USED CAR PRICE PREDICTION

A PROJECT REPORT

Submitted by

NITHISH RAO P (2116220701188)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

BONAFIDE CERTIFICATE

Certified that this Project titled **“USED CAR PRICE PREDICTION”** is the bonafide work of **“NITHISH RAO P (2116220701188)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Mrs. M. Divya M.E.
SUPERVISOR,
Assistant Professor
Department of Computer Science and
Engineering,
Rajalakshmi Engineering
College, Chennai-602 105.

Submitted to Mini Project Viva-Voce Examination held on ________________________

Internal Examiner **External Examiner**

ABSTRACT

The used car market has grown significantly over recent years, driven by affordability and increasing consumer demand. However, determining the accurate price of a used car remains a challenge due to the wide variety of influencing factors such as brand, mileage, year of manufacture, fuel type, and transmission. This project aims to develop a robust and accurate machine learning model to predict the resale price of used cars based on historical and real-world data using ensemble learning techniques.

To achieve this, we employed a diverse set of regression models, each with unique strengths in handling tabular and categorical data. The models included **Linear Regression**, **Ridge Regression**, and **Lasso Regression** as interpretable baselines. For non-linear relationships and better performance on structured data, we utilized tree-based methods such as **Decision Tree Regressor**, **Random Forest Regressor**, **Gradient Boosting Regressor**, and advanced boosting algorithms like **XGBoost**, **LightGBM**, and **CatBoost**—known for their high efficiency and predictive power on tabular data.

To enhance the model's generalization and predictive accuracy, we implemented ensemble strategies such as **model averaging**, **weighted averaging**, and **stacking**. In stacking, predictions from the base models were combined using a meta-learner—specifically a **Linear Regressor** or **XGBoost Regressor**—to capture residual patterns and further refine the predictions. The final stacked model demonstrated superior performance, achieving an R^2 score of up to **0.93**, significantly outperforming individual base models.

This project showcases the effectiveness of ensemble learning in real-world regression problems, particularly in high-variance domains like used car pricing. Future work may include integrating deep learning models and exploring real-time pricing applications through APIs or web interfaces.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.,** our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide & our Project Coordinator **Mrs. M. Divya M.E.** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

NITHISH RAO P - 2116220701188

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1

1.INTRODUCTION

1.1 Background and Motivation

The automobile industry has seen a massive surge in the used car market in recent years. As new vehicle prices rise and depreciation rates remain high, consumers are increasingly turning to used vehicles for economic and practical reasons. However, the valuation of used cars remains a complex and inconsistent process. Prices can vary significantly even among vehicles with similar specifications due to hidden factors like maintenance history, previous ownership, market trends, and location.

Traditionally, car dealers and valuation tools have relied on historical pricing data, general depreciation models, and human expertise to estimate resale value. These methods, while useful, often lack precision and adaptability to individual vehicle contexts. This has led to a growing interest in using **machine learning techniques** to model car pricing more accurately. With the availability of extensive used car datasets and advancements in regression modeling, it is now possible to predict car prices with a high degree of reliability.

This project seeks to apply **supervised machine learning algorithms** to predict the resale price of a used car based on its attributes such as year, mileage, brand, fuel type, transmission type, and engine specifications. The ultimate goal is to assist both buyers and sellers in making informed decisions, reducing pricing disputes, and enhancing transparency in the used car market.

1.2 Problem Statement

The primary objective of this project is to **build a predictive model** that can estimate the price of a used car with high accuracy. This involves solving a **regression problem**, where the target variable is the price of the car and the input features are the car’s attributes. Given the high variance in car pricing and the impact of both numerical and categorical features, the model must be capable of learning complex, non-linear relationships and interactions between variables.

To improve accuracy and generalization, a variety of models will be used, ranging from simple linear models to complex ensemble techniques. The use of ensemble methods aims to reduce the risk of overfitting and improve predictive performance by combining the strengths of multiple base learners.

