Artificial Intelligence
&
Machine Learning

Project Report

Semester-IV (Batch-2022)

CAR PRICE PREDICTION

Supervised By:

Dr. Kirandeep Singh

Submitted By:

Arshaan Singh, 2210990168

Dapinder Singh, 2210990248

Arshdeep Singh, 2210990171

Harkirat Singh, 2210990993

Department of Computer Science and Engineering Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Introduction

Utilizing Machine Learning for Predicting Car Prices.

Background

The concept of car price prediction is rooted in the automotive market dynamics. When evaluating the price of a car, several factors influence its market value. These factors include the car's specifications, market demand, brand reputation, age, condition, and economic conditions. This price evaluation is influenced by variables such as vehicle type, features, mileage, year of manufacture, brand perception, market trends, and geographic location.

Objectives:

Develop a Python program using machine learning to predict car prices. Utilize data analysis techniques to train the model, enabling accurate estimations of car prices based on features like make, model, year, mileage, and fuel type, simplifying the buying and selling process for users.

Significance

- **Simplified Buying Process**: Car price prediction through machine learning simplifies the buying process for both sellers and buyers. Sellers can accurately gauge the market value of their vehicles, while buyers can make more informed decisions based on predicted prices, reducing the time and effort involved in negotiations.

- **Personalized Recommendations**: Machine learning models can analyze individual preferences, such as brand affinity, desired features, and budget constraints, to provide personalized recommendations. This assists buyers in finding vehicles that align with their specific needs and preferences, enhancing their overall satisfaction with their purchase.

- **Integration with Online Marketplaces**: Online car marketplaces can leverage car price prediction models to provide real-time pricing estimates for listed vehicles. This enables sellers to set competitive prices, increasing the likelihood of a successful sale, while buyers can quickly assess whether a listing offers good value for money.

- **Transparent Pricing**: By utilizing machine learning to predict car prices, the pricing process becomes more transparent for both parties involved. Sellers can justify their asking prices based on data-driven insights, and buyers can trust that the prices they're paying are fair and reflective of the vehicle's characteristics and market demand.

Problem Definition and Requirements

Problem Statement:

The challenge at hand pertains to the necessity for precise estimation of car prices to facilitate efficient buying and selling processes within the automotive market. Conventional methods frequently fall short in providing accurate assessments, resulting in hurdles such as prolonged negotiations and uncertainty regarding fair market value. Consequently, there exists a demand for sophisticated predictive models capable of harnessing comprehensive data sources to furnish users with timely and reliable insights into the valuation of vehicles.

Software Requirements:

Python is the primary programming language recommended for developing predictive models for car price prediction. Here's a list of Python libraries and tools commonly used in this context:

Python Libraries:

1. **NumPy** : For numerical computations and data manipulation.
2. **Pandas** : For data preprocessing and analysis.
3. **Scikit-learn** : For implementing machine learning algorithms for predictive modeling.
4. **TensorFlow or PyTorch** : For building and training neural network models, if applicable.
5. **Jupyter Notebook** : interactive development and experimentation with Python code.

Data Visualization Libraries:

1. **Matplotlib** : For creating static plots and visualizations.
2. **Seaborn** : For creating more sophisticated and visually appealing statistical plots.
3. **Plotly** : r interactive and web-based visualizations.

Development Environment: An integrated development environment (IDE) such as PyCharm, Visual Studio ded for coding, debugging, and testing the predictive models.

Additionally, depending on the specific requirements and data sources, you may need access to:

Wearable Device APIs or SDKs: If integrating with wearable devices for collecting data, you may require APIs or SDKs provided by the manufacturers to access sensor data and integrate it into your predictive models.

With these tools and libraries, you can develop robust predictive models for car price prediction, enabling more informed decision-making in the automotive market.

Database Management System (DBMS): While optional, a DBMS can be advantageous for storing and managing large volumes of data collected from various sources, including market trends, historical sales data, and vehicle specifications. Popular choices include PostgreSQL, MySQL, or MongoDB, depending on the specific needs and scalability requirements of the project.

