

Gisell Bennett

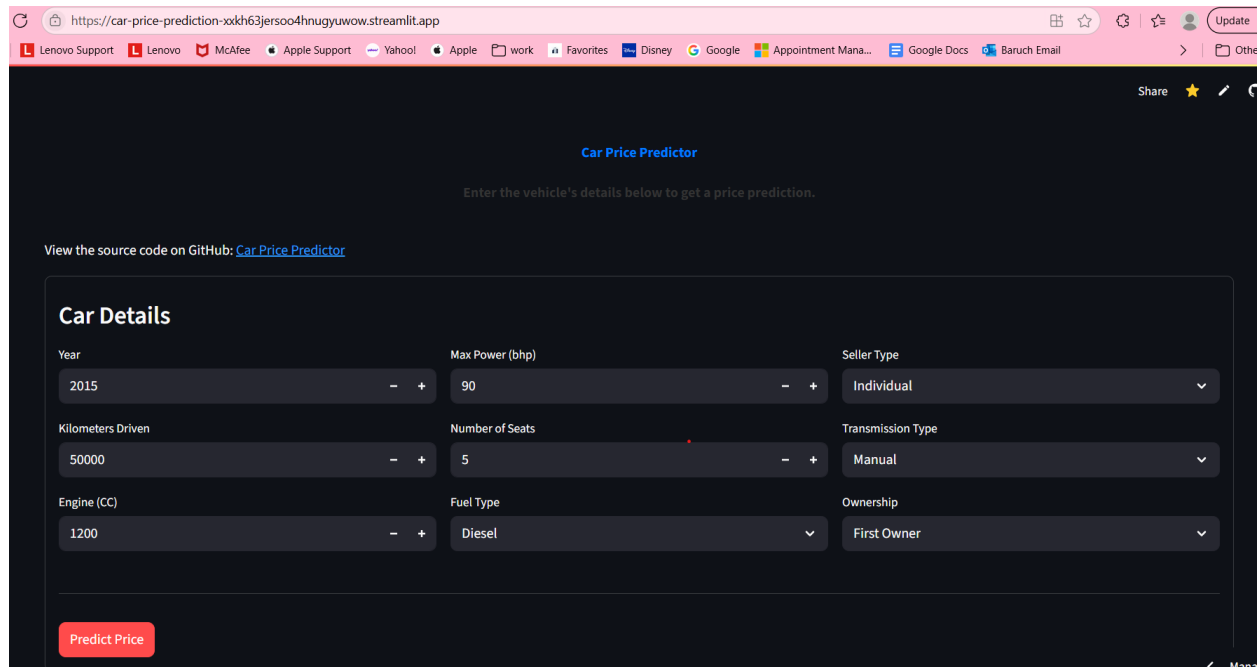
Data Mining 9660

Project 1

Technical Report: Car Price Prediction Model

Introduction

This report outlines the technical details and methodology used to develop a machine learning model for predicting the selling price of used cars. The project's goal is to create a robust and accurate predictor that can be deployed as a user-friendly web application using Streamlit.



The screenshot shows a web browser window with the URL <https://car-price-prediction-xxkh63jersoo4hnugyuwow.streamlit.app>. The browser's address bar and tabs are visible at the top. The application interface has a dark theme. At the top, it says "Car Price Predictor" and "Enter the vehicle's details below to get a price prediction." Below this, there is a link to "View the source code on GitHub: [Car Price Predictor](#)". The main section is titled "Car Details" and contains several input fields with minus and plus icons for numerical values and dropdown menus for categorical values. The inputs are: Year (2015), Max Power (bhp) (90), Seller Type (Individual), Kilometers Driven (50000), Number of Seats (5), Transmission Type (Manual), Engine (CC) (1200), Fuel Type (Diesel), and Ownership (First Owner). At the bottom left of the form is a red "Predict Price" button.

