Used Car Price Prediction Using ML

🗣️ **"Hey, Looking for the right price on a specific car model? No worries—let me ask my ML agent and get you an accurate estimate in seconds!"**

By

Dudekula Abid Hussain

Email - dabidhussain2502@gmail.com | Kaggle - https://www.kaggle.com/abiabid | Github - https://github.com/Abi-Abid

Introduction

In this project, we aim to build an accurate Machine Learning (ML) model that predicts the prices of used cars based on their specifications. While this may initially sound simple, the real-world applications are highly valuable. Imagine you're part of a car dealership, and a customer walks in asking for the price of a specific car model with certain features. With dozens of models and varying configurations, determining an accurate price on the spot can be challenging.

Now, picture having a smart system or ML model on your tablet—where you simply enter the car's details, and within seconds, it provides a reliable price estimate. Sounds exciting, right? That's exactly what this project sets out to achieve.

Although this project starts as a prototype with a relatively small dataset, it lays a strong foundation for analyzing features and understanding how different car specifications influence pricing. It also serves as a practical introduction to various ML models and the mathematical intuition behind them. This kind of system not only supports smarter decision-making for businesses but also enhances the customer experience.

Data Overview

Each row in the dataset represents a **car model and its specifications**— these are typically *used car specs* or older models meant for analysis or price prediction. It is a good volume of data set we have, the data set contains 205 rows and 25 features. Understanding these features is paramount. Link to access the data can be found here. Below table gives in depth understanding of each feature.

S.No	Feature	Description
1.	symboling	Symboling is like a risk rating for a car, mainly used by insurance companies. Imagine like you're insuring a car, some cars are riskier than others. A sports car that goes very fast might be involved in more accidents or cost more to repair. So, insurance companies give that car a **higher symboling score**, like +3. On the other hand, a safe, family-friendly car like a small sedan might have a low or even negative symboling score like -2, meaning it's less risky to insure.
2.	normalized-losses	Normalized losses show how much money insurance companies have lost, on average, for a specific car model — based on past insurance claims.
3.	make	This feature represents the manufacturer, like which company made the car.
4.	fuel-type	This represents the fuel type, like what fuel is used (gas, petrol, diesel)

5.	aspiration	Type of engine air intake — standard or turbocharged. [This gives how air enters the egine, Standard (std) in natural way or Turbo, a turbo forces more air into the engine, boosting the power and performance.
6.	num-of-doors	Number of doors (two or four).
7.	body-style	Car body type — sedan, hatchback, convertible, etc.
8.	drive-wheels	Type of drive system — front-wheel, rear-wheel, or 4-wheel drive.
9.	engine-location	Location of the engine — front or rear.
10.	wheel-base	Distance between front and rear wheels (in inches).
11.	length	Length of the car (in inches).
12.	width	Width of the car (in inches).
13.	height	Height of the car (in inches).
14.	curb-weight	Weight of the car without passengers or cargo.
15.	engine-type	Type of engine — ohc, ohcv, rotor, etc. [refers to the design or configuration of the engine's internal components. It indicates how the engine is built and how its cylinders are arranged or function. For example - ohc (Overhead Camshaft), dohc (Double Overhead Camshaft) - more power and used in performance engines, rotor - found in rotary engines like Mazda RX series.]
16.	num-of-cylinders	Number of engine cylinders — two, four, six, etc.
17.	engine-size	Total engine displacement (in cc); affects power.
18.	fuel-system	Type of fuel delivery system — mpfi, 2bbl, etc. [The **fuel system** refers to the mechanism in a car that stores and delivers fuel to the engine for combustion. **mpfi** (Multi-Point Fuel Injection): More efficient, injects fuel into each cylinder, **2bbl** (2-Barrel Carburetor): An older method, uses two openings to mix fuel and air, **1bbl** (1-Barrel Carburetor): Similar, but only one opening—less efficient, **spdi**, **spfi**, etc.: Variants of fuel injection systems.
19.	bore	Diameter of engine cylinder (in inches); relates to power.
20.	stroke	Distance the piston travels (in inches); relates to torque.
21.	compression-ratio	Ratio of cylinder volume before and after compression.
22.	horsepower	Power output of the car's engine.
23.	peak-rpm	Engine revolutions per minute at peak horsepower.
24.	city-mpg	Fuel efficiency in city driving (miles per gallon).
25.	highway-mpg	Fuel efficiency on the highway (miles per gallon).
26.	price	Final selling price of the car (target variable).

