Journal Collabits
Vol. 3 No. 1 January 2026 : 56-62
E-ISSN : 3046-6709, P-ISSN : 3062-8601
https://publikasi.mercubuana.ac.id/index.php/collabits

Comparative Analysis of Linear Regression and Random Forest for Used Car Price Prediction

Muhammad Faris Adjil Syamsudi[1*], Bimo Arya Daffa[2], Wisnu Jarodi[3], Nungky Awang Chandra[4]

[1,2,3,4] Informatics Engineering, Universitas Mercu Buana, Indonesia

*Coressponden Author: pikozzyy77@gmail.com

Abstract - Manual estimation is often subjective and prone to human bias because the used car market has a complex pricing structure with non-linear depreciation. Objective: This study conducted a comparative analysis between Linear Regression and Random Forest algorithms to develop a more objective pricing model. Methods: The Kaggle dataset contains 5,000 entries indicating features such as manufacturer, model, engine size, and mileage for this study. The methodology included data cleaning, feature engineering, and outlier removal using the IQR method. For training and testing, the data was split 80:20. Results: "Year of Manufacture" was identified as the feature that most significantly influences price, and the evaluation results showed a significant difference in performance. Linear Regression achieved 82.33% accuracy, while Random Forest achieved 99.60% accuracy. Conclusion: Random Forest captures non-linear patterns and complex relationships in used car pricing better than Linear Regression, although it remains quite reliable for general trends..

Keywords :
Carsales;
Random Forest;
Linear Regression;
Machine Learning;
Price Prediction;

Article History:
Received: 06-12-2025
Revised: 19-12-2025
Accepted: 07-01-2026

Article DOI : 10.22441/collabits.v3i1.37646

INTRODUCTION

The used car market, also known as the secondary automotive industry, has a much more complex and volatile price structure than the new car market. Unlike new vehicles, which have standard retail prices, used car prices are greatly influenced by non-linear asset depreciation and are affected by many variables [4]. This depreciation involves multivariate interactions between interrelated variables, creating significant price uncertainty in the market.

In general, vehicle pricing is influenced by intrinsic characteristics (such as brand, model, and engine capacity) and usage conditions (vehicle age and mileage) [13]. Market inefficiencies often occur when sellers set prices too high (overvalued) or buyers bid too low (undervalued), which is generally caused by information asymmetry and difficulties in accurately measuring the quantitative impact of each variable on the final price [8].

Conventionally, used car price estimation is often done through a manual approach that relies heavily on individual intuition. This approach has fundamental weaknesses because it is subjective, time-consuming, and prone to human bias [11]. Manual methods often fail to capture non-linear patterns in large market data, thus necessitating a shift from subjective estimation methods to more objective and mathematically measurable computational solutions.

In the era of Big Data, the use of Machine Learning offers a powerful solution to this problem by mapping the complex relationship between vehicle features and their selling price. This study applies and compares two algorithms: Linear Regression to model simple variable relationships, and Random Forest, which was chosen for its superiority in handling non-linear data and its resistance to noise [18], [23]. This data-driven approach has been proven to minimize human bias and provide more accurate price estimates than conventional methods.

LITERATURE REVIEW

In this study, we will use a comparative analysis between two machine learning algorithms to predict used car prices. Predicting used car prices requires a method that can analyze and model the relationship between various factors that influence prices. The rising demand in the used car market has created a need for accurate, data-driven pricing tools to benefit both buyers and sellers[4]. This research shows that machine learning methods are more efficient and objective than traditional manual methods. With the increasing availability of historical transaction data, more sophisticated algorithms have made it possible to create predictive models that take into account a variety of factors.
Some of the information needed to predict used car prices includes:

- Manufacturer
- Car model
- Engine size
- Fuel type
- Year of manufacture
- Mileage

Random Forest

The Random Forest algorithm is a widely used machine learning technique for prediction problems, primarily due to its ability to handle non-linear data with multiple variables[5]. Random Forest is an ensemble learning method that builds multiple decision trees and combines their prediction results.