Web Framework (Optional): If the predictive models are intended to be deployed as web services or applications, integrating a web framework such as Flask or Django can facilitate the development of the backend infrastructure. This allows for the creation of user-friendly interfaces for accessing predictive features, as well as seamless integration with other web-based systems.

Version Control System: Using a version control system like Git is essential for managing code changes, collaborating with team members, and maintaining a history of modifications throughout the development process. Platforms like GitHub or GitLab offer additional features such as issue tracking, code reviews, and continuous integration, which can streamline the development workflow.

Documentation and Reporting Tools: Documentation tools such as LaTeX, Microsoft Word, or Google Docs can be utilized to document the development process, record results, and communicate findings effectively. Creating detailed documentation ensures transparency, reproducibility, and easy sharing of insights with stakeholders, team members, and future developers.

Hardware Requirements:

Computer: A desktop or laptop computer with sufficient processing power and memory is essential for running the software tools and performing data analysis tasks effectively. A modern multi-core processor such as Intel Core i5 or AMD Ryzen 5, paired with at least 8 GB of RAM, is recommended for optimal performance during model training, data preprocessing, and analysis.

Internet Connection: A stable internet connection is necessary for downloading software tools, accessing online resources and APIs, and collaborating with team members if working in a distributed environment. It also facilitates data collection from online sources such as car listings, market trends, and historical sales data. **Data Sources:** Access to various data sources such as online car marketplaces, historical sales databases, and automotive industry reports may require internet connectivity for data collection and retrieval. Integration with APIs or web scraping techniques may be employed to gather relevant data for training predictive models.

Wearable Fitness Trackers (Optional): While not directly related to car price prediction, wearable fitness trackers may be used for personal health and fitness monitoring by project team members. These devices can provide insights into physical activity levels, heart rate, and sleep patterns, contributing to overall well-being and productivity.

Mobile Devices (Optional): Mobile devices may be utilized for testing mobile applications or web services developed as part of the car price prediction project. They can also serve as additional data collection tools for capturing images, videos, or location data relevant to the automotive market.

External Storage (Optional): External hard drives or cloud storage services can be utilized for storing large datasets, model checkpoints, and other project-related files. This helps free up space on the computer's internal storage and ensures that important data is backed up securely.

Development Boards (Optional): In certain scenarios where the project involves hardware prototyping or integration with embedded systems, development boards like Raspberry Pi, Arduino, or microcontrollers may be necessary for testing and validation purposes. These boards allow for experimentation with physical interfaces and real-world interactions related to automotive systems.

Power Source: It's crucial to ensure access to a reliable power source for the computer, wearable devices, and any other hardware components used in the project. This helps prevent interruptions during data collection, analysis, and model training, ensuring smooth progress throughout the development process.

Datasets:

Publicly Available Datasets:

1. **Used Car Listings:** Websites like Kaggle, eBay Motors, or government databases may offer datasets

containing information on used car listings, including make, model, year, mileage, price, and additional features. These datasets can serve as valuable sources for training predictive models for car price estimation.

2. **Automotive Market Reports:** Organizations such as Kelley Blue Book, Edmunds, or automotive industry associations often publish reports and datasets containing market trends, average prices, and sales data for different types of vehicles. These datasets provide insights into pricing dynamics and market demand, aiding in the development of accurate predictive models.

3. **UCI Machine Learning Repository:** While primarily focused on health and fitness datasets, the UCI Machine Learning Repository also offers datasets related to automotive data, such as car evaluation, vehicle specifications, and fuel efficiency. These datasets can supplement existing data sources and provide additional features for model training.

4. **Custom Collected Datasets:** Researchers, automotive industry stakeholders, and organizations may collect their own datasets using proprietary methods, such as scraping online car listings, conducting surveys, or collaborating with dealerships. These custom datasets can be tailored to specific research questions and demographics, providing more relevant and accurate data for car price prediction.

5. **Wearable Device Data (Optional):** While not directly related to car price prediction, some wearable fitness trackers and health monitoring devices offer APIs or SDKs that allow developers to access raw sensor data. While not applicable to car price prediction, these datasets can serve as examples of how sensor data can be leveraged for predictive modeling in different domains.

