Final Report of DS 675-005

Price Prediction of Used Cars

Ram Jatin Vegulla, Venkata Surya Moparthi, Mohan Sai Bandarupalli, Prem Kiran Reddy Kallem

Keywords: Exploratory data analysis, Data Cleaning, Feature Engineering, Data Visualization, Data Preprocessing, Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE), Regression Models, XGBoost Regressor

Abstract: The objective of this project is to develop a machine learning model for predicting used car prices using the Craigslist Cars and Trucks dataset. The dataset, obtained from Kaggle, contains valuable information about various car attributes, including year, manufacturer, model, condition, cylinders, fuel type, odometer reading, title status, transmission type, drive, and vehicle type. The goal is to leverage this dataset to create a predictive model that can estimate the price of used cars based on these features. The project involves a series of steps, including data exploration, cleaning, and preprocessing, followed by the implementation of a machine learning model. The selected model, a linear regression algorithm, will be trained on a subset of the dataset and evaluated for its predictive performance. The results will be visualized and analysed to assess the model's accuracy and effectiveness in predicting used car prices. This project aims to provide insights into the factors influencing used car prices and demonstrate the application of machine learning techniques in the automotive domain.

Dataset: https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data

Code Link: https://colab.research.google.com/drive/1uG-yFAsCtOEZaZhTeqKHruy3eqGg4S13?usp=sharing

INTRODUCTION

The used car market is a complex and dynamic environment where various factors contribute to the determination of vehicle prices. Understanding and predicting these prices can be invaluable for both buyers and sellers. In this context, machine learning offers a powerful toolset to analyse and model the relationships between different car attributes and their corresponding prices.

In the following sections, we will delve into the details of the dataset, the steps taken for data preprocessing, the implementation of the machine learning model, and a comprehensive analysis of the results obtained.

EXPLORATORY DATA ANALYSIS:

The purpose of this is to perform an in-depth Exploratory Data Analysis (EDA) on a dataset containing information about used cars. The dataset was obtained from Used Cars dataset, and it includes details such as car brand, condition, fuel type, transmission, and more.

To take a closer look at the data we took help of ".sample()" function of pandas library which returns the given number of observations of the data set. Our dataset comprises 427880 observations and 12 characteristics. i.e., Total Rows: [426880]. Total Columns: [26]. It is good practice to know the columns and their corresponding data types along with finding whether they contain null values or not. Data has float and integer values. After that we have found the missing values is several columns of our dataset.

DATA CLEANING

Data cleaning is a crucial step in preparing data for machine learning algorithms. This process involves dealing with inconsistencies, missing values in the data to ensure its quality and accuracy. There are several common techniques used in data cleaning:

Handling Missing Values: There are several methods to deal with missing values, such as mean/median/mode imputation, regression imputation, and random sampling imputation. Alternatively, one can choose to delete rows or columns with missing values, although this approach can result in loss of data. So, in our dataset we have converted all missing values as 'unknown'.

Handling Duplicate Values: The total number of duplicated rows in the dataset is 53043. This indicates that there are entries that are exact duplicates of other entries in the dataset. Depending on the nature of the data, determine whether it is appropriate to keep or remove duplicated rows. Considerations include the impact on analysis, data integrity, and the source of duplication. By eliminating duplicate values, we will get new shape of data.

Categorical Data Cleaning: This report aims to analyze the distribution of car manufacturers within the used cars dataset. and the focus of this analysis is on the 'manufacturer' column. From this report we can observe that ford has highest count of 69,597 entries and land rover has lowest count of 21. Similar transformations were applied to region and model columns. An overview of the unique values in each remaining column after transformations: Region: 51 unique values, Price: 15,274 unique values, Year: 107 unique values, Manufacturer: 21 unique values, Model: 51 unique values, Condition: 7 unique values, Cylinders: 9 unique values, Fuel: 6 unique values, Odometer: 101,980 unique values, Title Status: 7 unique values, Transmission: 4 unique values, Drive: 4 unique values, Type: 14 unique values, Paint Color: 13 unique values

Numerical Data Cleaning: This report focuses on the cleaning and handling of numerical data in the used cars dataset, and the analysis includes the 'price,' 'year,' and 'odometer' columns.