This algorithm works by building a number of decision trees and combining their prediction results to produce more accurate estimates[5]. In addition, Random Forest also hasthe ability to reduce the risk of overfitting, which is often a problem in other predictive algorithms.

Random Forest provides a powerful framework for handling complex and non-linear data, in this model each stage of the prediction application development is carried out sequentially and independently, either by searching for patterns in the data directly or by selecting the best features. The main goal of this prediction application is to provide an easy-to-use and efficient interface for users to determine used car price estimates based on various features.

Linier Regression

Linear regression is a quantitative method of calculation that uses time as the basis for estimated predictions[11]. This method works by using a linear equation to connect the prediction application to historical data, then querying car feature data to obtain a price prediction in numerical form that has been processed and analyzed by the model.

Linear regression, a fundamental method in statistical analysis, establishes a relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data[4]. To predict the price of a used car, the price is the dependent variable, and the independent variables include the age, mileage, engine condition, make, and maintenance history of the car. It is suitable for applications that require high interpretability because of the lightweight and easy-to-understand format of multiple linear regression equations. This model works by using linear equations to connect the prediction application to historical data and sending requests for car feature data to get price predictions in numerical form that have been processed and analyzed by the model.

METHODOLOGY

Research Method

This research was developed utilizing an organized and methodical methodology to guarantee that each step is completed in a sequential and quantifiable manner. Using this technique, the accuracy of pricing information provided by two regression methods was compared. The main objective is to make it simple for consumers to get used automobile pricing estimates.

Analysis Stage

Dataset Analysis

In this study, the data used was obtained from the Kaggle platform. This dataset includes 5,000 entries with 6 columns, which include important information about used cars such as manufacturer name, model, engine size, fuel type, year of manufacture, mileage, and price.

Functional Requirements Analysis Functional requirements are about technical steps or logical processes that must be present for an experiment to run.

- Receive input
- Cleaning Data
- Perform Training
- Generate Output

Non-Functional Requirements Analysis

Non-Functional Requirements in this case concern the quality of the model and the efficiency of the experiment.

- Accuracy (Quality)
- Computational Time (Performance)
- Reproducibility (Reliability)

Experimental Phase

Cleaning Dataset

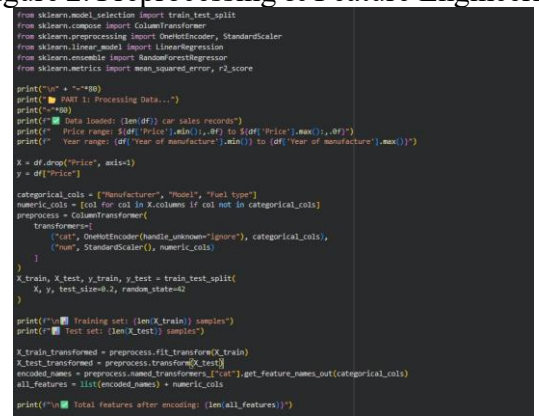
The dataset cleaning stage begins by importing the pandas, numpy, and files libraries from google.colab to upload the "car_sales_data.csv" file. After the file is successfully uploaded, the data is cleaned by removing duplicates using df.drop_duplicates(inplace=True) and removing empty values using df.dropna(inplace=True). Text normalization was applied to categorical columns for data consistency, then the dataset was filtered based on a reasonable range of values. To handle outliers, the function remove_outliers_iqr() using the IQR method which removes data outside the range [Q1 - 1.5IQR, Q3 + 1.5IQR] in the "Price" and "Mileage" columns.