Methodology

Before applying machine learning models, we need to preprocess the data, which includes:

- Identify and handle missing values in the dataset. We can use imputation or drop missing values based on the context.
- Features like "Make," "Fuel Type," and "Body Style" are categorical. These need to be encoded using techniques like **one-hot encoding** or **label encoding** so that the machine learning models can work with them.
- Features like "Horsepower," "Curb Weight," and "MPG" should be scaled to ensure that no feature dominates the model due to different ranges.

We will be apply several machine learning models to predict the car prices (like Linear regression, Decision Forest, Random forest, gradient boosting..etc.,). After training the models, we will evaluate their performance using the metrics like 'R2 Score', 'MAE', and 'RMSE'.

Exploratory Data Analysis

For the given dataset, the first step involved cleaning the data by handling missing values and correcting incorrect data types. After preprocessing, a statistical summary was analyzed, leading to the following insights - The dataset consists of 201 cleaned entries, representing individual used cars. Starting with the `symboling` column, which indicates insurance risk, we observe values ranging from -2 to +3. The average symboling score is approximately 0.84, with most cars falling between 0 and 2, suggesting moderate insurance risk for most vehicles.

The `normalized-losses` column, representing insurance loss risk, varies significantly between 65 and 256, with a mean of 122. This wide range indicates diverse risk levels across car models, though most cars fall between 101 and 137 (25th to 75th percentile), showing a moderate loss expectation overall.

Physical dimensions such as `wheel-base`, `length`, `width`, and `height` provide insight into car size. The average wheelbase is around 98.8 inches, with a range from 86.6 to 120.9 inches. Car lengths vary from compact (141.1 inches) to long (208.1 inches), with a mean of 174.2 inches. The width ranges between 60.3 and 72 inches, with an average of 65.9 inches, while the height spans from 47.8 to 59.8 inches, indicating a variety of car types from low sports cars to taller sedans or wagons.

The curb-weight (weight of the car without passengers or cargo) averages at 2555 kg, with a substantial standard deviation (~517), indicating variation from lightweight compact cars (minimum of 1488 kg) to much heavier vehicles (maximum of 4066 kg). The engine-size also shows significant spread, from 61 to 326 cc, with an average of ~127, implying the dataset includes both small, fuel-efficient engines and powerful ones.

For engine internals, bore and stroke describe the cylinder dimensions. The bore (cylinder diameter) averages 3.33 inches and stroke (piston travel length) is around 3.26 inches, both showing low variability. Compression-ratio, a measure of engine efficiency and performance, ranges from 7.0 to 23.0, with most values clustered below 10.

The horsepower values show a wide range, from a modest 48 to a powerful 262, with an average of 103.4. This confirms the presence of both basic economy cars and high-performance vehicles. Peak-rpm, the engine speed at maximum power, averages 5118 rpm, again showing typical passenger car values, with a maximum of 6600.

Fuel efficiency is captured through city-mpg and highway-mpg. Cars in the dataset range from as low as 13 mpg in the city to as high as 49 mpg, and 16 to 54 mpg on the highway. The average values (~25 city, ~31 highway) reflect moderate fuel efficiency, suitable for everyday driving.

Finally, the price column reveals a right-skewed distribution of car prices. The average price is around $13,200, but with a large standard deviation, confirming the skewness — most cars are priced in the low to mid-range, with a few high-end models pulling the average up.