Features:

1. **Vehicle Age**: Similar to duration in physical activities, the age of the vehicle can be a fundamental feature for estimating its price. Generally, older vehicles tend to have lower prices compared to newer ones.

2. **Vehicle Type**: The type of vehicle (e.g., sedan, SUV, truck) corresponds to the activity type in physical activities. Just as different activities have varying energy requirements, different vehicle types have different market demands and pricing structures.

3. **Vehicle Mileage**: Similar to duration in physical activities, the mileage of the vehicle is a critical factor

for estimating its price. Higher mileage typically correlates with lower prices, as it indicates more wear and tear and potentially higher maintenance costs.

4. **Vehicle Condition**: The condition of the vehicle, measured in terms of its overall wear and tear, corresponds to the intensity of the activity in physical activities. Higher intensity activities result in greater wear and tear on the body, similarly, a vehicle in poor condition may have lower prices compared to one in excellent condition.

5. **Vehicle Make and Model**: Similar to how age and gender can influence calorie expenditure rates, the make and model of the vehicle can impact its price. Certain makes and models may have higher resale values or be in greater demand, leading to higher prices.

6. **Market Demand**: Just as environmental factors can impact calorie expenditure during physical activities, market demand influences car prices. Factors such as trends, seasonality, and economic conditions affect the demand for specific vehicle types and brands, consequently impacting their prices.

7. **Vehicle Features**: Features such as engine size, fuel efficiency, technology, and safety features correspond to individual characteristics like resting metabolic rate (RMR) and fitness level. Vehicles with more features or higher performance capabilities may command higher prices in the market.

Proposed Design / Methodology

1. **Data Collection:** Gather a dataset containing labelled examples of vehicle listings along with corresponding price measurements. Utilize publicly available datasets from platforms like Kaggle, eBay Motors, or government databases. Alternatively, collect data from online car marketplaces or dealership websites.

2. **Data Preprocessing:**
   - Clean the dataset by handling missing values, removing outliers, and ensuring data consistency.
   - Perform feature engineering to extract relevant features from the raw data. This may include variables such as vehicle make, model, year, mileage, condition, and features.
   - Normalize or scale the features to ensure they are on a similar scale, which can improve the performance of machine learning algorithms.

3. **Split Data:** Split the dataset into training and testing sets. Reserve a larger portion of the data (e.g.,

80%) for training and the remaining portion for testing (e.g., 20%).

4. **Model Selection:**
    - Choose appropriate machine learning algorithms for regression tasks. Common choices include linear regression, support vector regression, decision trees, random forests, and gradient boosting regressors.
    - Experiment with different algorithms to determine which one performs best for your specific dataset and prediction task.
5. **Model Training:**
    - Train the selected machine learning models using the training dataset. Fit the models to the training data and adjust their parameters as needed.
    - Use cross-validation techniques (e.g., k-fold cross-validation) to assess the models' performance and ensure they generalize well to unseen data.
6. **Model Evaluation:**
    - Evaluate the trained models using the testing dataset. Calculate metrics such as mean absolute error (MAE), root mean squared error (RMSE), and R-squared to measure the models' accuracy and predictive performance.
    - Compare the performance of different models and select the one that achieves the best results.

RESULTS:

Importing the dependencies/Libraries

```
In [116]: input_data_model = pd.DataFrame(
              [[5,2022,12000,1,1,1,1,12.99,2494.0,100.6,5.0]],
              columns=['name','year','km_driven','fuel','seller_type','transmission','owner','mileage','engine','max_power','seats'])
```