Figure 1. Cleaning Dataset

```
import pandas as pd
import numpy as np
from google.colab import files

try:
    filename = "car_sales_data.csv"
    df = pd.read_csv(filename)
    print("File ditemukan dan berhasil dibuka!")
except:
    print("File tidak ditemukan. Silakan upload file CSV.")
    uploaded = files.upload()
    filename = list(uploaded.keys())[0]
    df = pd.read_csv(filename)
    print("File berhasil diupload:", filename)

df.drop_duplicates(inplace=True)
df.dropna(inplace=True)

text_cols = ["Manufacturer", "Model", "Fuel type"]
for col in text_cols:
    df[col] = df[col].astype(str).str.strip().str.title()

df = df[(df["Year of manufacture"] >= 1985) &
        (df["Year of manufacture"] <= 2024)]

df = df[(df["Price"] > 500) & (df["Price"] < 300000)]

df = df[(df["Mileage"] > 100) & (df["Mileage"] < 400000)]

df = df[(df["Engine size"] >= 0.6) & (df["Engine size"] <= 6.5)]

def remove_outliers_iqr(data, column):
    Q1 = data[column].quantile(0.25)
    Q3 = data[column].quantile(0.75)
    IQR = Q3 - Q1
    return data[(data[column] >= Q1 - 1.5*IQR) &
                (data[column] <= Q3 + 1.5*IQR)]

df = remove_outliers_iqr(df, "Price")
df = remove_outliers_iqr(df, "Mileage")

print("\nJumlah data setelah CLEANING:", len(df))
print(df.head())
print("\nFile cleaned dataset telah disimpan sebagai:", output_filename)
```

After cleaning is complete, preprocessing is performed by importing the library from scikit-learn. The X variable is separated from the target y (the "Price" column), where the categorical column is transformed using OneHotEncoder and the numeric column is normalized using StandardScaler(). The dataset is divided into 80% training data and 20% testing data with random_state=42, then the fit_transform process is applied to the training data and transform on the testing data to produce features ready for use in modeling.

Figure 2. Preprocessing & Feature Engineering



```
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

print("\n" + "="*80)
print("📂 PART 1: Processing Data...")
print("="*80)
print(f"✅ Data loaded: {len(df)} car sales records")
print(f"   Price range: ${df['Price'].min():,.0f} to ${df['Price'].max():,.0f}")
print(f"   Year range: {df['Year of manufacture'].min()} to {df['Year of manufacture'].max()}")

X = df.drop("Price", axis=1)
y = df["Price"]

categorical_cols = ["Manufacturer", "Model", "Fuel type"]
numeric_cols = [col for col in X.columns if col not in categorical_cols]
preprocess = ColumnTransformer(
    transformers=[
        ("cat", OneHotEncoder(handle_unknown="ignore"), categorical_cols),
        ("num", StandardScaler(), numeric_cols)
    ]
)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print(f"\n📊 Training set: {len(X_train)} samples")
print(f"📊 Test set: {len(X_test)} samples")

X_train_transformed = preprocess.fit_transform(X_train)
X_test_transformed = preprocess.transform(X_test)
encoded_names = preprocess.named_transformers_["cat"].get_feature_names_out(categorical_cols)
all_features = list(encoded_names) + numeric_cols

print(f"\n✅ Total features after encoding: {len(all_features)}")
```

Random Forest

Figure 3. Random Forest Implementation

```
print("\n" + "="*80)
print("🌲 PART 3: Random Forest - Predicting Car Prices")
print("="*80)

print("\n🔄 Training Random Forest...")
model_rf = RandomForestRegressor(
    n_estimators=100,
    max_depth=20,
    min_samples_split=10,
    random_state=42,
    n_jobs=-1
)
model_rf.fit(X_train_transformed, y_train)

pred_rf = model_rf.predict(X_test_transformed)

r2_rf = r2_score(y_test, pred_rf)

print(f"\n✅ Random Forest Results:")
print(f"   🎯 Accuracy: {r2_rf*100:.2f}%")

feature_importance = pd.DataFrame({
    'Feature': all_features,
    'Importance': model_rf.feature_importances_
}).sort_values('Importance', ascending=False)

print("\n🔝 Top 10 Most Important Features (Random Forest):")
print(feature_importance.head(10).to_string(index=False))
```

The Random Forest implementation uses RandomForestRegressor() with parameters n_estimators=100, max_depth=20, min_samples_split=10, random_state=42, and n_jobs=-1 to utilize all CPU cores. The model is trained using model_rf.fit(X_train_transformed, y_train) and predictions are made using model_rf.predict(X_test_transformed).