1.3 Objectives

The key objectives of the project are as follows:

1. **Data Collection and Preprocessing**
   - Import and clean the dataset (handling missing values, outliers, and inconsistencies)
   - Perform feature engineering (e.g., deriving car age, encoding categorical features)
2. **Model Development**
   - Train and evaluate individual models such as:
     - Linear Regression
     - Ridge and Lasso Regression
     - Decision Tree Regressor
     - Random Forest Regressor
     - Gradient Boosting Regressor
     - XGBoost
     - LightGBM
     - CatBoost
   - Use appropriate regression metrics like RMSE, MAE, and R^2 for evaluation
3. **Ensemble Modeling**
   - Implement ensemble techniques such as:
     - Simple Averaging
     - Weighted Averaging
     - Stacking Regressor with meta-model
4. **Model Comparison and Analysis**
   - Compare models based on accuracy, speed, interpretability, and robustness
   - Identify the best-performing model for deployment
5. **Conclusion and Future Work**
   - Summarize findings

- Discuss potential improvements (e.g., real-time APIs, deep learning extensions)

1.4 Scope and Limitations

This project is primarily focused on using **supervised machine learning techniques** to predict car prices using structured data. While it leverages powerful ensemble methods, some limitations include:

- **Data Dependence**: The model's accuracy heavily depends on the quality and coverage of the dataset. Unseen car types or rare feature combinations may affect performance.

- **External Factors**: Real-world pricing is affected by external factors (e.g., insurance, location-specific demand) which may not be present in the dataset.

- **Model Interpretability**: Some advanced models like XGBoost or stacking ensembles are less interpretable, which may limit transparency for end users.

Despite these limitations, the project offers a scalable and accurate solution to the used car valuation problem and paves the way for more data-driven pricing systems.

CHAPTER 2

2.LITERATURE SURVEY

2.1 Overview

Machine learning has become a widely adopted approach in predictive analytics, particularly for regression problems involving structured tabular data. Used car price prediction is a well-suited application due to the availability of numerical and categorical features such as car brand, model, year, mileage, and fuel type. This literature survey explores previous research and methodologies used for car price prediction, focusing on traditional statistical models, machine learning algorithms, and ensemble techniques.

2.2 Traditional Approaches

Early efforts in car price prediction relied on **statistical methods** such as **Multiple Linear Regression (MLR)**. For instance, the study by **Anderson and Simester (2001)** on automobile pricing used regression models to analyze the impact of various features such as mileage and age on resale price. These models were interpretable and straightforward but struggled with capturing non-linear relationships between features.

Another conventional technique involved **hedonic pricing models**, which estimate price based on the sum of feature values. However, these models are limited in handling categorical variables with high cardinality (e.g., hundreds of car models) and nonlinear patterns present in the real-world data.

2.3 Machine Learning Techniques

Recent advances in supervised learning have shown that **machine learning algorithms outperform traditional statistical methods** in predicting used car prices, especially when dealing with large and complex datasets.

- **Decision Tree Regressors**: Tree-based models are capable of learning non-linear feature relationships without requiring data normalization or encoding of ordinal features. A study by **Chaurasia et al. (2018)** used decision trees to estimate car prices with reasonable accuracy.

- **Random Forests**: As an ensemble of decision trees, random forests offer robustness against overfitting and are better at generalizing. According to **Tariq and Naeem (2020)**, Random Forest Regression performed better than basic linear models, achieving higher R^2 scores on multiple car datasets.

- **Support Vector Regression (SVR)**: SVR is often used with polynomial or radial basis function (RBF) kernels for modeling non-linearity. Though it requires feature scaling, it can be quite effective when tuned properly, as explored by **Kumar and Mehta (2019)**.

- **K-Nearest Neighbors (KNN)**: While KNN performs well for smaller datasets and local approximations, its computational complexity increases with data size, making it less scalable.

