**Car Price Prediction**

**ABOUT DATASET**

This dataset is used to create the project that builds a machine learning model to predict car prices based on features such as brand, year, engine size, mileage, fuel type, and transmission. It involves data preprocessing, exploratory data analysis, model building, and evaluation.  
**Source: This project is inspired by a car price prediction dataset on Kaggle. https://www.kaggle.com/datasets/zafarali27/car-price-prediction/data#:~:text=more\_vert-,Car%20Price%20Prediction,-Car%20Price%20Prediction**

**PROJECT-OVERVIEW**

This machine learning project is designed to predict car prices using a dataset that includes various car features. The goal is to develop a regression model that can accurately estimate a car's price based on these characteristics.

**BASED ON STEPS**

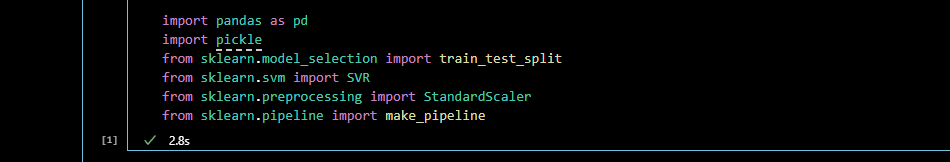
1. Data collection & data Understanding

2. Data splitting

3. Model Train (SVM)

4. Model Store in pickle

**LIBRARIES:**



import pandas as pd

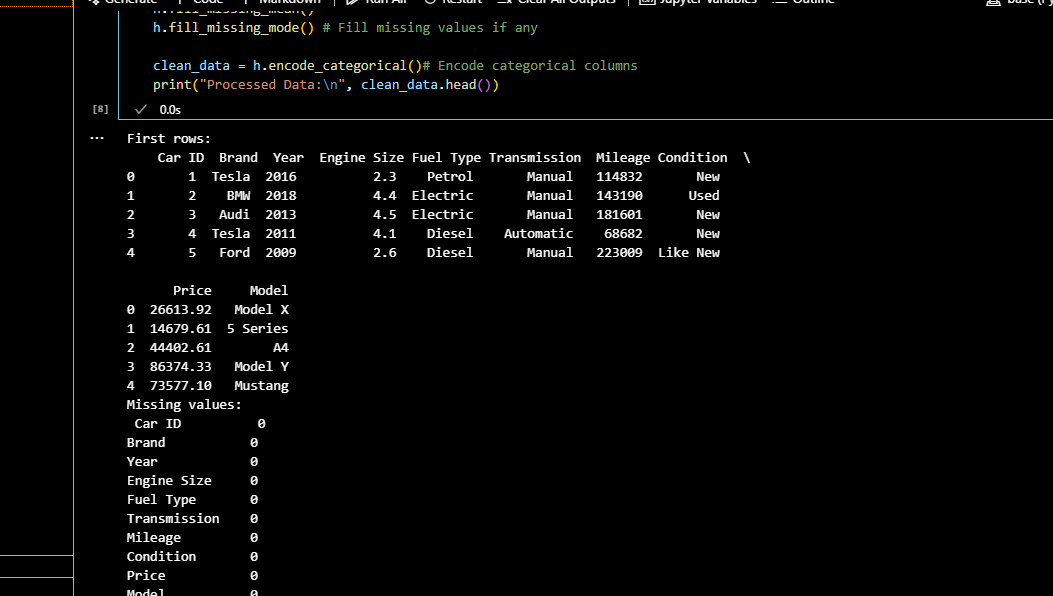
from sklearn.model\_selection import train\_test\_split

import pickle

from sklearn.svm import SVR

from sklearn.preprocessing import StandardScaler

from sklearn.pipeline import make\_pipeline

**STEP 01 (Data Collection And Preprocessing):**

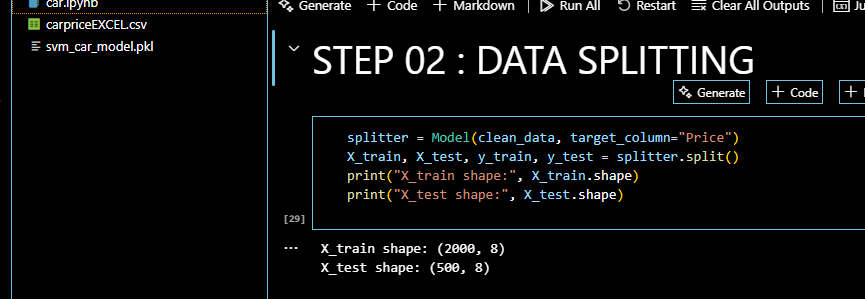
The dataset is typically collected from a structured source like Kaggle, CSV files, or a car database. It includes features such as:  
• Brand  
• Year  
• Engine Size  
• Mileage  
• Fuel Type  
• Transmission  
• Car Price (target)

Once loaded, we inspect the data using:  
1. df.head() — preview the top rows  
2. df.info() — understand datatypes and missing values  
3. df.describe() — view summary statistics  
4. df.isnull().sum() — check for nulls  
5. df.corr() — correlation between numeric variables

This step helps to:  
- Detect data quality issues  
- Identify feature importance  
- Guide preprocessing and modeling decisions

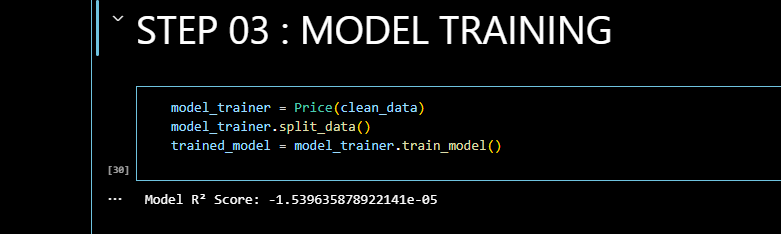
**STEP 02 (Data Splitting):**

Splits the dataset into:  
• Training Set (80%): To train the model  
• Testing Set (20%): To evaluate model performance on unseen data  
  
Why it matters:  
- Prevents overfitting  
- Ensures fair model evaluation



**STEP 03 (Model Training - Support Vector Machine):**

Support Vector Regression (SVR) is a version of SVM used for predicting continuous values.  
1. Feature Scaling using StandardScaler (SVM is sensitive to feature scale)  
2. Model Training using SVR  
3. Hyperparameter Tuning using GridSearchCV for best results  
  
Why SVM?  
- Works well for both linear and non-linear relationships  
- Robust to overfitting (especially with proper regularization)



**STEP 04 (Model Storage - Pickle):**

Once trained and tested, the model is serialized using pickle, allowing reuse without retraining.  
Benefits of Pickle:  
- Store models for deployment  
- Share models across environments  
- Save time during inference