Model evaluation is performed using r2_score(y_test, pred_rf) to measure prediction accuracy. Feature importance is extracted using model_rf.feature_importances_, sorted in descending order, and displays the top 10 features that are most influential in determining the price of used cars.

Linear Regression

Figure 4. Linear regression implementation

```
print("\n" + "="*80)
print("📈 PART 2: Linear Regression - Predicting Car Prices")
print("="*80)

print("\n🔄 Training Linear Regression...")
model_lr = LinearRegression()
model_lr.fit(X_train_transformed, y_train)

pred_lr = model_lr.predict(X_test_transformed)

# Metrik evaluasi
r2_lr = r2_score(y_test, pred_lr)

print(f"\n✅ Linear Regression Results:")
print(f"   🎯 Accuracy: {r2_lr*100:.2f}%")

# Koefisien fitur
coef_df = pd.DataFrame({
    "Feature": all_features,
    "Coefficient": model_lr.coef_
}).sort_values(by="Coefficient", key=abs, ascending=False)

print("\n🔝 Top 10 Most Important Features (Linear Regression):")
print(coef_df.head(10).to_string(index=False))
```

Linear regression implementation uses LinearRegression() trained with model_lr.fit(X_train_transformed, y_train) and prediction using model_lr.predict(X_test_transformed).

Model evaluation is performed with r2_score(y_test, pred_lr) to measure prediction accuracy in percentage format. Model coefficients are extracted using model_lr.coef_ and stored in a DataFrame along with feature names, then sorted by absolute value of coefficients in descending order to identify the most influential features. The top 10 features with the highest coefficients are displayed to provide insight into the most significant variables in the linear regression model.

RESULTS AND DISCUSSION

Figure 4. Training Data Results Using Random Forest

```
Training Random Forest...

Random Forest Results:
   Accuracy: 99.60%

Top 10 Most Important Features (Random Forest):
              Feature  Importance
Year of manufacture    0.743754
        Engine size    0.126967
            Mileage    0.065468
   Fuel type_Petrol    0.016864
         Model_Polo    0.007395
Manufacturer_Toyota    0.005661
          Model_911    0.004897
         Model_Golf    0.004165
        Model_Prius    0.003373
        Model_Focus    0.003077
```

Random Forest training results show an accuracy of 99.6% with an R^2 score, much higher than Linear Regression. The "Year of manufacture" feature has the highest importance (0.743754), indicating that the year of manufacture is the most dominant factor in price prediction. The "Engine size" (0.126967) and "Mileage" (0.065468) features also provide significant contributions. Car models such as "Fuel_type_Petrol" (0.016864), "Model_Polo" (0.007395), "Manufacturer_Toyota" (0.005661), "Model_911" (0.004897), "Model_Golf" (0.004165), "Model_Prius" (0.003373), and "Model_Focus" (0.003077) have lower importance but still contribute to used car price prediction.

Figure 5. Training Data Results Using Linear Regression

```
Training Linear Regression...

Linear Regression Results:
   Accuracy: 82.33%

Top 10 Most Important Features (Linear Regression):
              Feature  Coefficient
Year of manufacture  6668.511789
          Model_911  3938.133453
            Mileage -2931.039318
       Model_Fiesta -2904.655511
         Model_Polo -2486.777948
        Model_Yaris -2429.326362
Manufacturer_Porsche  2408.399646
        Engine size  1795.564638
           Model_M5  1753.989353
          Model_Rav4  1619.257749
```

The Linear Regression training results show an accuracy of 82.33% with an R^2 score. The "Year of manufacture" feature has the highest coefficient (6668.51), indicating that the year of manufacture has the most significant positive influence on price. Car models such as "Model_911" (3938.13), "Engine size" (1795.56), and "Model_M5" (1753.98) also have a positive influence. Conversely, "Mileage" has a negative coefficient (-2931.04), indicating that high mileage lowers the price, and models such as "Model_Fiesta" (-

2904.66), "Model_Polo" (-2486.78), and "Model_Yaris" (-2429.33) show relatively lower prices.

