepower (BHP).
* **Mileage Unit**: Unit of mileage (e.g., kmpl).
* **Mileage**: The car’s mileage (fuel efficiency).
* **Engine (CC)**: Engine displacement in cubic centimeters.

2.2 DATASET POTENTIAL BIASES

Class Imbalance in Categorical Features:

* Fuel Type: There is a strong imbalance with CNG (19 cars) and LPG (1 car) being underrepresented, compared to Diesel and Petrol.
* Seller Type: Most cars are listed as Individual (1919 cars), followed by Dealer (169 cars), and Trustmark Dealer (7 cars). There’s a significant imbalance here as well.
* Transmission: The dataset has a high imbalance with Manual transmission cars (1996 cars) far outweighing Automatic transmission cars (99 cars).
* Owner: The First Owner category has most cars (1325), while Test Drive Car has only 1 car, indicating a possible bias towards cars that are initially owned.

Fuel Type and Selling Price:

* CNG has the highest average selling price at 369,789 but only has 19 cars, making this value highly unrepresentative.
* Diesel and Petrol have relatively lower selling prices at 508,503 and 341,262, respectively, with Diesel being more abundant.
* LPG has the lowest selling price at 222,000, but it has only 1 car, making it unrepresentative.

**Seller Type and Selling Price:**

* Dealer has the highest average selling price (559,757), followed by Individual (415,425), and Trustmark Dealer (478,571). The Dealer category is far more represented than Trustmark Dealer, which may lead to an overrepresentation of dealer prices.

**Owner and Selling Price:**

* The average selling price is significantly higher for Test Drive Car (1,350,000), but there is only 1 car in this category, making it an outlier.
* First Owner cars have an average price of 486,206, while the prices for Second Owner and Third Owner cars are lower (336,392 and 291,184, respectively), showing a price decrease with ownership.

2.3 PROPROCESSING STEPS

* Encoding Categorical Variables: One-hot encoding was applied to categorical variables such as Fuel\_Type, Seller\_Type, Transmission, owner, mileage unit to convert them into numerical format.
* Feature Transformation: The Year feature was used to create a new feature called Car\_Age, which was calculated as the difference between the current year (2025) and the Year the car was manufactured. This feature helps capture the car’s age, which is important for price prediction. Additionally, the year and name columns were dropped since they were no longer necessary.
* Handling Missing Data: Any missing values in the dataset were removed using dropna (), ensuring that only complete rows were used for model training. This step helps maintain the quality and integrity of the dataset.
* Filtering Invalid Data: Rows with selling\_price less than or equal to zero were filtered out, as these do not represent valid car sales data.
* Scaling: Numerical features like Km\_Driven, selling\_price was scaled using StandardScaler to standardize the data, ensuring that all features have a mean of 0 and a standard deviation of 1. This allows models to perform optimally by treating each feature equally, especially when they have different ranges.
* Log Transformation: A log transformation was applied to the km\_driven feature using np.log1p () to reduce its skewness, making the data more suitable for linear models by stabilizing variance.
* Feature Engineering: Polynomial features were created using Polynomial Features with a degree of 2 to capture interaction terms between features, potentially improving model performance by accounting for non-linear relationships.
* Handling Noisy Features: Random continuous noise was added to the training and test sets, and random discrete noise (categorical) was also introduced. The categorical noise was then encoded into dummy variables. This experiment tested the model’s robustness against irrelevant or noisy features.

**3 BASELINE PERFORMANCE**

This section provides the training and evaluation of four different regression models to predict the target variable (selling\_price). These models were selected based on their ability to handle both linear and non-linear relationships in the data:

**Random Forest**: A robust ensemble model that works well for both regression and classification tasks by creating multiple decision trees and averaging their predictions. It is known for handling non-linear relationships and interactions between features effectively.

**Gradient Boosting**: Another ensemble model that builds decision trees sequentially, where each tree corrects the errors made by the previous one. This model is also known for producing high performance, especially when dealing with non-linear data.

**Linear Regression**: A simple model that assumes a linear relationship between the features and the target variable. It is easy to interpret but may struggle with complex, non-linear relationships.

**Support Vector Regression (SVR)**: A model based on the Support Vector Machine algorithm, typically effective for both linear and non-linear regression. SVR tries to fit the error within a certain threshold and can handle high-dimensional data.

These models were trained and evaluated on the raw data, providing the following baseline performance metrics:

Random Forest:

* **MSE**: 4920944859.17
* **R²**: 0.8959
* The Random Forest model performed well, with a high R² value, indicating that it explained a significant portion of the variance in the target variable. The MSE is relatively low, which suggests that the model’s predictions are reasonably close to the actual values.

Gradient Boosting:

* **MSE**: 4961681479.86
* **R²**: 0.8950
* Gradient Boosting yielded similar performance to Random Forest, with slightly better R², which means it also captured most of the variance in the data. The MSE is slightly lower than Random Forest, suggesting a slightly more accurate model.