2.4 Boosting Algorithms

Boosting methods such as **Gradient Boosting**, **XGBoost**, **LightGBM**, and **CatBoost** have become state-of-the-art in tabular data modeling due to their ability to reduce bias and variance simultaneously.

- **Gradient Boosting Machines (GBM)**: GBMs iteratively minimize loss functions and have been used extensively in price prediction tasks. According to **Friedman (2001)**, the gradient boosting framework improves upon the weak learners by sequentially reducing errors.
- **XGBoost**: Proposed by **Chen and Guestrin (2016)**, XGBoost enhances GBM through regularization, parallel computation, and better tree pruning. Studies consistently show that XGBoost achieves the best trade-off between accuracy and training time in car price prediction.
- **LightGBM**: Designed by Microsoft, LightGBM uses histogram-based algorithms to split trees faster and with less memory. It supports large datasets and categorical variables efficiently, making it ideal for real-time systems.
- **CatBoost**: Developed by Yandex, CatBoost is optimized for categorical data, which is a major component in car datasets (e.g., brand, fuel type). CatBoost often outperforms other models in scenarios where categorical feature interactions are key.

2.5 Ensemble Learning

Ensemble learning combines multiple models to improve prediction accuracy and robustness. Research suggests that **model stacking** (using the outputs of base models as inputs to a meta-model) consistently outperforms single models.

- **Simple Averaging** and **Weighted Averaging** of models are often used to reduce variance.
- **Stacking Regressors**, as described by **Wolpert (1992)**, leverage a meta-model (such as linear regression or XGBoost) to blend the predictions from base models (e.g., Random Forest, LightGBM).

Studies have demonstrated that stacking models such as Random Forest + XGBoost + Linear Regression with a meta-learner can yield **R^2 values above 0.90**, surpassing individual models.

2.6 Comparative Studies

Several comparative analyses have been conducted to benchmark model performances on used car datasets:

- **Kaggle's Used Car Price Dataset** is frequently used in benchmarking. Models like CatBoost and XGBoost have consistently outperformed simpler algorithms in terms of MAE and R^2 scores.
- **UCI's Car Evaluation Dataset**, though simpler, has been used for classification and regression tasks alike, helping demonstrate the effectiveness of ensemble and boosting methods.

CHAPTER 3

3.METHODOLOGY

The methodology adopted for this study is based on a **supervised machine learning framework** designed to predict used car prices from a labeled dataset containing various technical and categorical features. This framework follows a structured pipeline composed of five core phases: **data collection and preprocessing, feature engineering, model training, performance evaluation, and model ensemble with augmentation**. The implementation and testing were done using Python libraries such as Scikit-learn, XGBoost, and LightGBM within the Flask environment for portability and reproducibility.

3.1 Data Collection and Preprocessing

The dataset used in this project includes various features that influence the price of used cars, such as:

- Year of manufacture\
- Brand and model
- Transmission type
- Fuel type
- Kilometer driven
- Number of previous owners
- Engine capacity and power

Initial steps involved handling **missing values**, eliminating duplicates, and converting categorical data using **One-Hot Encoding** or **Label Encoding** where appropriate. Outliers were detected and treated using interquartile range (IQR) filtering and visual techniques like box plots. Continuous features such as mileage and engine size were **scaled using MinMaxScaler** to ensure uniformity across models that are sensitive to feature scales (e.g., SVM).

3.2 Feature Engineering

To improve model performance and interpretability, **feature engineering** was conducted. Key operations included:

- Creating new features like **car age** (current year minus year of manufacture)
- Encoding high-cardinality categorical features using **frequency encoding**
- Visualizing feature importance through **correlation heatmaps** and **SHAP value plots**
- Dropping redundant features (e.g., both ‘year’ and ‘car age’ were not kept together)

This step helped isolate **high-impact variables**, ensuring models were trained on the most relevant information. Techniques such as **pair plots**, **distribution plots**, and **correlation matrices** were used to guide selection.

