CAR PRICE PREDICTION

ALX1_AIS3_S1e IBM Data Scientists

TEAM MEMBERS

Mariam Naeim Nassim

Menna Mahmoud Abd EL-Rahman

Yomna Wael Muhammad

DEPI Final Project

Predicting Used Cars Prices Using Machine Learning

SOURCE

Predicting Used Cars Prices Using Machine Learning

Problem Statement

The used car market is highly dynamic, with prices fluctuating based on factors like brand, model, mileage, engine type, and accident history. Buyers and sellers face challenges in determining a fair market price due to the numerous influencing factors. This complexity often leads to underpricing or overpricing, resulting in inefficiency and mistrust in the market.

This project aims to address these challenges by building a machine learning model that can accurately predict used car prices. This will provide a solution for car dealerships, individual sellers, and buyers to improve market efficiency, ensure fair transactions, and foster customer satisfaction.

Objectives

1. Develop a machine learning model that predicts the price of a used car based on features such as brand, model year, mileage, engine type, and accident history.
2. Identify key factors that influence used car prices the most.
3. Improve prediction accuracy over traditional pricing models or manual pricing strategies.

Data Description

The dataset for this project was obtained from an online car marketplace and is divided into two sets:

- Training dataset: 188,533 records with 13 columns.
- Test dataset: 125,690 records with 13 columns.

Columns included:

- **id:** Unique identifier for each car
- **brand:** Car brand (e.g., Toyota, BMW)
- **model:** Specific model of the car
- **model_year**: Year the car was manufactured
- **mileage:** Distance the car has traveled
- **fuel_type:** Type of fuel used by the car (e.g., gasoline, diesel)
- **engine:** Engine size in liters
- **transmission:** Type of transmission (automatic or manual)

- **ext_col:** Exterior color of the car
- **int_col:** Interior color of the car
- **accident:** Record of accidents (Yes/No)
- **clean_title:** Whether the car has a clean title (Yes/No)
- **price**: Price of the car (only in training data)

There were several missing values in the dataset that were handled during preprocessing:

- **Fuel type:** 5,083 missing values in training, 3,383 in testing
- **Accident history:** 2,452 missing values in training, 1,632 in testing
- **Clean title:** 21,419 missing values in training, 14,239 in testing

ABSTRACT

This project aims to predict used car prices using machine learning techniques, addressing the challenge of fair pricing in the used car market due to various factors like brand, mileage, engine type, and accident history. Using a dataset from an online car marketplace, consisting of 188,533 training and 125,690 test records, the study implements models such as Linear Regression, Decision Trees, Random Forest, and Support Vector Regressor (SVR) to estimate car prices. Missing values are addressed through imputation, and categorical data is encoded for machine learning.

The Linear Regression model achieved an **RMSE of 0.526 and an R-squared score of 0.61**, indicating moderate performance. The Support Vector Regressor (SVR), after optimization, yielded an **RMSE of 71018.67, an MAE of 0.42**, and **an R-squared score of 0.53**, showing less predictive power compared to the other models. The Decision Tree model, tuned with parameters like min_samples_split=5, min_samples_leaf=10, and max_depth=10, achieved an **RMSE of 0.51, MAE of 0.36**, and an R-**squared score of 0.63,** showing solid performance in predicting car prices. However, the Random Forest model, with hyperparameters optimized as min_samples_split=5, min_samples_leaf=5, and max_depth=10, outperformed the others, achieving an **RMSE of 69098.21, an MAE of 0.35, and an R-squared score of 0.65**, demonstrating its superior effectiveness in predicting used car prices.

This project showcases a full machine learning pipeline, from data preprocessing to model evaluation, emphasizing the crucial role of feature engineering and hyperparameter tuning in improving predictive accuracy.

NOTE

The project begins with a clear problem statement: pricing used cars accurately is difficult due to numerous influencing factors. The goal is to create a machine learning model that improves price predictions compared to manual or traditional methods. The project uses a dataset of used cars, with features like mileage, engine size, transmission type, and accident history. The dataset contains missing values that are handled through imputation methods, including iterative imputation for more complex features.