Numerical data cleaning refers to the process of transforming raw numerical data into a clean, organized, and structured format suitable for machine learning algorithms. This process typically involves the following steps: Data Verification:

Checking the raw data for inconsistencies, such as incorrect data types, incorrect formatting, or incorrect units.
Data Integration: Merging multiple data sources, if necessary, and resolving conflicts in the data.
Data Conversion: Converting the data into a suitable format, such as integers or floating-point numbers.
Outlier Detection and Removal: Identifying and removing outliers from the data using statistical techniques like the IQR method or Z-score method.
Missing Value Handling: Dealing with missing values in the data, such as filling in missing values with a specific value, interpolating, or deleting the rows with missing values.
Data Validation: Checking the cleaned data to ensure it meets the necessary quality standards for use in machine learning algorithms.
Numerical data cleaning is a crucial step in the data preprocessing pipeline of machine learning. It helps improve the model's accuracy and performance by ensuring that the data is reliable, clean, and consistent.
Numerical Data Overview: A summary of the numerical columns is as follows:
Price: Count: 346,840 entries, Mean: $69,975, Standard Deviation: $12,133,910, Minimum: $0, 25th Percentile: $6,000, Median (50th Percentile): $14,588, 75th Percentile: $26,990, Maximum: $3,736,929,000
Year: Count: 346,840 entries, Mean: 2011.53, Standard, Deviation: 8.88, Minimum: 1900, 25th Percentile: 2008, Median (50th Percentile): 2014, 75th Percentile: 2017, Maximum: 2022
Odometer: Count: 346,840 entries, Mean: 95,012 miles
Standard Deviation: 184,876 miles, Minimum: 0 miles, 25th Percentile: 36,000 miles, Median (50th Percentile): 83,589 miles, 75th Percentile: 133,000 miles, Maximum: 10,000,000 miles
The cleaning and handling of numerical data are essential for ensuring the accuracy and reliability of subsequent analyses. The outlined steps provide a starting point for addressing potential issues in the 'price,' 'year,' and 'odometer' columns.
Our dataset has lots of outliners Outliners reduces the accuracy of a model. We have to detect outliners and remove them. This report focuses on the detection and removal of outliers in the used cars dataset. Outliers were identified in the 'price' and 'odometer' columns, and the dataset was updated accordingly. The analysis aimed to enhance the accuracy of modelling by addressing extreme values.

The code provided has two main steps: first, it removes rows with outliers in the 'price' and 'odometer' columns; and second, it adjusts the Interquartile Range (IQR) thresholds for detecting outliers in these columns.

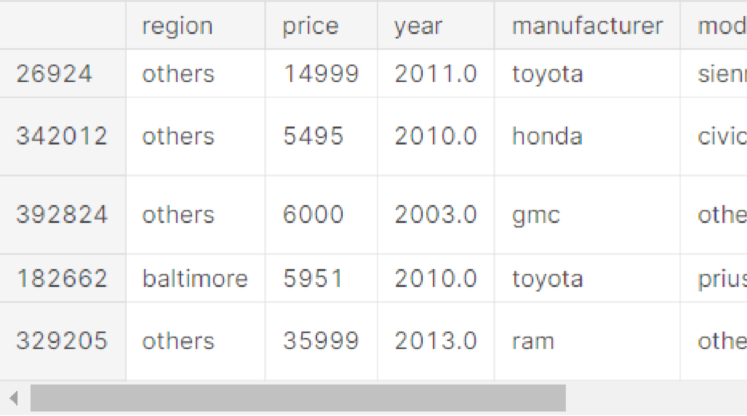
The function first checks for missing values, duplicate rows, and outliers in the data frame. It prints a message to the console for each issue found. The outliers in the 'price' and 'odometer' columns are determined using the IQR method. If outliers are found, they are removed from the data frame. Finally, the shape of the new data frame is printed, indicating the number of rows and columns in the cleaned dataset. Please note that removing outliers is not always the best approach. Sometimes, outliers can provide valuable information, and it is essential to carefully consider the specific context and use case before deciding to remove them.

