Second Ride — Used Car Price Prediction Report

📘 Project Overview

This project focuses on predicting used car prices using a complete data science workflow — from raw data to a final deployed model. The dataset includes various attributes such as brand, mileage, engine specifications, transmission, and condition, providing a strong base for building a reliable predictive model.

🧾 Step 1: Data Loading & Initial Exploration

- Dataset loaded successfully with **4,009 rows** and **12 columns**.
- **Key features:** brand, model, model_year, milage, fuel_type, engine, transmission, ext_col, int_col, accident, clean_title, price.
- Most columns are categorical, while others like price, milage, and model_year are numerical.
- No loading issues detected, and dataset size ≈ **2.58 MB**.

✅ The dataset is ready for cleaning and preprocessing.

🧹 Step 2: Data Cleaning & Type Conversion

- Removed symbols ($, mi, ,) from price and milage → converted to numeric.
- Standardized fuel_type and transmission labels.
- Converted all columns to proper types, reducing memory usage from **2.58 MB** → **0.72 MB**.
- Filled missing values:
  - fuel_type by brand mode → ‘OTHER’
  - accident → ‘None reported’
  - clean_title → ‘Unknown’
- Removed 67 cars with model_year < 2000.
- Removed outliers in price (250 rows) and milage (61 rows) using the IQR method.

Final dataset shape: 3,631 rows, clean and consistent.

✅ Data prepared for feature engineering.

🧠 Step 3: Feature Engineering

Extracted features:

- Derived **hp (horsepower)** and **engine_displacement** from text.
- Created binary flag **is_v_engine** for V-type engines.

Handled missing values:

- Filled hp by brand mean.
- Filled engine_displacement by brand mode → median.

Derived new features:

- **Vehicle_Age = 2025 - model_year**
- **Mileage_per_Year = milage / Vehicle_Age**

Binned and encoded:

- Vehicle_Age → 4 quantiles (New, Mid, Old, Very Old)
- milage → 4 quantiles (Low, Medium, High, Very High)
- Applied one-hot encoding on binned features.

Condition encoding:

- accident → binary **Accident_Impact**
- clean_title → binary (1 = clean)

Dropped redundant columns: model, model_year, engine, int_col, ext_col, accident

Final dataset shape: (3,630, 18) | Size: 0.41 MB

✅ Data is now numeric, feature-rich, and model-ready.

📊 Step 4: Exploratory Data Analysis (EDA)

Correlation Analysis:

- hp & engine_displacement → strong positive correlation with price
- Vehicle_Age & milage → negative correlation with price

Feature Distributions:
Moderate skewness observed in most numerical variables.

Categorical Insights:

- Gasoline cars dominate; hybrids come next.
- Automatics are most common and typically higher priced.

Brand Insights:

- Luxury brands (Lexus, BMW, Mercedes) retain higher prices.

Condition-based patterns:

- Accident_Impact = 0, clean_title = 1, is_v_engine = 1 → higher prices.

✅ EDA confirmed key relationships that guide model design.

🤖 Step 5: Machine Learning Modeling

Preprocessing:

- Converted categorical features (brand, fuel_type, transmission, etc.) to numeric.
- Split dataset: **80% train / 20% test**.
- Normalized numeric features with **StandardScaler**.

Trained models:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Random Forest Regressor
- XGBoost Regressor

Evaluation Metrics:

- **R^2:** measures model explanatory power
- **RMSE:** measures average prediction error
- **MAE:** measures average absolute error

Results Summary:

Model	R^2 Train	R^2 Test	RMSE Train	RMSE Test	MAE Train	MAE Test	Fit Status
XGBoost	0.941355	0.869787	5092.268930	7638.526966	3712.012655	5418.589414	Good Fit
Random Forest	0.813068	0.774773	9091.544896	10045.986124	6515.067072	7133.375907	Good Fit
Linear Regression	0.718447	0.714633	11157.738102	11307.955249	8218.809357	8208.811167	Good Fit
Lasso Regression	0.718447	0.714633	11157.738102	11307.960749	8218.809593	8208.818835	Good Fit
Ridge Regression	0.718447	0.714611	11157.739850	11308.387854	8218.661032	8209.288014	Good Fit

✅ **XGBoost achieved the highest accuracy and lowest errors.**

💾 Step 6: Model Saving & Deployment

- Saved final model as **xgboost_used_car_price_model.pkl**
- Saved **StandardScaler** for consistent data preprocessing during deployment.

💡 The model can be easily integrated into a web app or API for real-time used car price prediction.