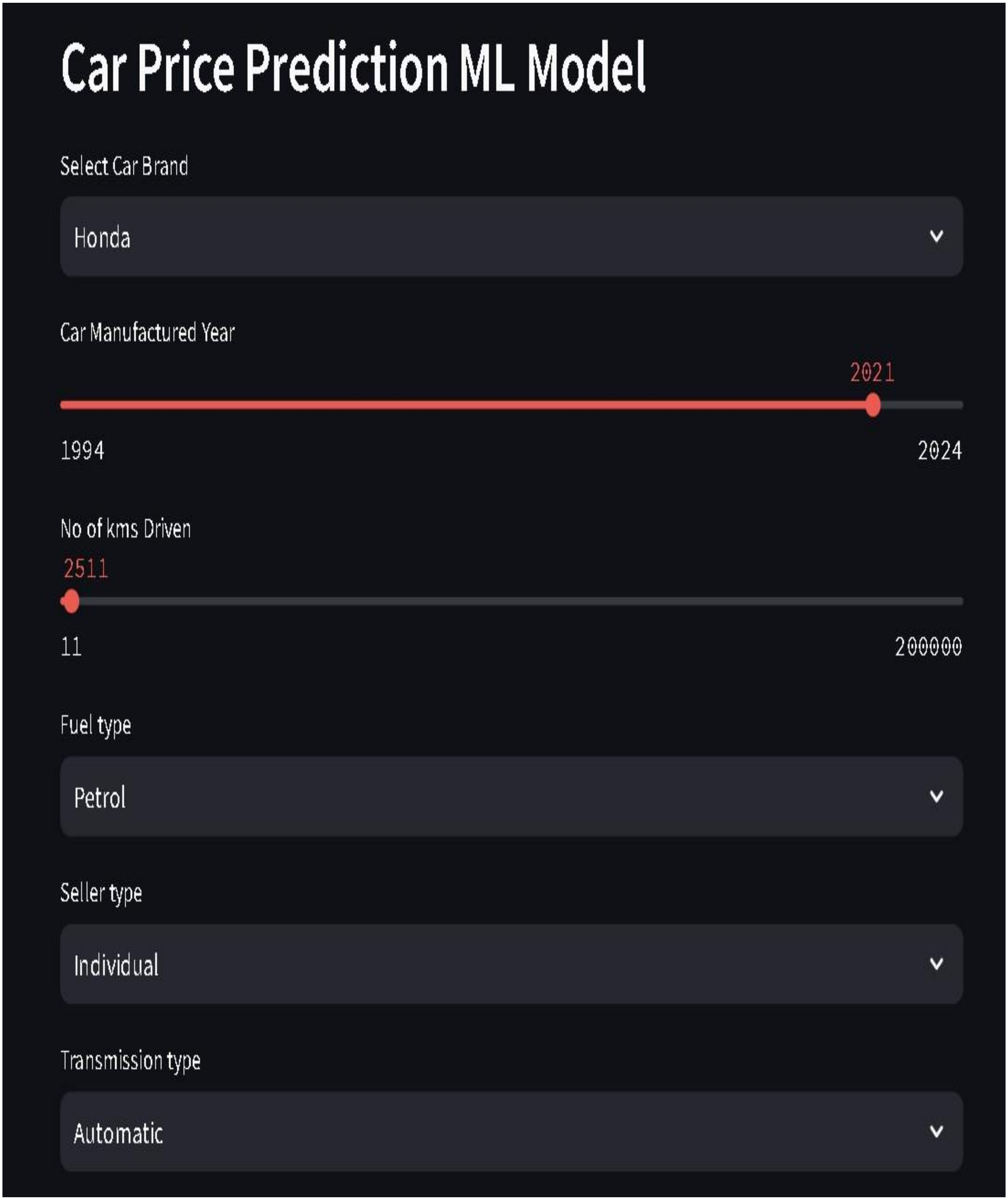
```
In [117]: input_data_model
```

Out[117]:

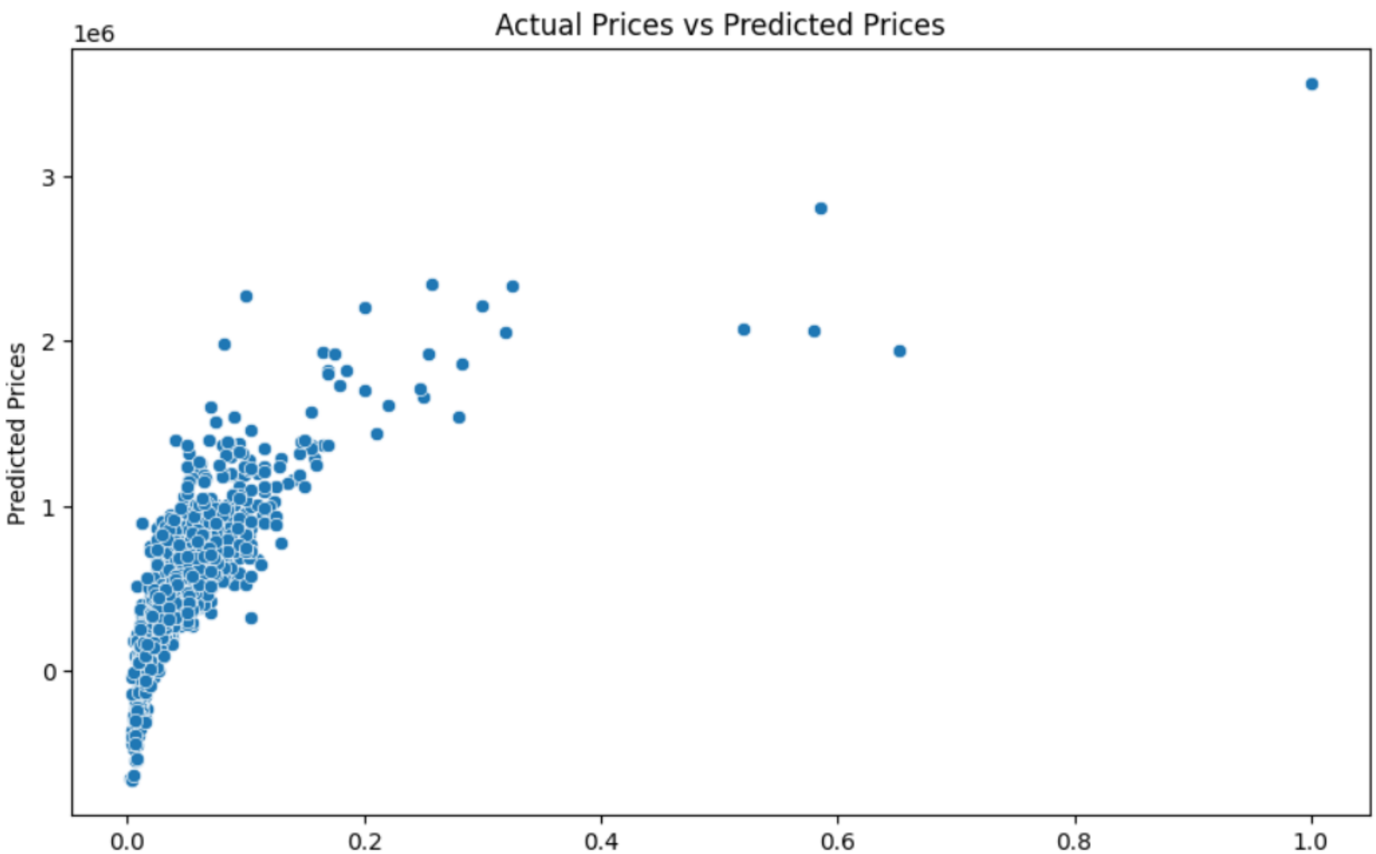
	name	year	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_power	seats
0	5	2022	12000	1	1	1	1	12.99	2494.0	100.6	5.0

```
In [118]: model.predict(input_data_model)
```

```
Out[118]: array([1050658.42244005])
```