Linear Regression:

* **MSE**: 11492886052.5336
* **R²**: 0.7569
* Linear Regression performed worse than the ensemble methods, with a significantly higher MSE and a lower R² score. This suggests that the linear model is not able to capture the underlying complexity in the data as effectively as the ensemble models.

Support Vector Regression (SVR):

* **MSE**: 47285294008.71
* **R²**: -0.0003
* SVR performed poorly in comparison to the other models, with a very high MSE and a near-zero R² value. This indicates that SVR was unable to model the relationship between the features and the target effectively, likely due to its sensitivity to noise and high-dimensional data.

The baseline performance metrics indicate that the **Random Forest** and **Gradient Boosting** models performed the best, with both models explaining a high percentage of the variance in the target variable (R²around 0.89) and relatively low MSE. On the other hand, **Linear Regression** and **SVR** performed significantly worse, with high MSE and lower R², indicating that they may not be well-suited for this dataset without further adjustments.

These baseline results serve as a foundation for future experimentation, where I will apply feature scaling, transformations, and feature engineering techniques to improve the performance of these models. The goal is to see if these adjustments can help further reduce MSE and increase R², leading to more accurate predictions of car prices.

**4 EXPERIMENTS**

Experiment 1: Feature Scaling

**Objective:**To evaluate how different feature scaling techniques impact model performance, especially for models sensitive to feature magnitudes like SVR and Linear Regression. **Applied three scaling techniques:**

* StandardScaler: Standardizes features by removing the mean and scaling to unit variance.
* MinMaxScaler: Scales features to a given range, usually [0, 1].
* Log Transformation: Applied a logarithmic transformation to skewed features to reduce the effect of outliers and improve linearity.

Results:

* Ensemble models (Random Forest and Gradient Boosting) were relatively unaffected by scaling, as expected.
* Linear Regression showed slight variations, with log transformation slightly reducing R².
* SVR showed very poor performance in all cases, with R² scores near zero or negative. SVR may require more tuning or different kernels to perform well.

Feature scaling did not significantly impact the performance of tree-based models. However, SVR was highly sensitive and performed poorly regardless of scaling, indicating it may not be suitable for this dataset without further optimization.

Experiment 2: Feature Engineering

**Objective:**  
To explore whether engineering new features using polynomial combinations can improve model performance by capturing non-linear relationships.

Generated second-degree polynomial features from existing features (e.g., square and interaction terms).

Results:

* Random Forest and Gradient Boosting models showed a slight improvement in both MSE and R².
* Linear Regression improved substantially (R² increased from 0.75 to 0.85).
* SVR still underperformed with a negative R².

**Conclusion:**  
Polynomial feature engineering notably improved the performance of Linear Regression, making it more competitive with ensemble models. However, SVR still lagged, reinforcing earlier findings.

Experiment 3: Feature Transformations (PCA)

**Objective:**  
To test whether reducing dimensionality using Principal Component Analysis (PCA) can enhance performance by eliminating redundant or noisy features.

Applied PCA to reduce the feature space before training models.

Results:

* Performance dropped across all models, especially ensemble ones.
* Random Forest and Gradient Boosting saw a noticeable decrease in R².
* Linear Regression also dropped to 0.71 R² from ~0.75.
* SVR remained ineffective.

Conclusion:  
PCA did not help and instead reduced model performance. This suggests that the original features already held significant predictive power, and dimensionality reduction likely removed useful information.

Experiment 4: Handling Noisy Features

**Objective:**  
To assess how robust each model is to irrelevant or noisy data.

Introduced synthetic random noise features (not correlated with the target) and re-evaluated model performance.

Results:

* Random Forest and Gradient Boosting saw slight drops in R² (about 0.89 from ~0.895).
* Linear Regression remained mostly stable.
* SVR again performed very poorly.

Conclusion:  
Tree-based models proved robust to noisy features, while Linear Regression was mildly impacted. SVR's performance was already weak and remained unaffected in the negative range. Random Forest and Gradient Boosting are better choices when dealing with noisy or high-dimensional datasets.

