**1. Data Preparation**

* **Input Dataset**: The dataset used for training contains features such as **modelyear**, **km**, **ownerno**, and categorical variables like **make**, **transmission**, **Body Type**, and **Fuel Type**. The target variable is **Prize**, representing the price of a used car.
* **Preprocessing**: The dataset was preprocessed to handle missing values, scale numerical features, and encode categorical features using **one-hot encoding**.

**2. Train-Test Split**

* **Train-Test Split**: To evaluate model performance properly, we split the dataset into two parts:
  + **Training Set**: 80% of the data used to train the models.
  + **Testing Set**: 20% of the data used to evaluate the models after training.

This ensures that the models are trained on one portion of the data and tested on unseen data, simulating how the model will perform in real-world scenarios.

**3. Model Selection**

We selected the following five regression models for comparison:

* **Linear Regression**: A simple model that assumes a linear relationship between input features and the target variable. It is fast but might not capture complex patterns.
* **Decision Tree Regressor**: A tree-based model that splits the data based on feature values. It can model non-linear relationships but is prone to overfitting.
* **Random Forest Regressor**: An ensemble method that aggregates predictions from multiple decision trees, making it more robust and less prone to overfitting.
* **Gradient Boosting Regressor**: A powerful ensemble model that builds trees sequentially, each correcting the errors of the previous one. It can capture complex relationships and usually performs well for structured data.
* **XGBoost Regressor**: An optimized implementation of Gradient Boosting, known for its speed and performance. It includes regularization to prevent overfitting and is highly popular in competitive machine learning.

**4. Hyperparameter Tuning**

For each model, **GridSearchCV** was used to search for the best hyperparameters. The key hyperparameters tuned for each model were:

* **Decision Tree Regressor**: max\_depth, min\_samples\_split
* **Random Forest Regressor**: n\_estimators, max\_depth, min\_samples\_split
* **Gradient Boosting Regressor**: n\_estimators, learning\_rate, max\_depth
* **XGBoost Regressor**: n\_estimators, learning\_rate, max\_depth

GridSearchCV performs an exhaustive search over the specified hyperparameter grid, using cross-validation to select the best combination of parameters that minimizes the error.

**5. Model Training and Evaluation**

* **Training**: The models were trained on the **training dataset** (80%) using the best combination of hyperparameters from GridSearchCV.
* **Evaluation**: After training, each model was evaluated on the **test dataset** (20%) using the following performance metrics:
  + **Mean Absolute Error (MAE)**: Measures the average magnitude of errors in predictions. A lower MAE indicates better accuracy.
  + **Mean Squared Error (MSE)**: Penalizes large errors more than MAE. Like MAE, a lower value is better.
  + **R-squared (R²) Score**: Measures how well the model explains the variance in the target variable. A higher R² score indicates that the model explains more of the variability in the data.

**6. Performance Comparison**

The results of the models' evaluation on the test dataset are as follows:

| **Model** | **MAE** | **MSE** | **R² Score** |
| --- | --- | --- | --- |
| **Gradient Boosting** | 0.0076 | 0.0002 | 0.8896 |
| **XGBoost** | 0.0078 | 0.0002 | 0.8874 |
| **Random Forest** | 0.0090 | 0.0002 | 0.8620 |
| **Linear Regression** | 0.0089 | 0.0002 | 0.8573 |
| **Decision Tree** | 0.0106 | 0.0003 | 0.8075 |

* **Gradient Boosting Regressor** has the best performance, as it achieves the lowest **MAE** and **MSE**, and the highest **R² Score**.
* **XGBoost Regressor** is the second-best performer with a very similar R² score to Gradient Boosting.
* **Random Forest** also performed well but was slightly behind **XGBoost** and **Gradient Boosting**.
* **Linear Regression** and **Decision Tree** performed the worst, with **Decision Tree** showing the highest **MAE**.

**7. Conclusion and Model Selection**

Based on the evaluation metrics, **Gradient Boosting Regressor** is selected as the best model for car price prediction, followed closely by **XGBoost Regressor**. Both models showed excellent performance, with Gradient Boosting Regressor slightly outperforming XGBoost in terms of R² and error metrics.

* **Best Model**: **Gradient Boosting Regressor**
* **Runner-Up**: **XGBoost Regressor**