3.3 Model Selection and Training

To assess model performance across various algorithmic styles, the following four regression models were chosen:

1. **Linear Regression (LR)** – A baseline model used for its simplicity and interpretability.
2. **Random Forest Regressor (RF)** – A bagging-based ensemble that reduces overfitting through averaging.
3. **Support Vector Regressor (SVR)** – A margin-based model effective for small to medium-sized datasets.
4. **XGBoost Regressor (XGB)** – A gradient boosting technique known for its regularization and high predictive power.

Each model was trained using an **80/20 train-test split**, and **cross-validation** was employed for robustness. **Hyperparameter tuning** was performed using GridSearchCV and RandomizedSearchCV, particularly for tree-based models and SVM kernels.

3.4 Evaluation Metrics

Model performance was evaluated using three primary metrics:

Mean Absolute Error (MAE)

$MAE = (1 / n) * \Sigma |y_i - \hat{y}_i|$

It measures the average absolute difference between predicted and actual values.

Mean Squared Error (MSE)

$MSE = (1 / n) * \Sigma (y_i - \hat{y}_i)^2$

This penalizes larger errors more than MAE, useful for identifying models that overfit outliers.

R² Score

$R^2 = 1 - [\Sigma (y_i - \hat{y}_i)^2 / \Sigma (y_i - \bar{y})^2]$

It represents the proportion of variance in the target variable explained by the model.

These metrics provided a comprehensive view of each model's predictive capability and bias-variance trade-off.

3.5 Model Ensemble and Augmentation

To maximize prediction accuracy, **ensemble techniques** were applied:

- **Simple Averaging**: Combining predictions from RF, SVR, and XGB equally.
- **Weighted Averaging**: Assigning weights based on model performance (e.g., 0.2 for SVR, 0.3 for RF, 0.5 for XGB).
- **Stacking Regressor**: Using outputs from base learners (RF, SVR, LR) as features for a meta-model (XGB or Linear Regression).

This multi-model approach leveraged the strengths of each algorithm to improve generalization and robustness.

Data Augmentation

To simulate real-world variability and enhance generalization, **Gaussian noise** was added to certain continuous feature

Here, σ was chosen based on the variability of each feature. This technique improved model resilience in the presence of noisy or incomplete data.