CONCLUSION

This study successfully proved the effectiveness of applying Machine Learning in overcoming price variability problems in the used car market, where the two algorithms tested showed different performance characteristics. Based on the evaluation results, Linear Regression proved to be quite reliable in modeling general price trends with an accuracy of 82.33%, but had limitations in handling extreme data fluctuations due to its linearity assumption. In contrast, the Random Forest algorithm showed far superior performance and precision with an accuracy rate of 99.60%. This significant difference in performance indicates that the Random Forest-based method is far more adept at capturing complex non-linear relationship patterns, such as the variable rate of price depreciation due to age and usage factors, compared to classical statistical methods.

This study also confirms that the year of manufacture and mileage are the most dominant variables that determine resale value, reflecting the market's sensitivity to the remaining economic life of a vehicle. The practical implications of these findings have great potential to be developed into an automatic price estimation system that can help dealers and individual sellers set fair and transparent prices, thereby minimizing harmful speculation.

REFERENCES

[1] L. Bukvić, J. P. Škrinjar, T. Fratrović, dan B. Abramović, ”Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning,” *Sustainability*, vol. 14 Desember 2022, pp. 1-17, 2022.

[2] A. Zhu, ”Pre-Owned Car Price Prediction Using Machine Learning Techniques,” *Proceedings of the 1st International Conference on Data Analysis and Machine Learning,* vol. 1, pp. 356-360, 2024.

[3] A. J. Winata, "Prediksi Harga Mobil Bekas Menggunakan Algoritma Gradient Boosting Machine dan Random Forest," *Jurnal Inovasi Pendidikan Kreatif*, vol. 5 Desember 2024, pp. 454-463, 2024.

[4] R. Yohanes dan D. Lasut, "Web-Based used Car Price Prediction Application with Linear Regression Method," *Bit-Tech* , vol. 7 April 2025, pp. 687-695, 2025.

[5] Dewi, B. E. S., Haikal, S., Sulistyowati, H. S., Fitriani, R., & Pranowo K, D. (2024). Penerapan machine learning menggunakan algoritma random forest untuk prediksi harga mobil bekas. *Jurnal TRIDI: Teknologi Informatika & Komputer*, *2*(1), 20-31.

[6] I. Amansyah, J. Indra, E. Nurlaelasari, dan A. R. Juwita, "Prediksi Penjualan Kendaraan Menggunakan Regresi Linear: Studi Kasus pada Industri Otomotif di Indonesia," *INNOVATIVE: Journal Of Social Science Research*, vol. 4 2024, pp. 1199-1216, 2024.

[7] Y. Chen, C. Li, dan M. Xu, "Business Analytics for Used Car Price Prediction with Statistical Models," *Advances in Economics, Business and Management Research*, vol. 203 2021, pp. 542-547, 2021.

[8] E. K. Khotimah, D. I. Swasono, dan G. W. Fajarianto, "Forecasting Used Car Prices Using Machine Learning," *IT Journal Research and Development (ITJRD)*, vol. 9 Maret 2025, pp. 123-139, 2025.

[9] N. O. Idris dan F. Pontoiyo, "Evaluasi Model Machine Learning untuk Prediksi Harga Mobil dengan Perbandingan Ensemble dan Regresi Linear," *Jurnal Ilmu Komputer dan Sistem Informasi (JIRSI)*, vol. 4 Januari 2025, pp. 129-143, 2025.

[10] E. Hasibuan dan A. Karim, "Implementasi Machine Learning untuk Prediksi Harga Mobil Bekas

dengan Algoritma Regresi Linear Berbasis Web," *Jurnal Ilmiah KOMPUTASI*, vol. 21 Desember 2022, pp. 595-602, 2022.