Exploratory data analysis (EDA) reveals insights into the dataset, including distributions of features and missing data patterns. Visualization tools such as Seaborn are employed to identify these patterns, and missing values are filled accordingly. For example, missing fuel type values are replaced with "Electric", clean title values with "No", and missing accident data with "None reported". Feature engineering plays a significant role, with the engine column being split into several features like Horsepower and Displacement. The project also scales continuous features and encodes categorical ones using one-hot encoding.

The machine learning phase involves building and tuning several models. The Decision Tree Regressor is optimized using RandomizedSearchCV. The Random Forest Regressor is also tuned using similar techniques, achieving competitive results. Finally, the SVR model is trained, though it underperforms compared to the tree-based models.

Key Points

1. Problem Definition: The project addresses the challenge of pricing used cars accurately in a dynamic market.
2. Data Preprocessing: Missing values are filled using iterative imputation and categorical variables are handled with encoding techniques.
3. Feature Engineering: Key features like Horsepower and Displacement are extracted from the engine column, adding granularity to the dataset.
4. Visualization: Missing values and feature distributions are visualized using Seaborn plots to understand the data better.
5. Model Tuning: Hyperparameter tuning is critical in improving model performance, as seen in the Decision Tree and Random Forest models.
6. Random Forest Performance: The Random Forest model with hyperparameter tuning provides a well-balanced RMSE of 0.51.
7. SVR Model: The Support Vector Regressor, while tested, does not perform as well as the tree-based models.
8. Data Scaling: Standard scaling is used on continuous features to improve model performance.
9. Feature Encoding: One-hot encoding is applied to categorical variables like brand and fuel type to make them suitable for machine learning models.

10. Cross-Validation: The use of 5-fold cross-validation ensures that models are robust and not overfitted to the training data.

Methodology

1. Data Collection: The dataset is sourced from an online car marketplace, split into training and test sets.
2. Data Cleaning: Missing values are imputed using both simple and iterative imputation techniques.
3. Feature Engineering: New features are created from the engine column, and irrelevant columns are dropped.
4. Model Training: Models like Decision Trees and Random Forest are trained using RandomizedSearchCV for hyperparameter tuning.
5. Evaluation: Models are evaluated using RMSE, MAE, and R-squared metrics on the validation set.

Findings

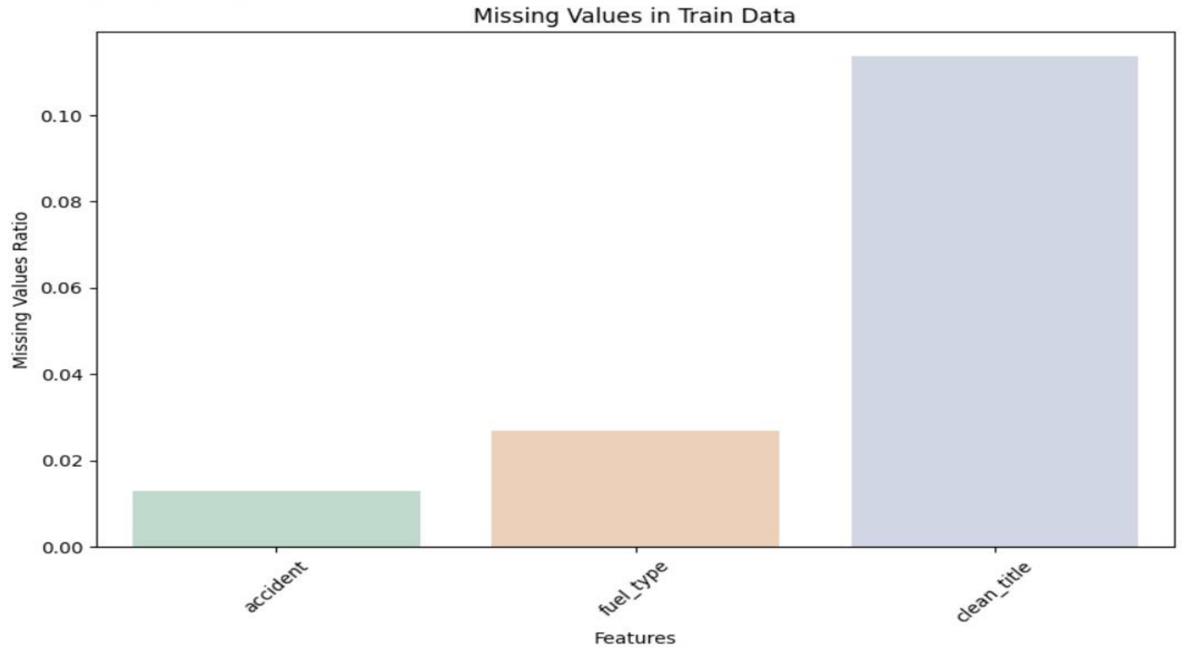
1. Importance of Feature Engineering: Features like Horsepower and Displacement significantly improve model performance.
2. Missing Value Handling: Proper imputation of missing values prevents data loss and improves model reliability.
3. Model Performance: Random Forest and Decision Tree models perform better than SVR, with Random Forest being the best.
4. Feature Impact: Brand, mileage, and engine characteristics are the most important factors affecting car prices.
5. Hyperparameter Tuning: Fine-tuning model parameters is essential for improving prediction accuracy.