MODEL PERFORMANCE ACROSS EXPERIMENTS

|  |  |  |  |
| --- | --- | --- | --- |
| Experiment | Model | MSE | R² |
| Scaling: StandardScaler | Random Forest | 4,992,838,974.66 | 0.8944 |
|  | Gradient Boosting | 4,961,554,109.77 | 0.8950 |
|  | Linear Regression | 11,492,886,052.53 | 0.7569 |
|  | SVR | 47,261,287,251.35 | 0.0002 |
| Scaling: MinMaxScaler | Random Forest | 4,985,749,946.73 | 0.8945 |
|  | Gradient Boosting | 4961,681,479.86 | 0.8950 |
|  | Linear regression | 11, 492,886,052.53 | 0.7569 |
|  | SVR | 47, 265,752,436.67 | 0.0001 |
| Scaling: Log Transform | Random Forest | 4,834, 178,040.56 | 0.8977 |
|  | Gradient Boosting | 4, 940,710,659.47 | 0.8955 |
|  | Linear Regression | 11, 671,162,724.85 | 0.7531 |
|  | SVR | 47,276, 364,115.21 | -0.0001 |
| Polynomial Features | Random Forest | 4, 785,259,727.06 | 0.8988 |
|  | Gradient Boosting | 4, 812,798,151.84 | 0.8982 |
|  | Linear regression | 7, 306,946,051.07 | 0.8454 |
|  | SVR | 47, 286,543,641.34 | -0.0004 |
| PCA Transformation | Random Forest | 7, 237,906,671.72 | 0.8469 |
|  | Gradient Boosting | 6, 845,414,624.62 | 0.8552 |
|  | Linear Regression | 13, 431,337,582.31 | 0.7159 |
|  | SVR | 47, 283,443,422.03 | -0.0003 |
| With Noisy Features | Random Forest | 5, 083,928,848.46 | 0.8924 |
|  | Gradient Boosting | 5, 105,377,335.81 | 0.8920 |
|  | Linear Regression | 11, 478,529,618.54 | 0.7572 |
|  | SVR | 47, 285,312,009.37 | -0.0003 |
| Hyperparameter Tuning | Random Forest (Tuned) | 4, 848,333,714.69 | 0.8974 |
|  | Gradient Boosting (Tuned) | 4, 703,333,372.42 | 0.9005 |

DISCUSSION

Across multiple experiments including scaling, feature engineering, PCA, and adding noise, the performance of models varied significantly based on preprocessing steps.

* Gradient Boosting consistently performed best, achieving the highest R² score of 0.9005 after hyperparameter tuning, with relatively low MSE values across experiments.
* Random Forest was a strong runner-up, especially with Polynomial Features, reaching an R² of 0.8988.
* Linear Regression showed moderate performance, peaking at 0.8454 with Polynomial Features but generally underperforming.
* SVR performed poorly across all experiments, with near-zero or negative R² values, making it unsuitable for this task.

RECOMMENDED MODEL:

Gradient Boosting Regressor (Tuned)

After conducting multiple experiments including feature scaling, feature engineering, dimensionality reduction, noise robustness, and hyperparameter tuning, the **Gradient Boosting Regressor** emerged as the best-performing model based on a combination of performance metrics, stability, and interpretability.

Key Justifications:

Highest Predictive Accuracy:

* After hyperparameter tuning, Gradient Boosting achieved an MSE of ~4.70 billion and an R² score of 0.9005, the highest among all models tested.
* This indicates that the model explains 90.05% of the variance in the target variable — a strong performance.

Consistent Results Across Experiments:

* Throughout the feature scaling, engineering, and noise robustness tests, Gradient Boosting consistently outperformed or matched the performance of Random Forest and significantly outperformed Linear Regression and SVR.

Resilience to Noise:

* In the presence of noisy features, the performance of Gradient Boosting remained stable (R² ~0.892), suggesting it generalizes well.

Improvement After Tuning:

* Hyperparameter tuning further improved the performance (from R² = 0.8950 to R² = 0.9005), demonstrating that Gradient Boosting is sensitive to tuning and can be optimized effectively.

Interpretability & Practical Use:

* Though not as interpretable as linear models, Gradient Boosting still allows for feature importance extraction, making it useful in real-world business applications.

REFERENCES

* Jackson, D. (n.d.). *Sample34 - Car Dataset*. Kaggle. Retrieved from [Car Price Prediction Dataset](https://www.kaggle.com/datasets/jacksondivakarr/sample34)
* Tamang, B., & Budha, D. B. (2023). *Car Price Prediction Using Machine Learning Techniques*. ResearchGate. Retrieved from [(PDF) CAR PRICE PREDICTION USING MACHINE LEARNING TECHNIQUES](https://www.researchgate.net/publication/379443793_CAR_PRICE_PREDICTION_USING_MACHINE_LEARNING_TECHNIQUES)
* Basnet, S. (2022). *Used cars price prediction and valuation using data mining techniques* (master’s thesis, Rochester Institute of Technology). RIT Scholar Works. <https://repository.rit.edu/theses/11086/>
* Stanford University. (2021). *Predicting Used Car Prices with Deep Learning* (CS230 Final Project). Retrieved from <https://cs230.stanford.edu/projects_fall_2021/reports/102475818.pdf>
* Tamang, B., & Budha, D. B. (2023). *Car Price Prediction Using Machine Learning Techniques*. ResearchGate. Retrieved from <https://www.researchgate.net/publication/379443793_CAR_PRICE_PREDICTION_USING_MACHINE_LEARNING_TECHNIQUES>