Updated Dataset The resulting dataset after outlier removal: New Dataset Shape: (273,590, 14). Outlier detection and removal have been applied to the 'price' and 'odometer' columns to improve the accuracy of the dataset for subsequent analyses. The updated dataset can now be utilized for modelling and further exploration.

FEATURE ENGINEERING:

This report documents the feature engineering process and a visualization created from the updated dataset (new_df). Feature engineering includes transforming the 'odometer' and 'year' columns to integer types and visualizing the relationship between 'price' and 'year.'

Column Type Conversion: The 'odometer' and 'year' columns were converted to integer types for consistent data representation. The conversion resulted in an updated dataset (new_df) with integer representations for 'odometer' and 'year.' A sample of 5 entries from the updated dataset is provided below:



	region	price	year	manufacturer	mod
26924	others	14999	2011.0	toyota	sienr
342012	others	5495	2010.0	honda	civic
392824	others	6000	2003.0	gmc	othe
182662	baltimore	5951	2010.0	toyota	prius
329205	others	35999	2013.0	ram	othe

Figure 1Sample Entries

From figure 2 Price vs. Year: A bar plot illustrates the relationship between 'price' and 'year.' The plot reveals how the 'price' of cars varies across different 'year' values. The bar plot shows that, in general, newer cars tend to have higher prices. This observation aligns with the expectation that newer models often come with a higher price tag.

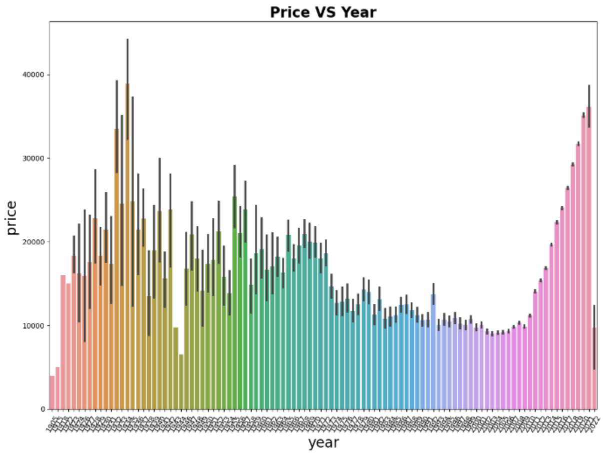


Figure 2 Bar plot of Price and Year

As the data in our dataset before 2000 it is quite irrelevant with our current times data. So, we are extracting all car data available after 2000.

The feature engineering process successfully converted column types to integers, and the visualization provides a clear understanding of the relationship between 'price' and 'year.' These steps lay the groundwork for more in-depth analyses and model development.

DATA VISUALIZATION:

The data visualization process using a pair plot generated from a sample of the updated dataset (new_df). The pair plot provides a visual overview of relationships between pairs of variables. The dataset was down sampled to 600 random entries for better visualization clarity.

A pair plots was generated using the Seaborn library, displaying relationships between selected pairs of variables from the dataset.