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	
count	201.000000	201.0	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201
mean	0.840796	122.0	98.797015	174.200995	65.889055	53.766667	2555.666667	126.875622	3.330692	3.256874	10.164279	103.405534	5117.665368	25.179104	30.686567	1320
std	1.254802	31.99625	6.066366	12.322175	2.101471	2.447822	517.296727	41.546834	0.268072	0.316048	4.004965	37.365700	478.113805	6.423220	6.815150	7947
min	-2.000000	65.0	86.600000	141.100000	60.300000	47.800000	1488.000000	61.000000	2.540000	2.070000	7.000000	48.000000	4150.000000	13.000000	16.000000	5118
25%	0.000000	101.0	94.500000	166.800000	64.100000	52.000000	2169.000000	98.000000	3.150000	3.110000	8.600000	70.000000	4800.000000	19.000000	25.000000	7775
50%	1.000000	122.0	97.000000	173.200000	65.500000	54.100000	2414.000000	120.000000	3.310000	3.290000	9.000000	95.000000	5125.369458	24.000000	30.000000	10295
75%	2.000000	137.0	102.400000	183.500000	66.600000	55.500000	2926.000000	141.000000	3.580000	3.410000	9.400000	116.000000	5500.000000	30.000000	34.000000	16500
max	3.000000	256.0	120.900000	208.100000	72.000000	59.800000	4066.000000	326.000000	3.940000	4.170000	23.000000	262.000000	6600.000000	49.000000	54.000000	45400

Fig 1 - Statistical Summary of Numeric features

After analyzing the correlation heat map of the numerical features: **Price shows a strong positive correlation** with several features: **Engine size** has the **highest correlation with price**, suggesting that cars with larger engines tend to be more expensive. **Curb weight** and **horsepower** are also **strongly correlated with price**, indicating that heavier and more powerful cars are priced higher. **Width** and **length** show moderate positive correlations with price, implying that larger cars in general are more expensive.

On the other hand, **city-mpg** and **highway-mpg** (miles per gallon) have a **negative correlation with price**, meaning that cars which are more fuel-efficient tend to be less expensive. This aligns with real-world patterns where luxury and performance cars often have lower fuel economy. Features like **compression ratio** and **peak-rpm** show **low or negligible correlation** with price, suggesting they might not significantly influence the car's cost and could be candidates for feature reduction depending on the modeling approach. **Normalized losses** has a mild positive correlation with price, indicating that cars with higher historical insurance losses might also be in the higher price segment, although this relationship is not very strong.