Car Details		
Year	Max Power (bhp)	Seller Type
2015	90	Individual
Kilometers Driven	Number of Seats	Transmission Type
50000	5	Manual
Engine (CC)	Fuel Type	Ownership
1200	Diesel	First Owner

Predict Price

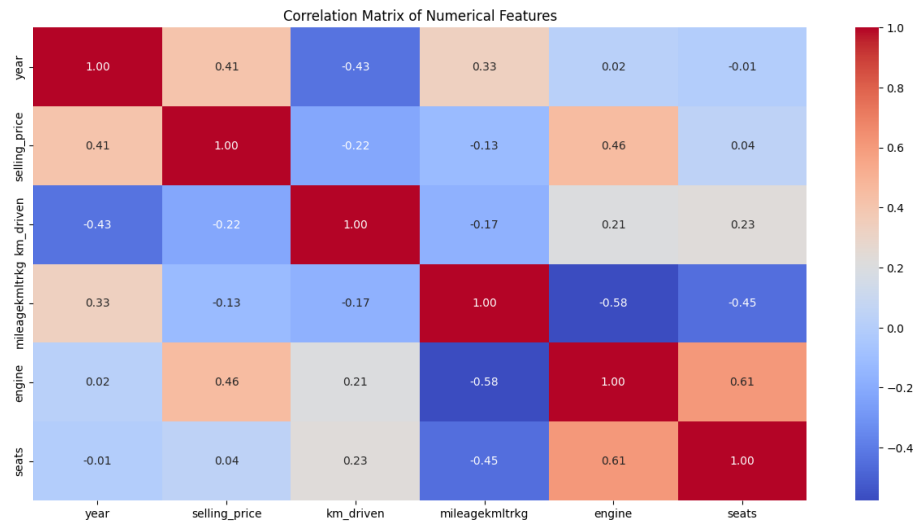
Project Goal and Data

The primary objective is to build a regression model that can accurately estimate a car's selling price based on several key features. The model leverages historical car data, which includes a mix of numerical features (e.g., `km_driven`, `engine`, `max_power`) and categorical features (e.g., `fuel`, `transmission`, `owner`).