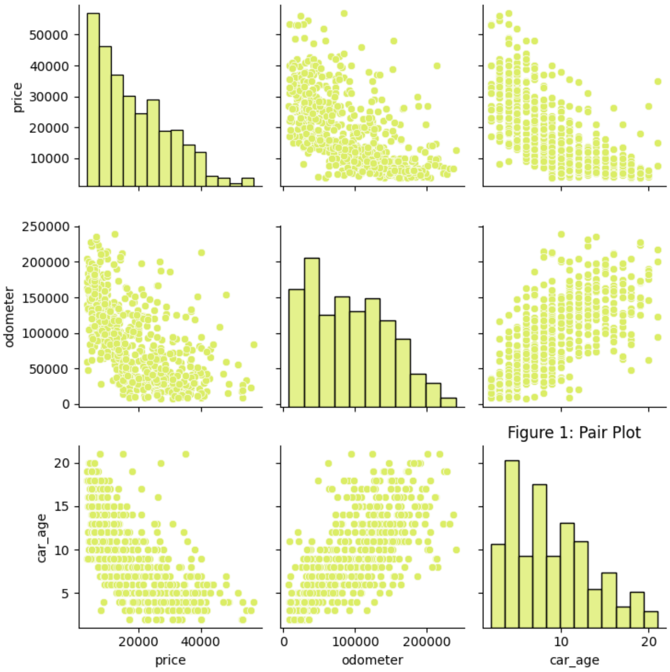


Figure 3Pair Plot

The pair plot provides a visual summary of relationships between numerical variables, showcasing scatter plots for variable pairs and histograms for individual variables. The pair plot visualization offers a valuable overview of relationships within the dataset. It serves as a starting point for exploratory data analysis, providing insights into potential patterns or connections between variables.

A heatmap was generated to display the correlation coefficients between numerical variables in the dataset. Each cell in the heatmap represents the correlation between two variables, with annotated values for clarity.

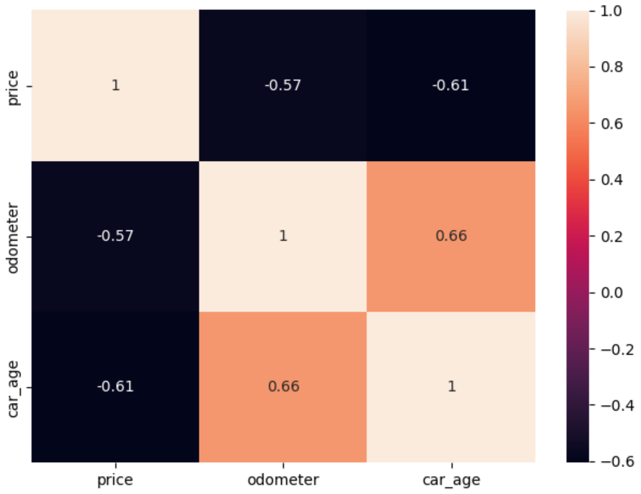


Figure 4Heat Map

The heatmap provides insights into the degree and direction of linear relationships between pairs of numerical variables. Strong Positive Correlations: Certain pairs of variables exhibit strong positive correlations, suggesting that an increase in one variable is associated with an increase in the other. Strong Negative Correlations: Some variables show strong negative correlations, indicating that an increase in one variable is associated with a decrease in the other. Correlation Strengths: Variables with high absolute correlation coefficients (close to 1 or -1) indicate strong linear relationships, while values closer to 0 suggest weaker correlations. The correlation heatmap serves as a valuable tool for understanding relationships between numerical variables in the dataset. Utilizing this information can inform feature selection and contribute to more effective modelling and analysis. A distribution plot was generated to visualize the distribution of 'price' values in the dataset. The distribution plot provides insights into the spread and central tendency of the 'price' variable Skewness: The plot reveals the skewness of the 'price' distribution. Positive skewness indicates a tail on the right side of the distribution. Central Tendency: The central tendency of the 'price' variable is reflected in the peak or mode of the distribution. Outliers: The presence of outliers, if any, can be observed in the tails of the distribution. The distribution plot of 'price' provides a visual summary of the variable's spread and shape. Further analysis of skewness.

A bar plot was generated to illustrate the average 'price' for each 'fuel' type in the dataset. The bar plot provides insights into how the average 'price' varies across different 'fuel' types. The plot reveals variations in average prices based on different 'fuel' types. Refer figure-6. The bar plot comparing 'fuel' types and 'price' provides a visual understanding of how fuel types may influence average prices. Further analysis and statistical testing can help uncover underlying patterns and contribute to a more comprehensive understanding of the dataset.