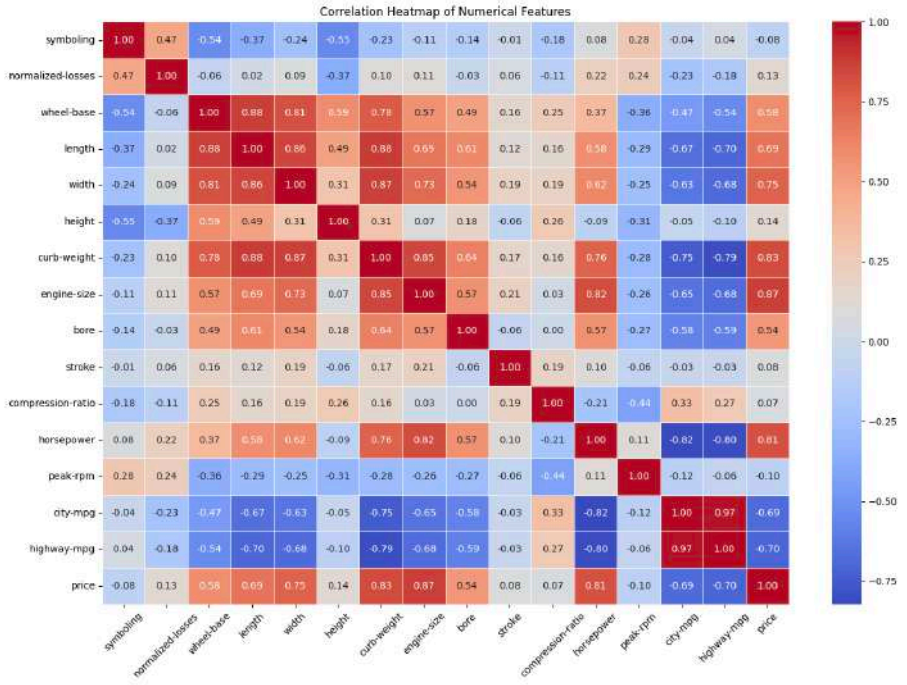

Fig 2 - Correlation Heat Map for Numeric features

The categorical feature summary reveals valuable patterns about the distribution of car types in the dataset. The most common car make is **Toyota**, which appears 32 times, making it the most frequently occurring brand among the 201 entries. Most vehicles use **gasoline** as the fuel type, with **181** out of 201 cars using gas, while only a few use diesel. Similarly, the majority of cars have **standard (std)** aspiration, suggesting they don't use turbocharging. When it comes to doors, **four-door** cars dominate the dataset. The most popular **body style** is **sedan**, showing that standard passenger cars are more prevalent than sportier or specialty models. Most vehicles are **front-wheel drive (fwd)** and have the **engine located at the front**, which is typical for mainstream cars. The most common **engine type** is **OHC (Overhead Camshaft)**, and most cars have **four cylinders**, which aligns with the fuel efficiency trend. Lastly, **MPFI (Multi-Point Fuel Injection)** is the most frequent **fuel system**, reflecting a common fuel delivery method in modern engines. These distributions indicate that the dataset primarily represents typical, mass-market, fuel-efficient cars rather than high-performance or luxury vehicles.

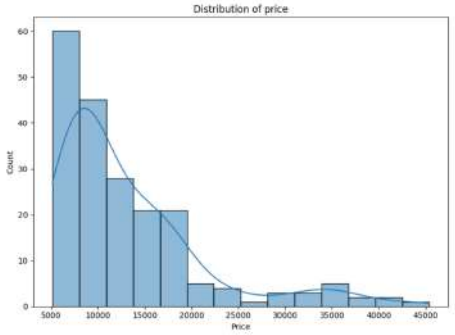

Fig 3 - Price Distribution.

From fig 3 we can notice that the price is right-skewed distribution of car prices. With most number of cars are in the price range of around $5000 to $10000.

In ML, dealing with a right-skewed target variable such as our problem - car price is crucial for building accurate and reliable models. This skewness can negatively impact model performance for several reasons. Firstly, many ML algorithms, especially linear regression—operate under the assumption that the target variable is normally distributed. When this assumption is violated, the models can become biased and less effective. Secondly, a skewed distribution may cause models to underestimate higher values, focusing too heavily on the majority of lower-priced data points. This leads to poor predictive performance, particularly for more expensive cars. Finally, skewness can inflate evaluation metrics such as RMSE (Root Mean Squared Error), as large deviations from true high-price values disproportionately affect the error.

To address this, one common preprocessing technique is to apply a **log transformation** to the target variable. This transformation compresses the range of higher values and spreads out lower ones, thereby reshaping the distribution into a more normal, bell-shaped curve. In practice, the `np.log1p()` function is often used to safely compute the logarithm even when values are zero. After training the model using the transformed target (e.g., `price_log`), predictions can be converted back to their original scale using the

inverse transformation `np.expm1()`. This approach helps stabilize variance, improve model accuracy, and ensure more balanced learning across the entire price range.

A scatter plot for the features distribution against the price is analyzed, the insights show that:

The scatter plots illustrate the relationship between each numerical feature and the target variable — **car price**. From these visualizations, we can observe that some features such as **engine-size**, **curb-weight**, and **horsepower** show a strong positive correlation with price. This suggests that these features are good predictors and should be retained during model training.