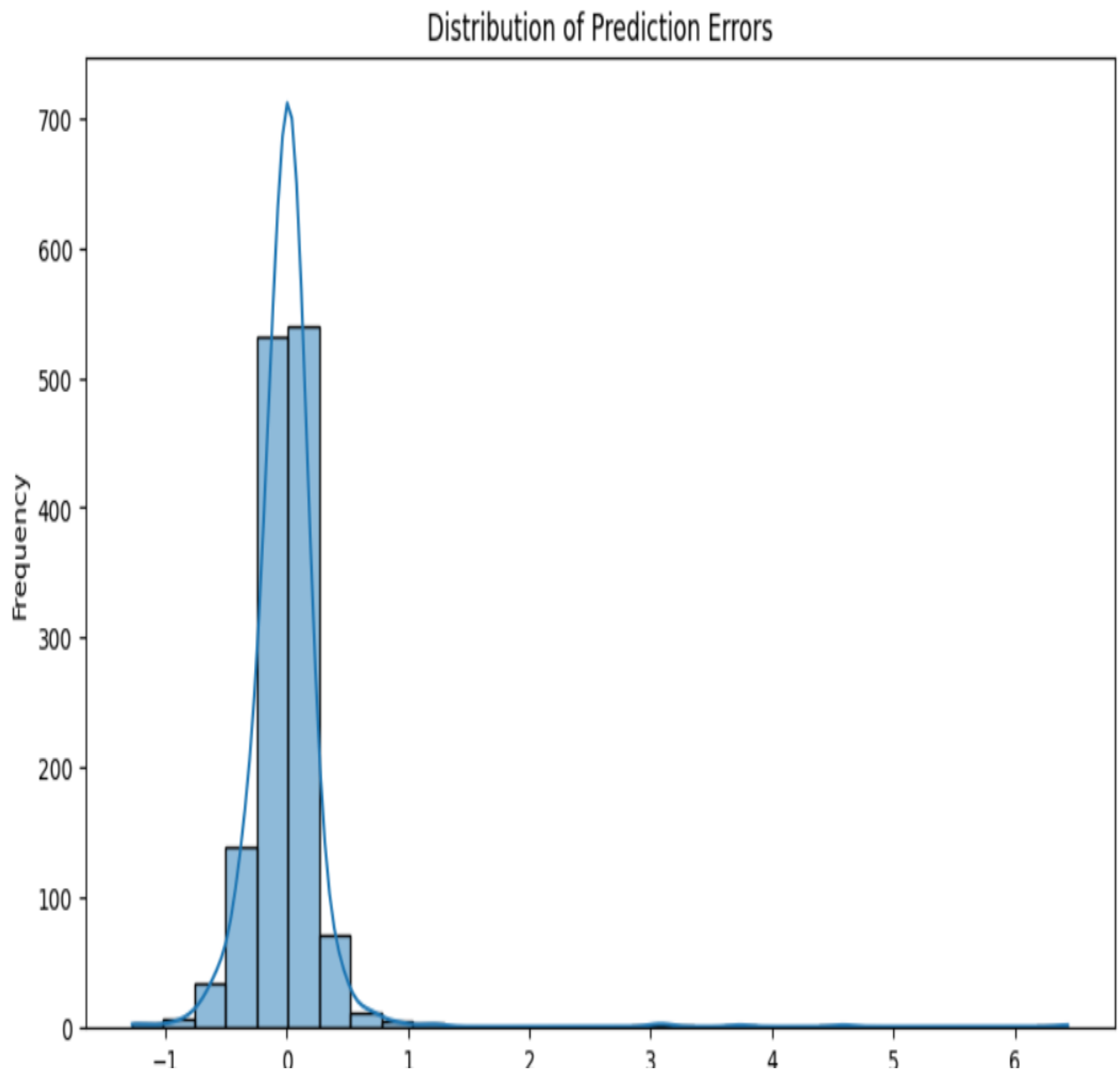

Car Price Prediction ML Model
Select Car Brand
Honda
Car Manufactured Year
2021
1994
2024
No of kms Driven
2511
11
200000
Fuel type
Petrol
Seller type
Individual
Transmission type
Automatic

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=predict)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual Prices vs Predicted Prices')
plt.show()
```



In [111…

```python
plt.figure(figsize=(10, 6))
sns.histplot(y_test - predict, bins=30, kde=True)
plt.xlabel('Error')
plt.ylabel('Frequency')
plt.title('Distribution of Prediction Errors')
plt.show()
```



References:

- https://thecleverprogrammer.com/2021/08/04/car-price-predictionwith-machine-learning/
-

- https://medium.com/analytics-vidhya/car-price-prediction-end-to-endmachine-learning-web-application-8e9e1fcbd8b3