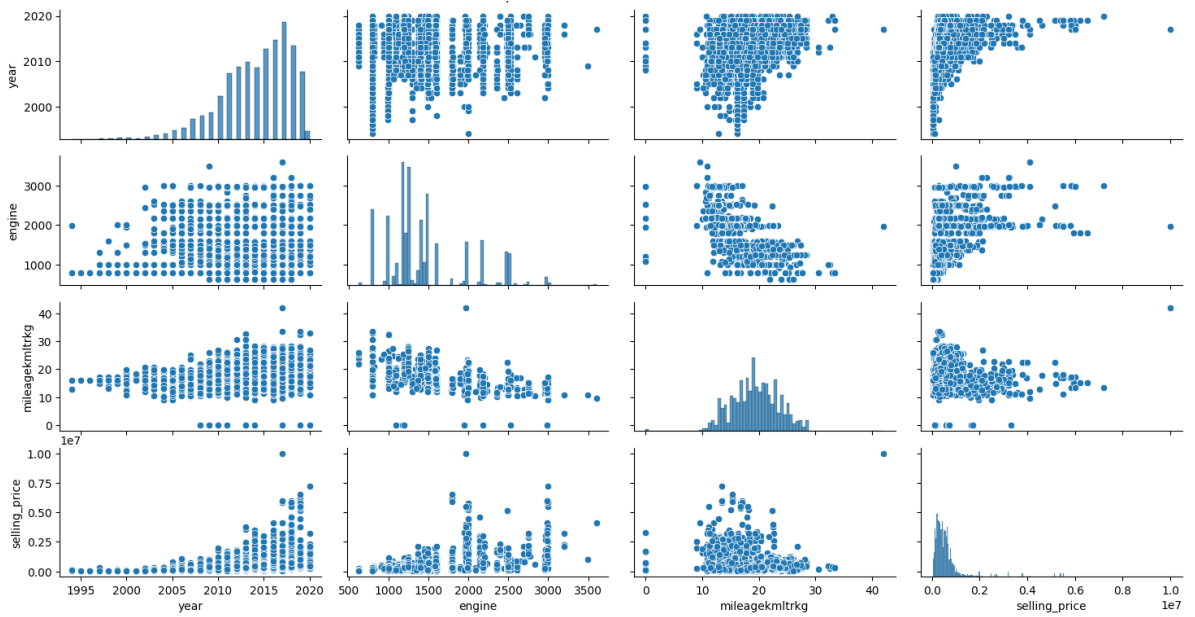
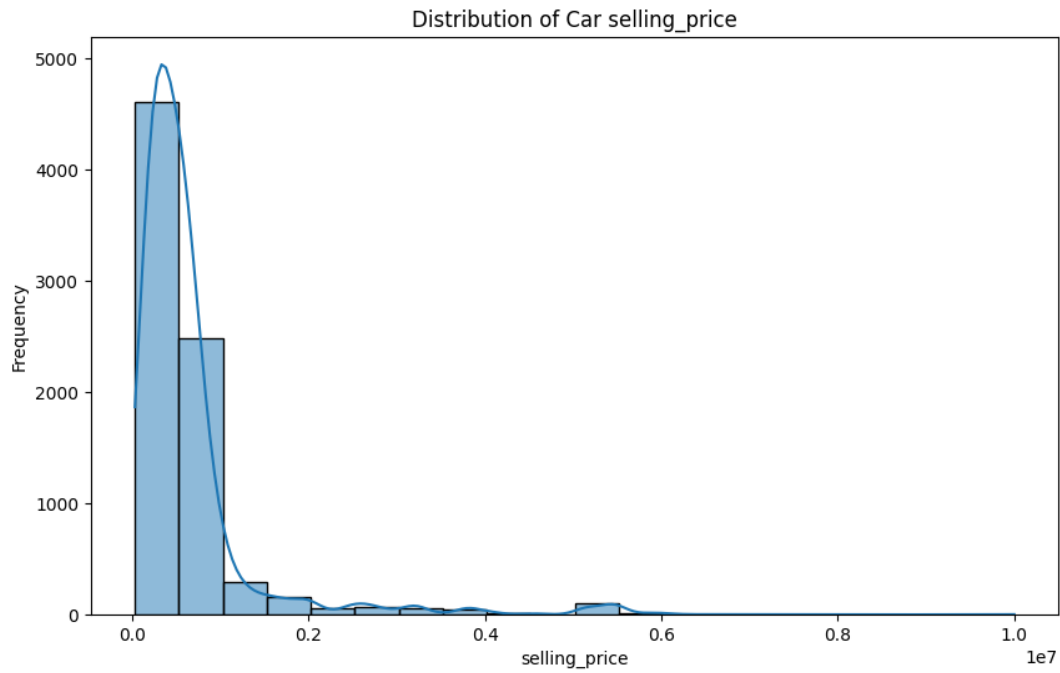
Exploratory Data Analysis (EDA)

Exploratory Data Analysis was performed to gain insights into the dataset's structure and the relationships between variables. Key visualizations helped in understanding the data distribution and identifying potential correlations.

- **Correlation Matrix:** A heatmap was generated to visualize the correlation between numerical features. This helped in understanding the linear relationships between variables such as `selling_price`, `km_driven`, and `year`.



- **Feature Distributions:** Histograms and pair plots were used to examine the distribution of individual features and their relationships with one another. The distribution of the target variable, `selling_price`, was analyzed to identify skewness and guide the choice of model evaluation metrics.



Data Preprocessing and Feature Engineering

Before training the model, the data undergoes several crucial preprocessing steps to ensure quality and compatibility:

- **Data Cleaning:** The dataset is checked for missing values (NaN). Any rows with missing data are dropped to maintain data integrity.
- **Column Standardization:** Column names are cleaned to ensure consistency, converting them to lowercase and replacing spaces with underscores.
- **Feature Engineering:** A new feature, **car_age**, is engineered by subtracting the car's year of manufacture from the current year. This provides a more direct measure of a car's age, which is often a strong predictor of price.
- **Categorical Encoding:** Categorical features such as fuel, seller_type, transmission, and owner are converted into a numerical format using **one-hot encoding**. This process creates new binary columns for each category, which machine learning algorithms can process effectively.
- **Feature Scaling:** Numerical features are standardized using **StandardScaler**. This transforms the data to have a mean of 0 and a standard deviation of 1, which helps prevent features with larger scales from dominating the model's learning process.

Model Selection and Training

Two different regression models were selected to build the predictive engine:

1. **Linear Regression:** A foundational model that establishes a linear relationship between features and the target variable. It provides a simple, interpretable baseline for performance.
2. **XGBoost Regressor:** A powerful gradient-boosting algorithm known for its high performance and ability to handle complex, non-linear relationships in the data.