On the other hand, features like **compression-ratio**, **height**, and **peak-rpm** do not show a clear trend with price, indicating they may have weaker predictive power or may need transformation or interaction with other variables.

We also notice the presence of **outliers** in many of the plots. These extreme values can impact the model's learning ability and may lead to **overfitting or underfitting**. These should be examined carefully and possibly treated or removed during preprocessing.

Additionally, the scatter plots confirm **non-linear relationships** in some cases. Therefore, using **non-linear models** or applying **feature transformations** (e.g., polynomial or log transformations) might help improve prediction accuracy.

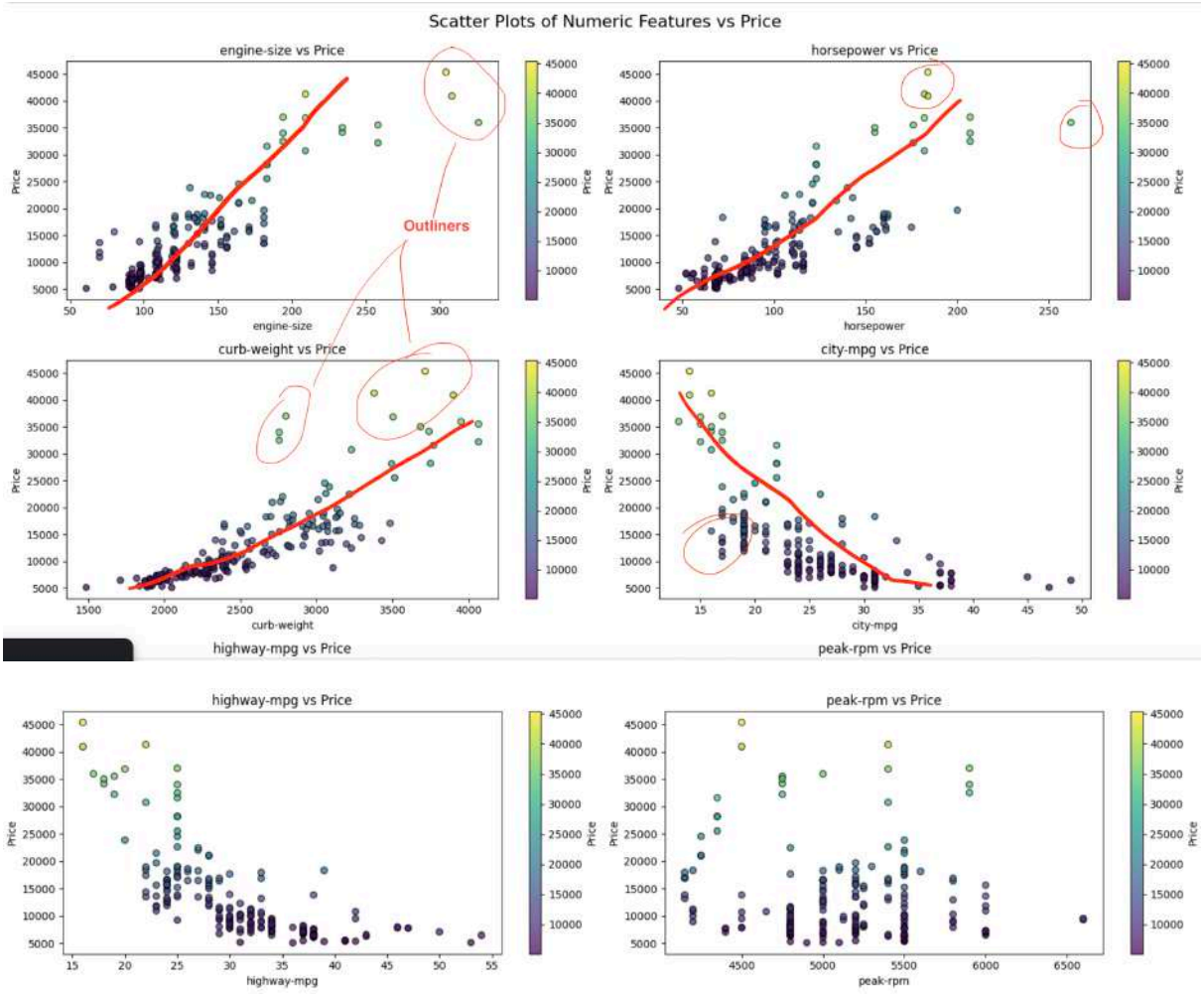

Fig 4 - Scatter plot price distribution for some Key numeric features

After analyzing the price distribution for key categorical features—**fuel type**, **body style**, and **drive wheels**—several observations were made:

- **Fuel Type**: Gas-powered cars dominate the dataset and exhibit a wide price range with several outliers.
- **Body Style**: Sedans are the most common and also show a high number of price outliers.
- **Drive Wheels**: Both FWD and EWD vehicles show scattered price values, again with visible outliers.