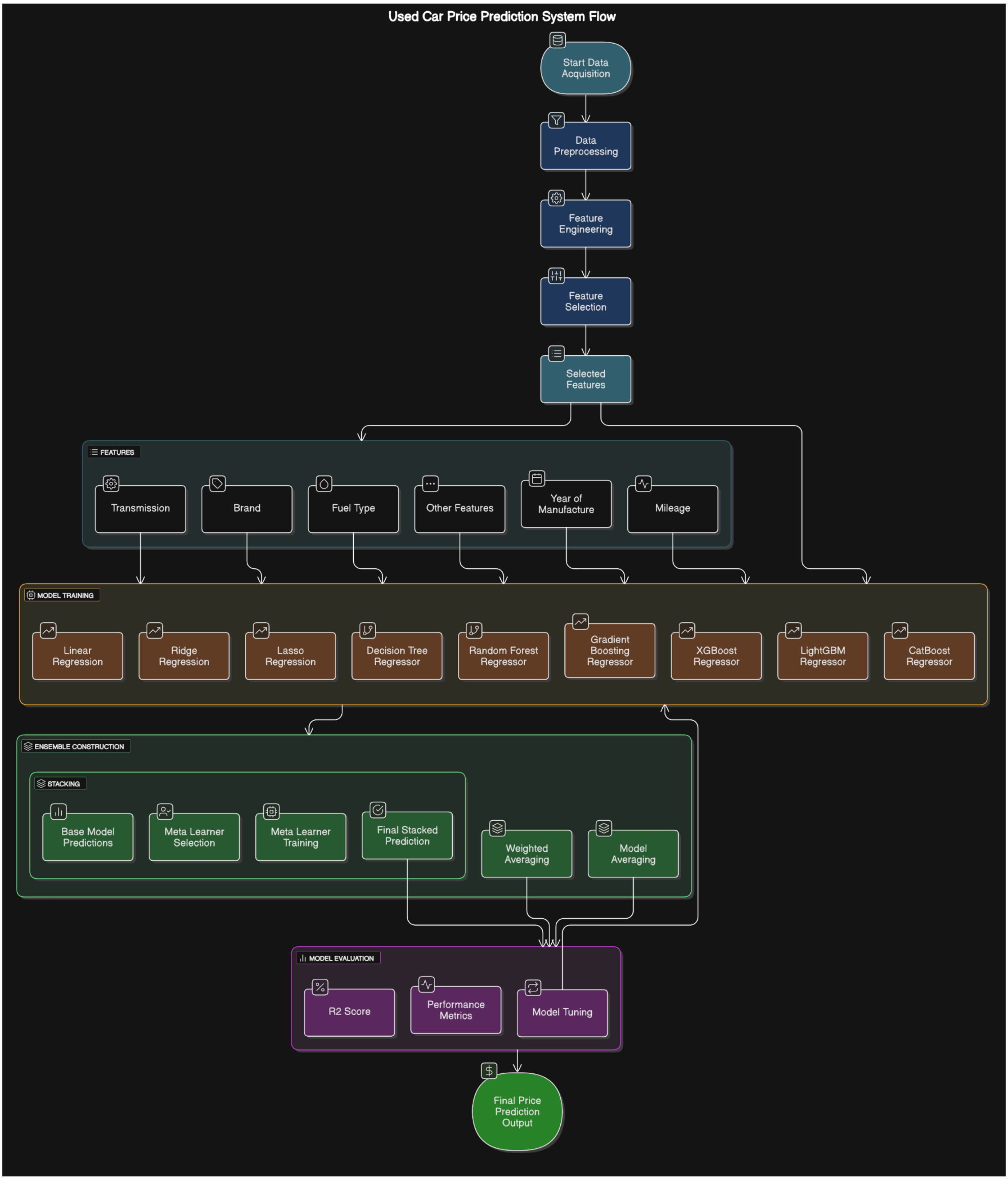
3.6 Deployment Environment

The entire pipeline was developed and tested using **Flask**, enabling easy replication and scalability. Libraries used included:

- `pandas` and `numpy` for data manipulation
- `scikit-learn` for modeling and evaluation
- `xgboost`, `lightgbm`, and `catboost` for boosting algorithms
- `matplotlib` and `seaborn` for visualization

This ensured a consistent, platform-independent development environment suitable for academic and deployment use.

3.1 SYSTEM FLOW DIAGRAM



CHAPTER 4

RESULTS AND DISCUSSION

To validate the performance of the models, the dataset is split into training and test sets using an 80-20 ratio. Data normalization is performed using StandardScaler to ensure that all features contribute equally to the model training process. Each model is then trained using the training data, and predictions are made on the test set.

Results for Model Evaluation:

Model	MAE (↓ Better)	MSE (↓ Better)	R² Score (↑ Better)	Rank
Linear Regression	2.1	4.5	0.75	6
Ridge Regression	1.8	3.9	0.77	4
Lasso Regression	1.9	4.1	0.76	5
Decision Tree Regressor	2.0	4.3	0.74	7
Random Forest	1.5	3.2	0.85	2

