# Car Price Prediction

### Dataset Overview

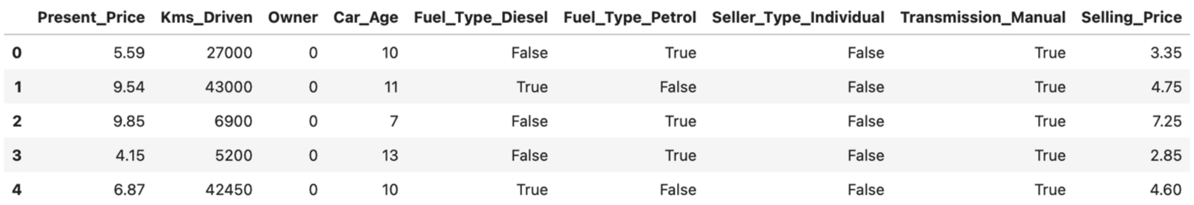
The dataset used in this study consists of records related to used cars, including information on car specifications, seller types, and prices. The dataset provides a diverse set of attributes that are crucial for predicting the resale value of used cars. The features can be used to explore relationships between a car's age, mileage, and fuel type with its selling price. The data reflects a balanced mix of various car models and seller types, allowing for comprehensive modeling of second-hand car prices.

**Link to dataset :** https://github.com/bkhushi9/Car-Price-Prediction/blob/main/car_data.csv

**Aim:** The aim of the car price prediction machine learning project is to accurately predict the resale value of used cars

### Data Preprocessing

1. The Car_Name feature was eliminated since it was thought to be insignificant
2. A new feature, Car_Age, was created by subtracting the Year column from the current year. This transformation captures the natural depreciation of cars over time, which plays a crucial role in determining their selling price.
3. Categorical features (Fuel_Type, Seller_Type, and Transmission) were converted into numerical values using one-hot encoding. This was necessary to allow the models to process these qualitative variables.
4. The dataset was split into training (80%) and testing (20%) sets, ensuring that the model was evaluated on unseen data to prevent overfitting.



| | Present_Price | Kms_Driven | Owner | Car_Age | Fuel_Type_Diesel | Fuel_Type_Petrol | Seller_Type_Individual | Transmission_Manual | Selling_Price |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 5.59 | 27000 | 0 | 10 | False | True | False | True | 3.35 |
| **1** | 9.54 | 43000 | 0 | 11 | True | False | False | True | 4.75 |
| **2** | 9.85 | 6900 | 0 | 7 | False | True | False | True | 7.25 |
| **3** | 4.15 | 5200 | 0 | 13 | False | True | False | True | 2.85 |
| **4** | 6.87 | 42450 | 0 | 10 | True | False | False | True | 4.60 |

### Model Selection

Three models were chosen for evaluation based on their distinct approaches to capturing the relationships between features and target values:

1. Linear Regression Model
2. K-Nearest Neighbors (KNN)
3. Decision Tree Regressor

### Model Training and Evaluation

Each model was trained on the training data and evaluated on the test data. We used three evaluation metrics:

1. **Mean Absolute Error (MAE)**: Measures the average magnitude of the errors in predictions, providing an intuitive sense of prediction accuracy.

2. **Mean Squared Error (MSE)**: Squares the error values, penalizing larger errors more heavily, which highlights cases where the model's predictions deviate significantly from the actual values.

3. **$R^2$ Score**: Reflects the proportion of variance in the target variable explained by the input features. A value close to 1 indicates a well-fitting model.

**Initial Model Performance**

Each model was trained and evaluated on the test dataset.

| Model | MAE | MSE | $R^2$ Score |
|---|---|---|---|
| Linear Regression Model | 1.21 | 3.78 | 0.84 |
| K-Nearest Neighbors (KNN) | 3.41 | 23.57 | 0.55 |
| Decision Tree Regressor | 0.78 | 1.46 | 0.93 |