These findings highlight the need to address outliers in the dataset, particularly in the `price` column, to prevent distortion in machine learning models.

Outlier Handling: Extreme values in the `price` column were removed using the Interquartile Range (IQR) method to improve model stability. **Categorical Encoding**: Non-numeric features such as `fuel-type`, `body-style`, and `drive-wheels` were converted using one-hot encoding to prepare the data for modeling. These preprocessing steps are crucial to ensure accurate and unbiased model training. The cleaned and encoded data is now ready for scaling and model development.

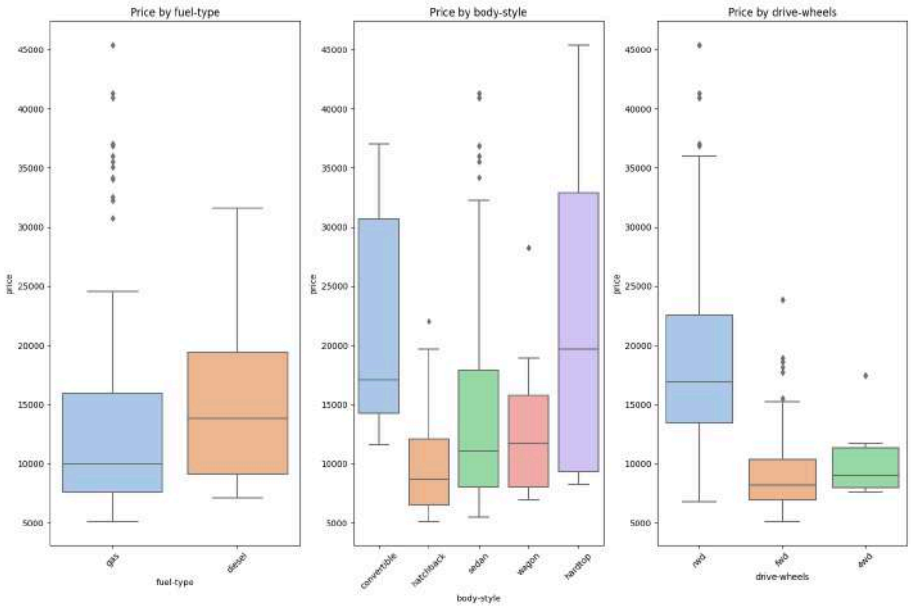

Fig 5 - Box plot price distribution for some key categorical features

Detecting and removing extreme price outliers, This will remove rows where price is extremely high or low across any category (not just fuel type or body style), helping you prevent model distortion. The result, reduced from 205 rows to 185 rows.

```
# Detect and remove extreme price outliers

"""This will remove rows where price is extremely high or low across any category (not just fuel type or body style),
helping you prevent model distortion."""

Q1 = df['price'].quantile(0.25)
Q3 = df['price'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter the dataset
df = df[(df['price'] >= lower_bound) & (df['price'] <= upper_bound)]
df.shape
```

```
(185, 26)
```