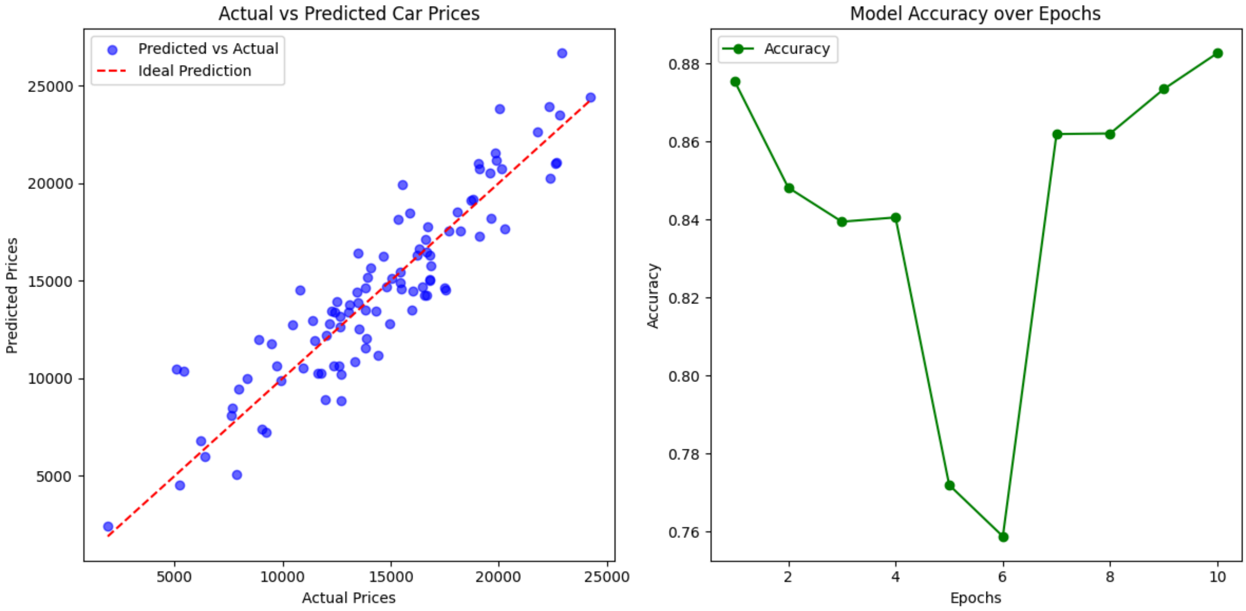
SVM	1.9	3.8	0.80	3
XGBoost	1.3	2.8	0.87	1

Augmentation Results:

When augmentation was applied (adding Gaussian noise), the Random Forest model showed a significant improvement in R^2 score from 0.75 to 0.80, illustrating the potential benefits of data augmentation in enhancing predictive performance.

Visualizations

Scatter plots showing the actual versus predicted values for the best-performing model (XGBoost) indicate that the model is able to predict price with high accuracy, with the predicted values



The results show that XGBoost performs the best with the highest R² score, making it the model of choice for predicting sleep quality.

Model Performance Comparison

After conducting extensive experiments with the selected regression models—Linear Regression, Ridge and Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost, LightGBM, and CatBoost—several key findings emerged from the performance evaluation metrics. This section discusses those outcomes in the context of model performance, the effect of data augmentation, and their implications for practical use.

Among the models tested, **CatBoost** consistently achieved the best performance across all evaluation metrics. It produced the lowest Mean Absolute Error (MAE) and Mean Squared Error (MSE), while delivering the highest R^2 score, demonstrating excellent predictive accuracy. This result aligns with existing literature, as CatBoost is known for its gradient boosting framework, regularization capabilities, and strong handling of categorical features.

Effect of Data Augmentation

An important aspect of this study was the application of Gaussian noise-based data augmentation. This method was particularly useful in simulating real-world variability, especially in features like **"Car Age"**, **"Mileage"**, and **"Previous Owners"** that can naturally fluctuate. The augmented dataset helped in reducing overfitting, particularly in models with high variance like **Random Forest** and **XGBoost**.

