# Car Price Prediction Using Machine Learning

## Introduction

For this assignment, I aimed to build a machine learning model to predict car prices based on various features of the cars. The initial dataset contained approximately 4.2 million rows, which presented significant challenges due to the limited computational resources available.

Here’s a step-by-step overview of the process, challenges faced, and solutions implemented.

## Dataset Preparation

### Initial Dataset:

- The original dataset had around 4.2 million rows.  
- Due to the high volume of data, my system faced performance issues, making it infeasible to process the entire dataset.

### Downsampling:

- To manage the computational limitations, I first reduced the dataset to 60,000 rows. However, the system continued to crash.  
- On my instructor’s advice, I further reduced the dataset to 2,000 rows. This allowed me to proceed with model training without overwhelming the system.

## Model Development

### Data Preprocessing:

- Handled missing values by filling them with appropriate defaults.  
- Encoded categorical variables using one-hot encoding.  
- Scaled numerical features using the StandardScaler.

### Feature Engineering:

- Extracted the car’s age from the year of manufacture and added it as a new feature.  
- Dropped less relevant columns to streamline the dataset.

### Model Selection:

- I trained four different regression models:  
- Linear Regression  
- Decision Tree Regressor  
- Random Forest Regressor  
- Gradient Boosting Regressor

## Performance Tuning

### Initial Evaluation:

- After training the models, I evaluated their performance on the test set.  
- The initial performance of the models was not satisfactory.

### Hyperparameter Tuning:

- To improve the performance, I conducted hyperparameter tuning on the Random Forest Regressor, which initially showed the most promise.  
- I used a reduced parameter grid to balance performance improvement and computational efficiency.

### Results:

- After tuning, the Random Forest Regressor showed significant improvement:  
 - Mean Absolute Error (MAE): 4365.67  
 - Mean Squared Error (MSE): 67405045.30  
 - Root Mean Squared Error (RMSE): 8210.06  
 - R-squared Score: 0.71

## Deployment

### Flask Application:

- To make the model accessible for predictions, I developed a Flask web application.  
- The application allows users to input relevant car details and receive a predicted price.  
- Only essential features are required for input, ensuring the app is user-friendly and efficient.

## Challenges and Learning

- The primary challenge was managing the large dataset with limited computational resources.  
- Downsampling was a crucial step to enable model training.  
- Hyperparameter tuning helped enhance the model’s performance significantly.  
- Developing the Flask app provided a practical deployment solution, making the model accessible for real-world use.

## Conclusion

This assignment provided valuable insights into data handling, model training, performance tuning, and deployment. Despite the initial challenges, I successfully developed a robust car price prediction model and a corresponding web application.

The process enhanced my understanding of machine learning workflows and the importance of resource management in large-scale data projects.