Fig 6 - Code and result after handing outliers with respect to price

Model Development and Evaluation

In the model development phase, the cleaned and preprocessed dataset was prepared for training by separating the target variable (car price) from the input features in 70-30, 70% train and 30% test. The dataset consisted of both numerical and categorical features. To ensure compatibility with ML algorithms and to improve performance, all categorical variables were encoded using one-hot encoding, which allowed the model to understand non-numeric features by converting them into binary indicator variables. Next, feature scaling was performed using the StandardScaler technique to normalize the numerical data. This step was crucial because many machine learning algorithms are sensitive to the scale of the input features—particularly those that rely on distance calculations like linear regression and support vector machines. Once the data was encoded and scaled, it was split into training and testing sets to evaluate the model's generalization performance. This structured approach ensured that the model would not only perform well on known data but also be robust when predicting on unseen samples. By applying these preprocessing steps, we laid a strong foundation for building accurate and reliable machine learning models for predicting used car prices.

A wide range of regression models were selected for experimentation, including both linear and non-linear approaches. These models included Linear Regression, Polynomial Regression (with degrees 2 and 3), Ridge Regression, Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, AdaBoost Regressor, and Support Vector Regressor (SVR). The inclusion of these models allowed us to compare simple linear approaches with more complex ensemble and kernel-based methods to evaluate how well each could capture the underlying relationships in the data. These models were applied in their basic forms without hyper parameter tuning. The rationale behind this choice was due to the relatively small size of the dataset—performing extensive hyper parameter tuning might lead the models to overfit by learning the data patterns too specifically, which

would reduce their ability to generalize to new, unseen data. This baseline comparison provided valuable insights into which model types are more suitable for used car price prediction in scenarios with limited data, and laid a strong foundation for future work involving hyper parameter optimization and model refinement on larger datasets.

	Model	R2 Score	MAE	RMSE
3	Ridge Regression	8.797688e-01	1.374886e+03	1.633919e+03
8	AdaBoost	8.616969e-01	1.258984e+03	1.752419e+03
6	Random Forest	8.500778e-01	1.264869e+03	1.824546e+03
7	Gradient Boosting	8.475209e-01	1.305238e+03	1.840040e+03
4	Lasso Regression	8.440617e-01	1.457035e+03	1.860794e+03
0	Linear Regression	8.341434e-01	1.493497e+03	1.919059e+03
5	Decision Tree	7.837417e-01	1.469044e+03	2.191333e+03
9	SVR	-1.834956e-02	3.564467e+03	4.755218e+03
1	Polynomial (Deg=2)	-4.285060e+18	5.080238e+12	9.754398e+12
2	Polynomial (Deg=3)	-1.643961e+21	1.082581e+14	1.910591e+14

Fig 7 - Final results of all ML models used for evaluation

Among all the models tested, **Ridge Regression** performed the best overall, achieving the highest R^2 score (0.879) and maintaining relatively low error values (MAE ≈ 1374 and RMSE ≈ 1634). This suggests that Ridge was able to generalize well to the data while effectively handling multicollinearity through regularization. **AdaBoost** also showed strong performance with a slightly lower R^2 (0.862) but had the **lowest MAE**, making it a good choice for minimizing average prediction error. **Random Forest** and **Gradient Boosting** followed closely, demonstrating that ensemble methods can provide solid predictive power, although with slightly higher errors than Ridge and AdaBoost.

Lasso Regression and **Linear Regression** offered decent but weaker performance compared to the top models, indicating that while linear relationships exist, regularization alone (Lasso) or a simple linear fit may not fully capture the complexity of the data. **Decision Tree Regressor** showed further decline in performance, likely due to its tendency to overfit small datasets without pruning or tuning.