Data

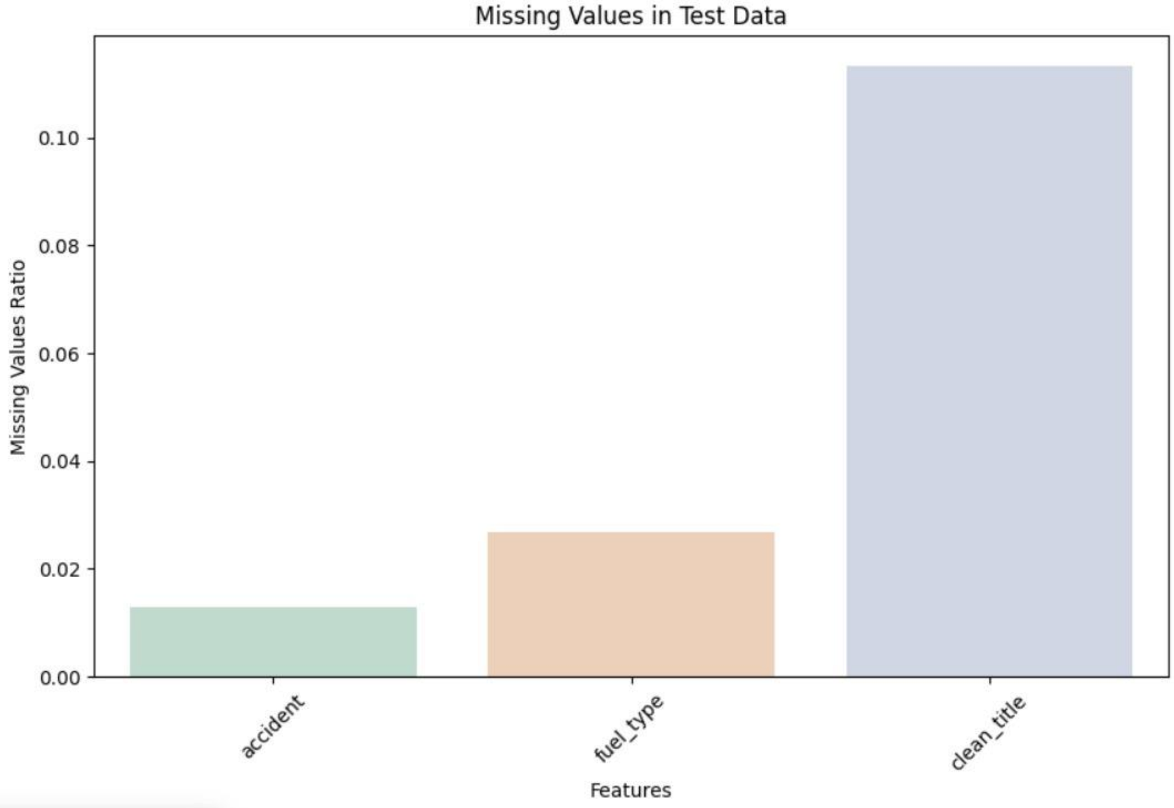
1. Train dataset: 188,533 records, 13 columns.
2. Test dataset: 125,690 records, 13 columns.
3. Columns: Include features like mileage, brand, model year, engine, transmission, and price.
4. Missing Data: Columns like fuel type and accident history have missing values.
5. Feature Engineering: Engine is divided into several sub-features like Horsepower, Displacement, and Cylinder Count.