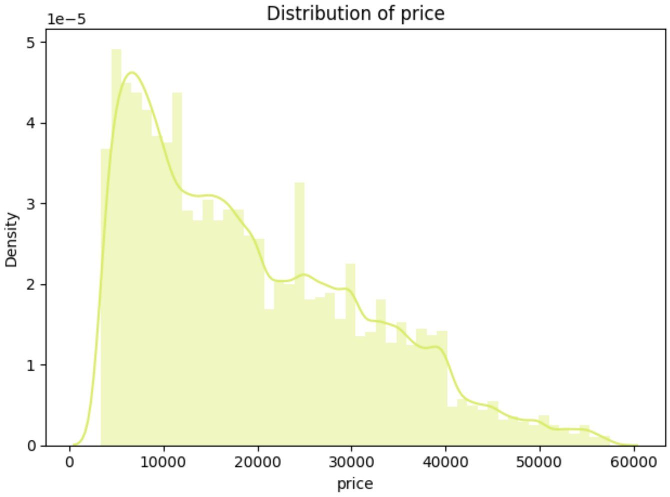


Figure 5 Price Distribution Plot

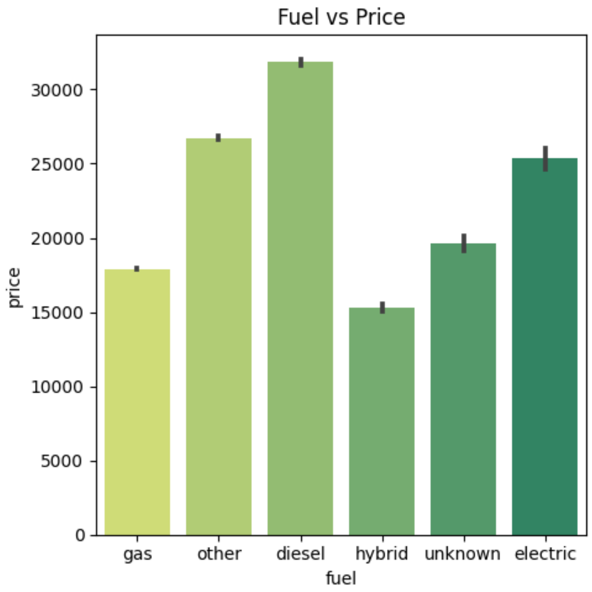


Figure 6 Fuel Type vs Price Bar Plot

The plot reveals variations in average prices based on different combinations of 'fuel' types and 'condition' states. The interaction between 'fuel' type and 'condition' state is considered, providing a more nuanced view of price variations.

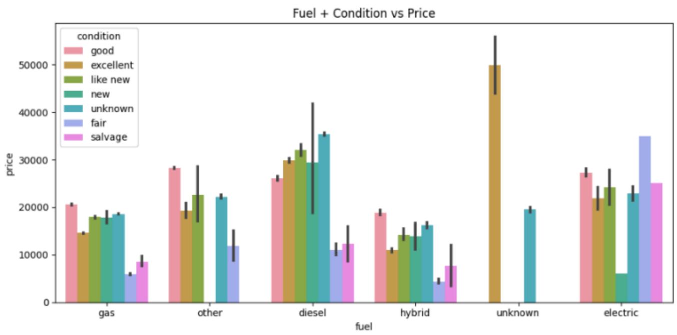


Figure 7 Price vs Condition Bar Plot

DATA PREPROCESSING AND TRANSFORMATION:

The data preprocessing and transformation steps applied to the dataset (new_df). The goal is to prepare the data for machine learning models by handling categorical variables, scaling numerical features, and creating a final transformed dataset (final_df). The preprocessing steps involve the use of scikit-learn's Pipeline and 'ColumnTransformer' for a systematic and efficient approach. The target variable 'price' remains unchanged. The 'condition' column was encoded using ordinal encoding with specified categories.