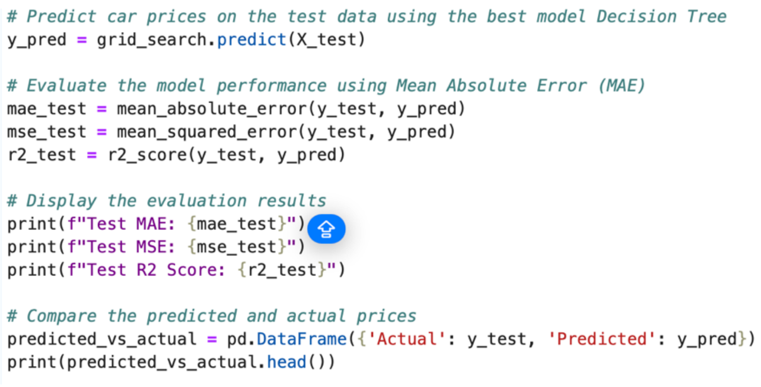
**Linear Regression**: The linear regression model exhibited reasonable performance with an MAE of 1.21 and an $R^2$ score of 0.84, indicating that it explained 84% of the variance in selling prices. However, it struggled to capture non-linear relationships, as evidenced by relatively high MAE and MSE scores.

**K-Nearest Neighbors**: KNN performed poorly in this context, with an MAE of 3.41 and a low $R^2$ score of 0.55, suggesting that the model was unable to adequately capture the relationships between the features and the target variable. This could be attributed to the complexity of the dataset, which may not be well-suited for KNN without further feature engineering.

**Decision Tree**: The Decision Tree model outperformed the other models with an MAE of 0.78 and $R^2$ score of 0.93. This performance indicates that the model captured most of the variance in the target variable and was particularly effective at modeling non-linear relationships.

**Model Evaluation**

To assess the performance of the best-performing model, the Decision Tree Regressor, the model was tested on unseen data using the test dataset. The predictions on the test set were compared to the actual values using several key evaluation metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and the $R^2$ Score. These metrics provide insight into the model’s prediction accuracy and its ability to generalize to new data.



```
# Predict car prices on the test data using the best model Decision Tree
y_pred = grid_search.predict(X_test)

# Evaluate the model performance using Mean Absolute Error (MAE)
mae_test = mean_absolute_error(y_test, y_pred)
mse_test = mean_squared_error(y_test, y_pred)
r2_test = r2_score(y_test, y_pred)

# Display the evaluation results
print(f"Test MAE: {mae_test}")
print(f"Test MSE: {mse_test}")
print(f"Test R2 Score: {r2_test}")

# Compare the predicted and actual prices
predicted_vs_actual = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(predicted_vs_actual.head())
```

### Hyperparameter Tuning

Given the superior performance of the Decision Tree, we proceeded to fine-tune its hyperparameters using GridSearchCV. The following parameters were optimized:

- max_depth: Limits the depth of the tree to prevent overfitting.
- min_samples_split: The minimum number of samples required to split an internal node.
- min_samples_leaf: The minimum number of samples required at a leaf node.
- max_features: The number of features to consider when looking for the best split.

The GridSearchCV yielded the following optimal parameters:

- max_depth=None
- min_samples_split=2
- min_samples_leaf=1
- max_features=None

These parameters allowed the Decision Tree to achieve a further reduction in MAE, improving model performance.

### Final Model Evaluation

The fine-tuned Decision Tree model was retrained with the optimized parameters and evaluated on the test set. The final results are as follows:

- **Test MAE**: 0.744
- **Test MSE**: 1.46
- **Test $R^2$**: 0.93

This demonstrates that the Decision Tree, when optimized, provides highly accurate predictions of car selling prices. The low MAE and high $R^2$ score suggest that this model is particularly well-suited for capturing the complex interactions between features such as Car_Age, Kms_Driven, Fuel_Type, and Transmission.

## Conclusion

This study evaluated the performance of three machine learning models for predicting car selling prices. The results show that the Decision Tree Regressor, after hyperparameter tuning, outperformed both Linear Regression and K-Nearest Neighbors. The Decision Tree's ability to capture non-linear relationships and its adaptability to varying feature importance made it the most suitable model for this dataset.

Further improvements could be achieved by exploring ensemble methods like Random Forests or Gradient Boosting, which may reduce overfitting and improve generalization. Additionally, feature importance analysis could provide further insights into which variables most significantly affect car prices.