Images

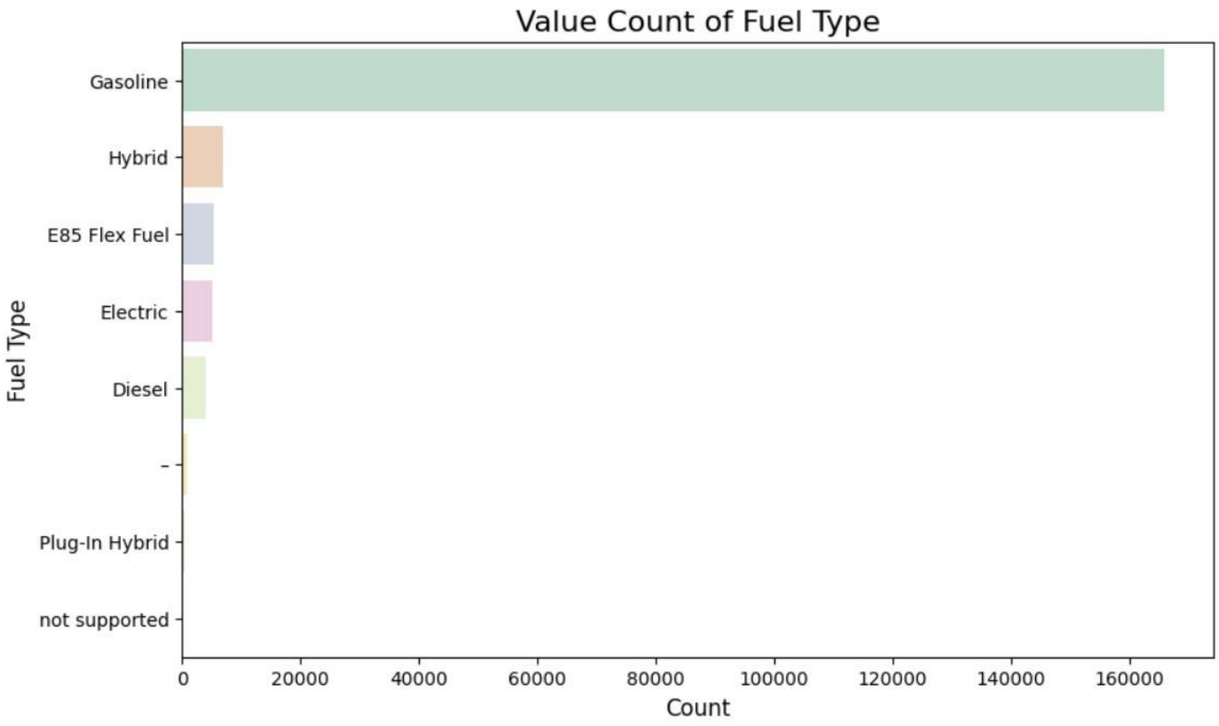
- Missing Values in Train Data



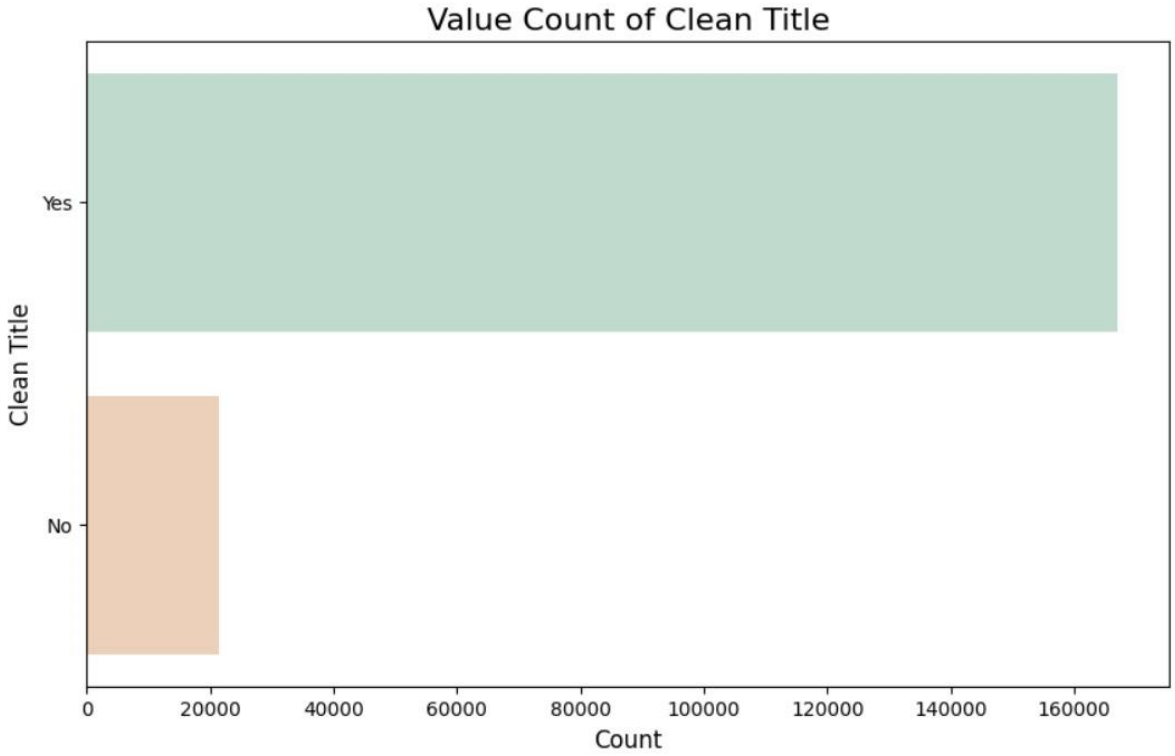
- Missing Values in Test Data



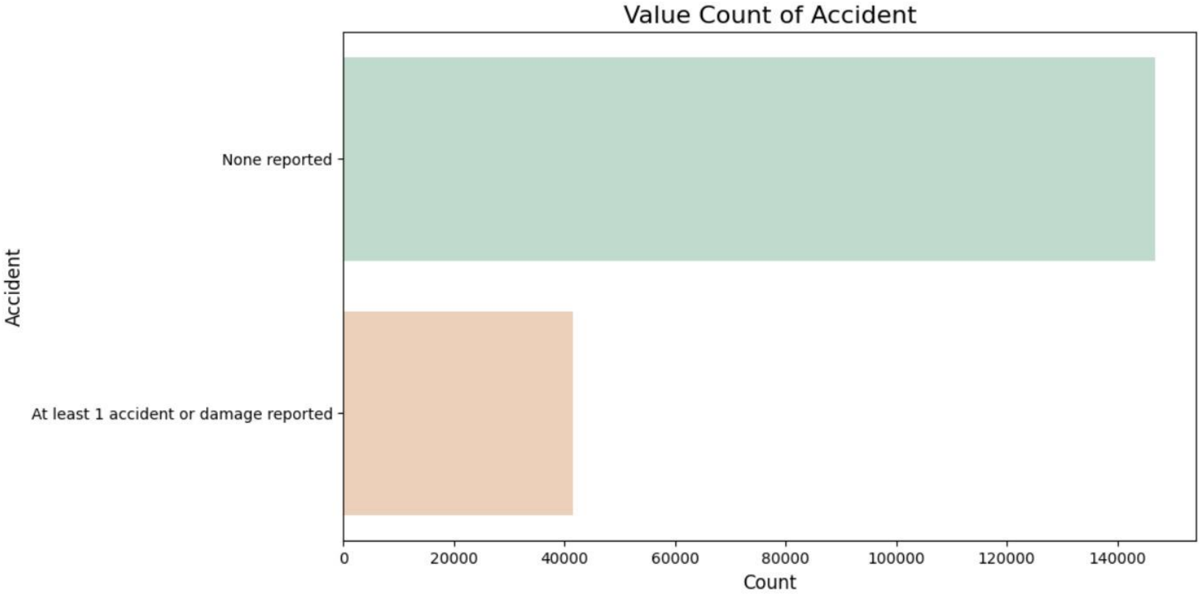
- Value Count of Fuel Type



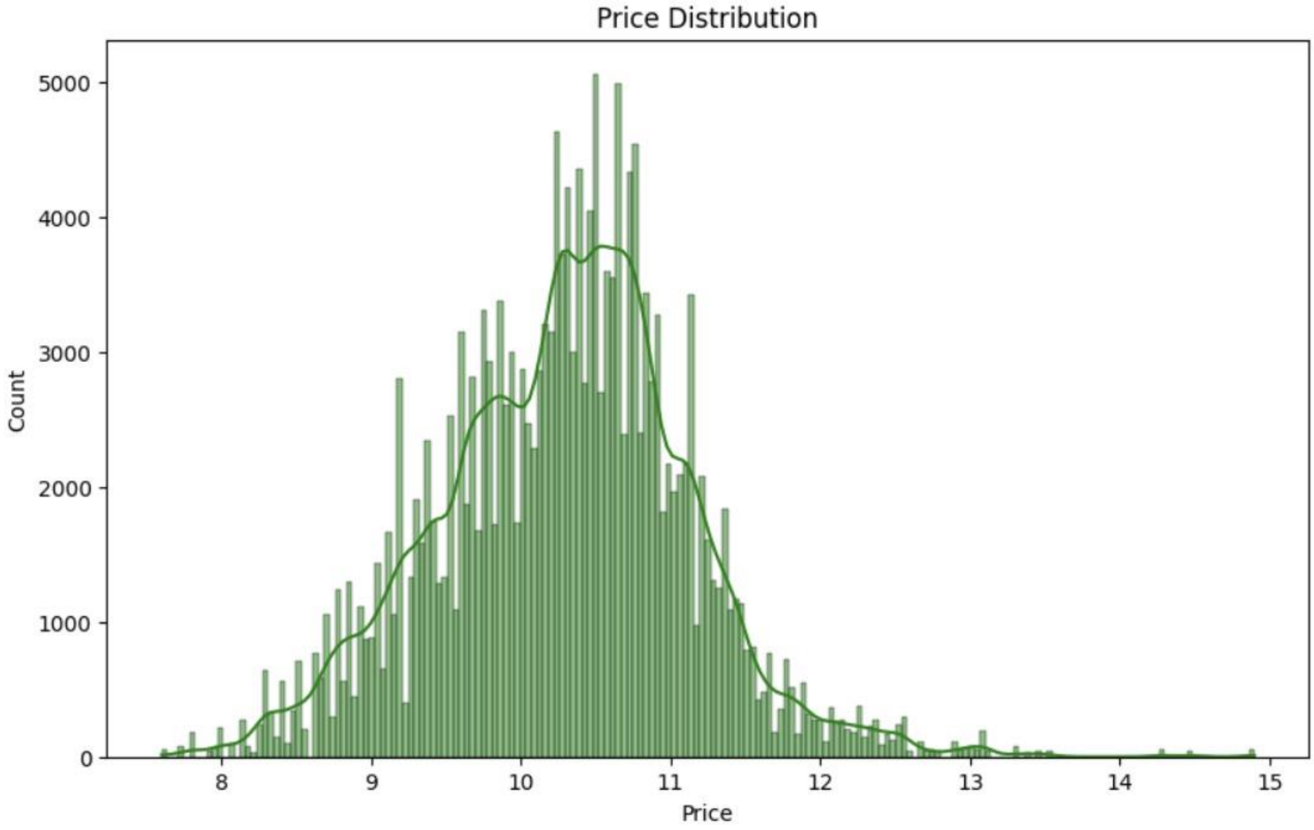
- Value Count of Clean Title



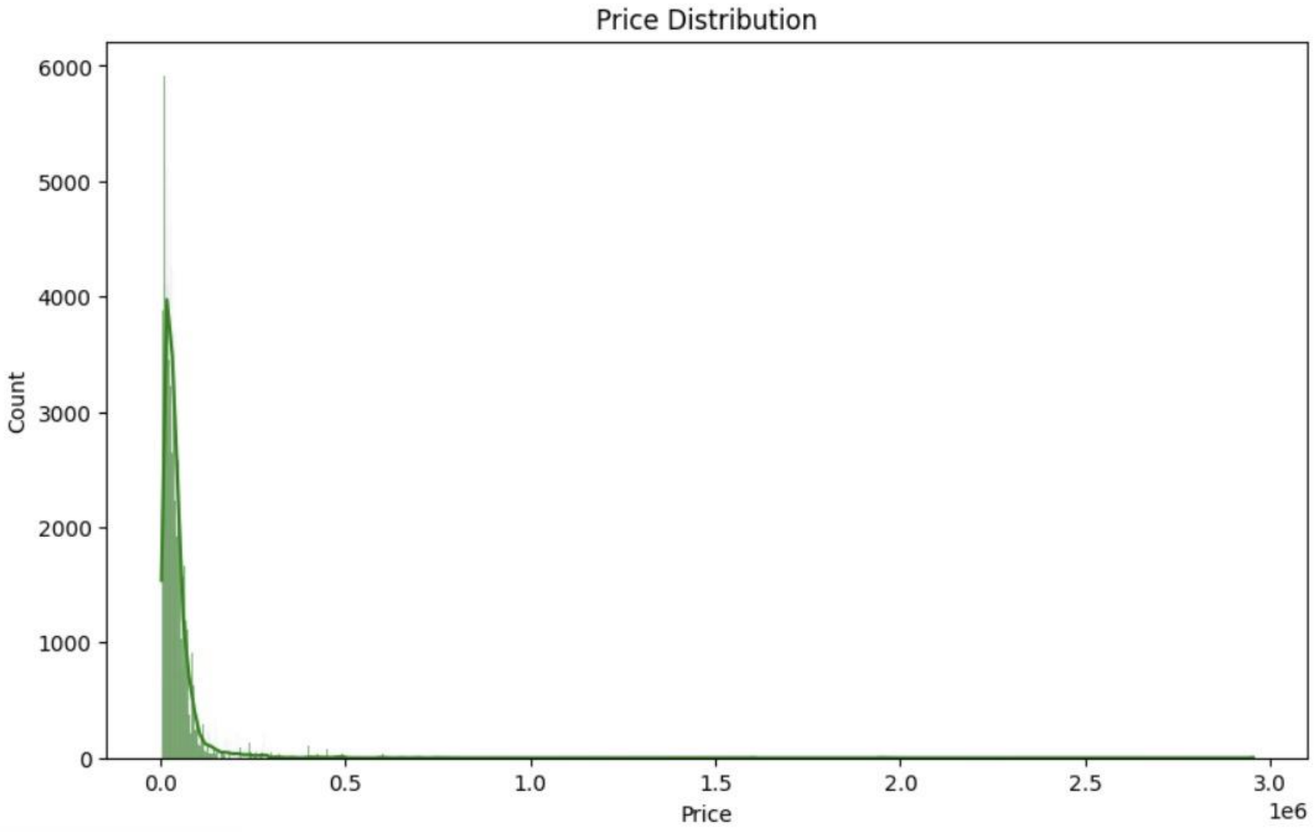
- Value Count of Accident

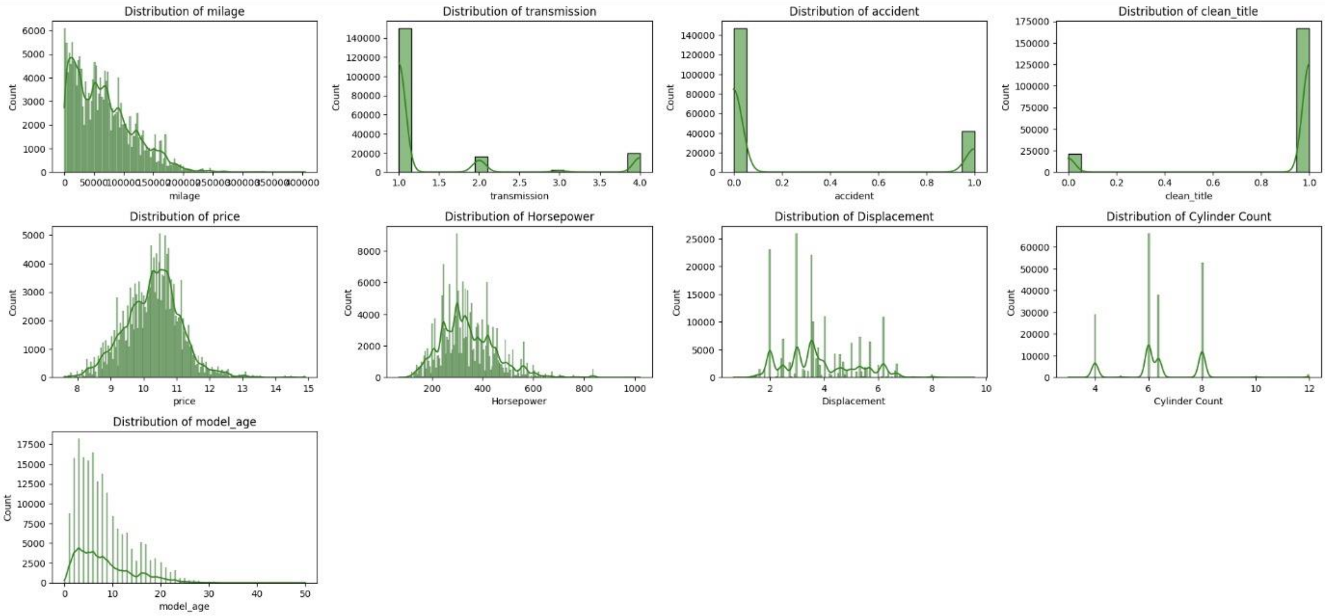


- Price Distribution



- Distributions of Various Numeric Features





Suggestions