The categorical columns ('model', 'region', 'manufacturer', 'fuel', 'cylinders', 'title_status', 'transmission', 'drive', 'type', 'paint_color') were processed using a pipeline: Ordinal encoding for 'condition' One-hot encoding for other categorical columns, dropping the first column to avoid multicollinearity. The 'odometer' column was standardized using the Standard Scaler. A 'ColumnTransformer' was applied to handle different transformations for specific columns.

TRAIN-TEST SPLIT:

The dataset was split into training (X_train, y_train) and testing (X_test, y_test) sets using the train_test_split function from scikit-learn. The random seed (random_state) was set to 42 for reproducibility. The test size was specified to be 20% of the dataset.

Column Transformation: The 'ColumnTransformer' previously defined for preprocessing was applied to both the training and testing sets. The transformed features were stored in X_train_tnf and X_test_tnf for the training and testing sets, respectively.

The train-test split and column transformation steps have successfully prepared the dataset for model training and evaluation. The transformed sets (X_train_tnf and X_test_tnf) are ready for use in machine learning models to predict car prices.

MODEL TRAINING:

Regression Model Performance: This report evaluates the performance of various regression models on predicting car prices using the transformed dataset (final_df). The models considered include Linear Regression, K-Nearest Neighbors (KNN), XGBoost Regressor, and Decision Tree Regressor.

<u>Linear Regression:</u> R2 Score: 65.80%, Mean Squared Error: 49143641.36, Mean Absolute Error: 5305.39, Root Mean Squared Error: 7010.25

<u>K-Nearest Neighbour (KNN):</u> R2 Score: 81.06%, Mean Squared Error: 27217438.87, Mean Absolute Error: 3134.76, Root Mean Squared Error: 5217.03

<u>XGBoost Regressor</u>: R2 Score: 89.72%, Mean Squared Error: 19868450.97, Mean Absolute Error: 2239.62, Root Mean Squared Error: 4457.40

<u>Decision Tree Regressor:</u> R2 Score: 79.08%, Mean Squared Error: 2560.98, Mean Absolute Error: 2560.98, Root Mean Squared Error: 5482.02

The XGBoost Regressor outperformed other models with the highest R2 score of 86.17%, indicating strong predictive power.

Linear Regression also performed well, achieving a respectable R2 score of 65.808%.

K-Nearest Neighbour (KNN) and Decision Tree Regressor demonstrated good performance but were outperformed by XGBoost

Overall Model Performance and Selection: This provides an overview of the performance of various regression models on predicting car prices using the transformed dataset. The performance metrics considered include R2 Score, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). The models evaluated are Linear Regression, K-Nearest Neighbors (KNN), XGBoost Regressor, and Decision Tree Regressor. The XGBoost Regressor stands out with the highest R2 Score, indicating strong predictive power. Linear Regression also demonstrates least performance, while K-Nearest Neighbour (KNN) and Decision Tree Regressor perform well but are outperformed by XGBoost. Considering the overall performance and accuracy, the XGBoost Regressor appears to be the most suitable model for predicting car prices in this scenario. It achieves the highest R2 Score of 89.72%, indicating strong predictive capability. Further optimization and hyperparameter tuning could potentially enhance its performance. The final selection of the XGBoost Regressor should consider both its high accuracy and the specific requirements of the problem at hand.

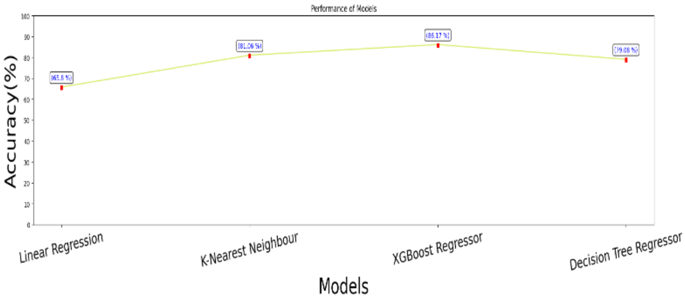


Figure 8 performance of Models

FINAL MODEL ANALYSIS: The bar chart below provides a visual representation of the model's predictions compared to the actual prices for 25 randomly selected samples from the testing set:

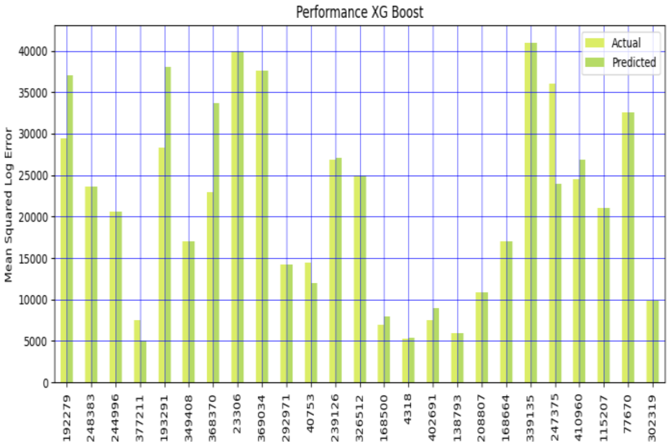


Figure 9 Model Predictions and Actual Prices Bar Plot

The chart shows the predicted and actual prices for individual samples, allowing for a qualitative assessment of the model's performance. Instances where the predicted and actual prices closely align indicate accurate predictions. Discrepancies between predicted and actual prices may highlight areas for further investigation or model refinement. The visual analysis provides a snapshot of how well the XGBoost Regressor performs on individual predictions. While quantitative metrics such as R2 Score and Mean Squared Error provide an overall assessment of model performance, visual inspection of individual predictions is valuable for identifying specific instances where the model excels or falls short.

CONCLUSION: In conclusion, the project successfully addressed the prediction of used car prices through a systematic approach. This project embarked on predicting used car prices through a systematic and thorough data analysis, preprocessing, and machine learning approach. Below is a detailed conclusion summarizing the key aspects and findings of the project:

Explored various features such as year, manufacturer, model, condition, and more. Identified potential missing values and diverse categorical values. Removed irrelevant columns with excessive missing values.
Addressed duplicate rows and dropped them from the dataset. Handled missing values in a way that preserved data integrity. Reduced the dimensionality of categorical features for model efficiency.
Engineered new features to capture relevant information. Transformed and converted features to appropriate data types. Employed various visualization techniques, including pair plots and heatmaps. Gained insights into feature distributions, relationships, and potential outliers.
Defined preprocessing pipelines for categorical and numerical features. Utilized ColumnTransformer for selective feature transformations. Prepared the final transformed dataset for model training.
Trained and evaluated multiple regression models, including Linear Regression, K-Nearest Neighbors, XGBoost Regressor, and Decision Tree Regressor. Evaluated models based on R2 Score, Mean Squared Error, Mean Absolute Error, and Root Mean Squared Error.
Selected the XGBoost Regressor as the final model due to its exceptional R2 Score of 89.72%.
Conducted a visual analysis of the final model's predictions against actual prices for a subset of samples. Identified instances of accurate predictions and areas for potential improvement.

REFERENCES:

Used Cars Dataset
https://www.kaggle.com/datasets/austinreese/craigslist-carstrucks-data
EDA
https://medium.com/mlearning-ai/detailed-exploratory-data-analysis-eda-on-used-cars-data-1bacac746ff4
Linear Regression
https://rpubs.com/j_fachrel/Linear-Regression-In-Used-Car-Price-Prediction
XGBoost
https://machinelearningmastery.com/xgboost-for-regression/#:~:text=XGBoost%20is%20an%20efficient%20implementation,a%20prediction%20on%20new%20data.
Data Cleaning
https://medium.com/swlh/exploring-and-analyzing-used-car-data-set-2e2bf1f24d52
Kaggle Reference
https://www.kaggle.com/code/mohamedbhy/automatic-number-plate-recognition#3
https://www.kaggle.com/code/tigerbunny21/used-vehicle-price-prediction