Both models were trained on 80% of the preprocessed data, with the remaining 20% reserved for evaluation.

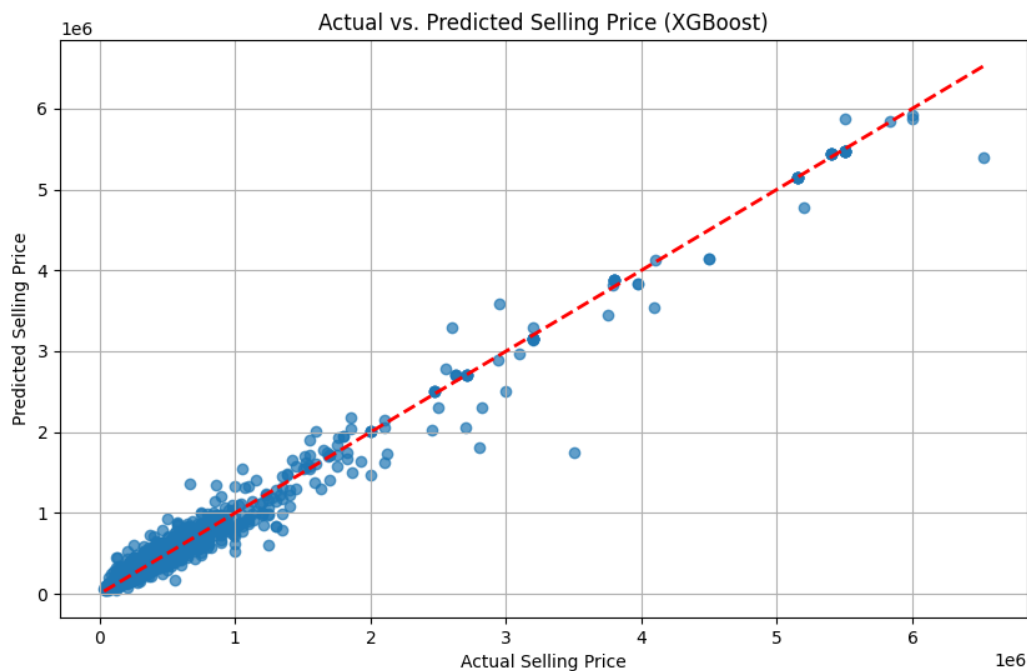
Model Evaluation

The performance of the trained models was evaluated using several standard regression metrics. The results on the test set are as follows:

- **Linear Regression Model:**
 - **Mean Absolute Error (MAE):** 280826.76
 - **Mean Squared Error (MSE):** 230231611897.85
 - **Root Mean Squared Error (RMSE):** 479824.56
 - **R-squared (R2) Score:** 0.69
- **XGBoost Regressor Model:**

- **Mean Absolute Error (MAE):** 74476.49
- **Mean Squared Error (MSE):** 15620326400.00
- **Root Mean Squared Error (RMSE):** 124981.30
- **R-squared (R2) Score:** 0.98
- **Cross-Validation:** The XGBoost model showed excellent performance with an average R-squared score of **0.97** across 5 folds.

The results clearly indicate that the XGBoost Regressor provides a significantly more accurate and robust prediction. A visual comparison of the model's predictions against the actual values further confirms this.



Deployment and Insights

The final, trained models and the StandardScaler are saved to disk using the joblib library. This allows them to be loaded directly into a Streamlit application for real-time price prediction without needing to be retrained.

The Streamlit app:

- Collects user input for various car features.

- Preprocesses the user input in the same way as the training data.
- Uses the loaded XGBoost model to generate a price prediction.
- Displays the predicted price to the user in a clear and intuitive format.

The app also provides insights into the **feature importance** of the XGBoost model, visualizing which features had the most influence on the final price prediction. This provides transparency and helps users understand the factors driving the model's output.

Conclusion

This project successfully demonstrates the end-to-end process of building, evaluating, and deploying a machine learning model for car price prediction. The XGBoost Regressor provides a high-performing model, and the Streamlit application makes the prediction functionality easily accessible to users.