1. Improve Feature Engineering: Extract more granular features from existing columns to improve model accuracy.
2. Additional Data Sources: Incorporate additional data such as car condition, owner history, and region to improve predictions.
3. Model Stacking: Try combining different models for better performance.
4. Fine-Tuning of Models: Further refine hyperparameters to optimize model results.
5. Automated Feature Selection: Use automated tools like recursive feature elimination (RFE) to select the most relevant features.

Principles

1. Machine Learning: Supervised learning techniques are used to predict car prices.
2. Imputation Techniques: Both simple and iterative imputation methods are used to handle missing values.
3. One-Hot Encoding: Categorical variables are converted into numerical form through one-hot encoding.
4. Scaling: Continuous features are scaled using StandardScaler for better model performance.
5. Cross-Validation: Ensures that models are robust and not overfitted to the data.

Proposal

1. Expand Data Collection: Gather more data to capture a broader range of car attributes.
2. Use Advanced Models: Experiment with more advanced models like XGBoost or LightGBM.
3. Hyperparameter Optimization: Use Bayesian optimization for more efficient tuning of models.
4. Feature Engineering Automation: Automate the process of feature extraction and engineering.
5. Model Deployment: Deploy the model into a web application for real-time car price predictions.

Conclusion

1. Random Forest performs best: It provides the most balanced accuracy for predicting car prices.
2. Feature Engineering is critical: Extracting the right features, such as Horsepower and Cylinder Count, greatly impacts performance.
3. Data cleaning is necessary: Proper handling of missing values is essential for model success.
4. Tuning is important: Hyperparameter tuning improves the precision of the model.
6. Future Directions: Further research can explore stacking models and incorporating additional data sources & working more on the accuracy of the model.

LINKS

[Source Code of the model]

REFERENCES

- Scikit-learn Documentation
- Matplotlib Documentation
- Online car marketplace dataset

DATE

2024-10-17