The **SVR (Support Vector Regressor)** performed poorly, with a negative R^2 and very high error values, indicating that the model failed to learn meaningful patterns from the data. Most notably, the **Polynomial Regression models (degrees 2 and 3)** drastically underperformed, with enormous error values and highly negative R^2 scores. This confirms that polynomial models severely overfit the small dataset, making them unsuitable without proper regularization or tuning.

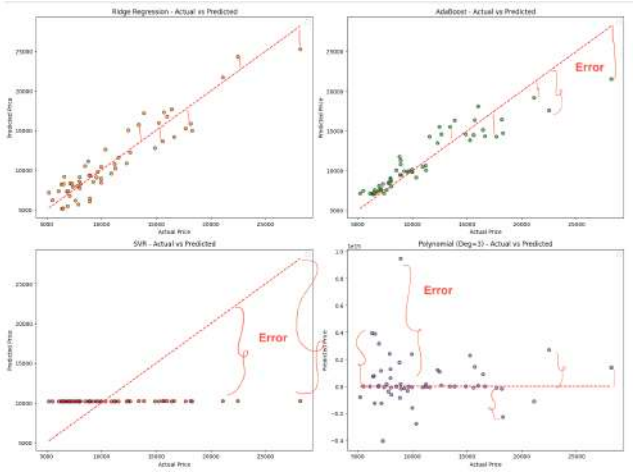

Fig 8 - Actual vs Predicted values visual representation

From the above graphs representing the actual vs. predicted values for the top two models — 'Ridge Regression' and 'AdaBoost' — and the least performing models — 'SVR' and 'Polynomial (Deg=3)' — we can observe a clear contrast in performance. In the plots for Ridge Regression and AdaBoost, the scatter points lie relatively close to the red dashed line (which represents perfect predictions), indicating that these models predicted the car prices with high accuracy and minimal error. In contrast, the plots for SVR and Polynomial (Deg=3) show scatter points that are widely spread and far from the ideal prediction line, suggesting that these models struggled to capture the underlying patterns in the data, resulting in poor prediction accuracy.

Key Insights / Conclusion

Based on the findings and comprehensive analysis:

- It was observed that the original dataset comprised 205 rows and 26 columns, representing 205 cars with 26 different specifications. Upon analyzing the price distribution, it was noted that the data was right-skewed, indicating a higher number of low-priced cars compared to expensive ones.
- Additionally, scatter plots of both numerical and categorical features showed a significant presence of outliers, which could negatively impact model performance. To address this, outliers were treated using the Interquartile Range (IQR) method, which

resulted in a reduced dataset of 185 rows.

- Despite the reduction, this cleaned dataset contained more consistent and evenly distributed price values, contributing to improved model stability.

After preprocessing, including one-hot encoding of categorical variables and standardization of numerical features, the dataset was split into training and testing sets with a **70-30 ratio.** Several machine learning models were trained and evaluated for performance.

- Among these, **Ridge Regression and AdaBoost Regressor demonstrated the best results,** achieving approximately **87% accuracy (R^2 score)** and comparatively lower MAE and RMSE values.
- On the other hand, Polynomial Regression (Degree 3) and Support Vector Regressor (SVR) performed poorly, with high errors and low predictive accuracy.

Due to the relatively small size of the dataset, no hyperparameter tuning was applied during model training. This decision was made to avoid the risk of overfitting or underfitting, ensuring that models learned the underlying data patterns without overcomplicating them. Ultimately, Ridge Regression emerged as the most reliable model, offering strong performance and serving as a solid foundation for future expansion as more data becomes available.

In practical application, especially in a used car dealership context, **the Ridge Regression model can be effectively used to predict the selling price of a car based on its specifications.** This not only aids buyers in making informed decisions but also helps sellers in setting fair and accurate prices.

> **Thanks for stopping by! If you found this helpful or have suggestions, feel free to leave feedback here. Happy learning and exploring new data!** 👋