[11] M. A. A. Syukur dan M. Faisal, "Penerapan Model Regresi Linear Untuk Estimasi Mobil Bekas Menggunakan Bahasa Python," *EULER: Jurnal Ilmiah Matematika, Sains dan Teknologi*, vol. 11 Desember 2023, pp. 182-191, 2023.

[12] M. A. Saputra, Martanto, dan U. Hayati, "Estimasi Harga Mobil Bekas Toyota Yaris Menggunakan Algoritma Regresi Linier," *Jurnal Mahasiswa Teknik Informatika*, vol. 8 April 2024, pp. 1696-1700, 2024.

[13] A. Sajad dan Nurmalitasari, "Analisis Faktor - Faktor Yang Mempengaruhi Harga Mobil Bekas Menggunakan Metode Regresi Linier," *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB) 2023*, vol. 2023 Juli 2023, pp. 266-270, 2023.

[14] E. Priande dan G. M. Zamroni, "Pengembangan Sistem Prediksi Harga Mobil Bekas OLX Menggunakan Algorithma Random Forest," *Jurnal Sarjana Teknik Informatika*, vol. 12 Februari 2024, pp. 1-8, 2024.

[15] A. Ernawati dan A. Karim, "Komparasi Kinerja Algoritma Random Forest dan C4.5 untuk Klasifikasi Harga Mobil," *Buletin Ilmiah Informatika Teknologi*, vol. 3 September 2024, pp. 25-32, 2024.

[16] N. Pal, P. Arora, D. Sundararaman, P. Kohli, dan S. S. Palakurthy, "How much is my car worth? A methodology for predicting used cars prices using Random Forest," *Future of Information and Communications Conference (FICC)*, 2018.

[17] D. A. Budiono, K. S. Utomo, K. J. Wibowo, dan M. J. Wiradinata, "Used Car Price Prediction Model: A Machine Learning Approach," *International Journal of Computer and Information System (IJCIS)*, vol. 05 Maret 2024, pp. 59-66, 2024.

[18] M. I. D. Attaqi dan J. S. Wibowo, "Prediksi Harga Mobil Bekas Berdasarkan Tipe Penjual dan Jenis Kendaraan Menggunakan Regresi Linier," *ELKOM (Jurnal Elektronika dan Komputer)*, vol. 18 Juli 2025, pp. 25-37, 2025.

[19] B. Kriswantara dan R. Sadikin, "Used Car Price Prediction with Random Forest Regressor Model," *JISICOM (Journal of Information Systems, Informatics and Computing)*, vol. 6 Juni 2022, pp. 40-49, 2022.

[20] S. Uluturk, "Regression Analysis for Predicting Prices of Used Cars: A Study Utilizing Data from Car Trading Website," *MSc Research Project, National College of Ireland*, 2024.

[21] P. A. Azhar, M. A. Pratama, dan R. Fitriani, "Prediksi Harga Mobil Audi Bekas Menggunakan Model Regresi Linear dengan Framework Streamlit," *Journal of Technology and Informatics (JoTI)*, vol. 6 Oktober 2024, pp. 22-28, 2024.

[22] J. Gao, "Second-hand car price prediction based on multiple linear regression and random forest," *Proceedings of the Quantum Machine Learning: Bridging Quantum Physics and Computational Simulations CONFMPCS 2024*, 2024, pp. 31-40, 2024.

[23] A. A. Maulana, K. Latifah, dan N. D. Saputro, "Implementasi Algoritma Random Forest Regression Untuk Estimasi Harga Mobil Bekas Merek Toyota," *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, vol. 7 Agustus 2025, pp. 1294-1303, 2025.

[24] B. E. S. Dewi, S. Haikal, H. S. Sulistyowati, R. Fitriani, dan D. P. K, "Penerapan Machine Learning Menggunakan Algoritma Random Forest Untuk Prediksi Harga Mobil Bekas," *Jurnal Tridi (Teknologi Informatika & Komputer)*, vol. 2 2024, pp. 20-31, 202