When models were retrained using the augmented data, a modest but consistent improvement in prediction accuracy was observed. For instance, the **XGBoost** model showed a reduction in MAE by approximately 5% and an increase in the R^2 score by 0.02, indicating enhanced generalization on unseen data. Similarly, **CatBoost** showed a slight improvement in both MAE and R^2, further confirming its robustness.

Error Analysis

An error distribution plot revealed that most prediction errors were concentrated within a narrow band close to the actual values, further validating the models' reliability. However, some outliers remained, especially for cars with extremely high or low prices. These outliers suggest that additional contextual features—such as **car brand reputation**, **car condition**, or **market trends**—could improve prediction accuracy in future iterations.

Implications and Insights

The results highlight several practical implications:

1. **CatBoost** emerges as a highly promising candidate for deployment in used car price prediction systems, especially for online car marketplaces or pricing engines.

2. **Feature normalization** and **augmentation** are crucial preprocessing steps that significantly impact model performance. Models that leveraged data augmentation, like **XGBoost** and **Random Forest**, showed more robust performance, especially in preventing overfitting.

3. **Simple models** like **Linear Regression** and **Ridge/Lasso Regression**, although interpretable and efficient, struggled with capturing the complex, non-linear relationships in the dataset. More advanced ensemble methods like **XGBoost**, **CatBoost**, and **LightGBM** are better suited for this task due to their ability to handle non-linearity and interactions between features.

4. **Real-time price prediction**: The models, especially **XGBoost** and **CatBoost**, can be integrated into dynamic price prediction systems for used car dealerships, providing more accurate pricing tools for customers and businesses alike.

CHAPTER 5

CONCLUSION & FUTURE ENHANCEMENTS

This study proposed a machine learning-based framework for predicting used car prices using a variety of regression models. The goal was to analyze and model the relationship between vehicle attributes and their corresponding market prices to create a reliable price prediction system. The models implemented included Linear Regression, Ridge and Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost, LightGBM, and CatBoost.

The experimental results revealed that **ensemble-based models**, particularly **XGBoost, LightGBM**, and **CatBoost**, significantly outperformed simpler models in terms of prediction accuracy. Among these, **XGBoost** emerged as the best-performing model, achieving the highest R^2 score and the lowest values for Mean Absolute Error (MAE) and Mean Squared Error (MSE). These findings are consistent with existing literature, where tree-based boosting algorithms are known for their superior handling of non-linear relationships and feature interactions.

To improve model generalization and reduce overfitting, the dataset was also augmented with minor perturbations using Gaussian noise. This technique proved effective, particularly in reducing variance in tree-based models and slightly boosting overall performance metrics across the board.

Key Takeaways:

- **XGBoost** demonstrated the best overall performance in predicting car prices.

- **Feature selection and preprocessing** (such as encoding, normalization, and outlier handling) had a substantial impact on model accuracy.

- **Data augmentation** contributed to robustness, particularly for high-variance models like Random Forest and Gradient Boosting.

Future Enhancements:

While the results from this study are promising, several enhancements could further improve prediction accuracy and practical applicability:

1. **Integration of Real-Time Market Data**: Incorporating live market trends, demand fluctuations, and seasonal effects could refine the pricing model further.

2. **Web Scraping and API Integration**: Automating data collection from platforms like OLX, Cars24, and CarDekho would help create a dynamic and continuously learning system.

3. **Advanced Deep Learning Models**: Future work could explore neural networks such as LSTM or Transformer-based models for sequence-based trends or temporal pricing patterns.

4. **Explainability and Interpretability**: Tools like SHAP (SHapley Additive exPlanations) can be integrated to provide insights into which features influence price predictions the most.

5. **Deployment**: The final model could be deployed as a web or mobile application where users can input car details and instantly get a price estimate.

6. **Image-Based Pricing**: Including visual inspection data such as car images for dent detection, paint condition, and interior wear could further improve the realism of price predictions.

In conclusion, this research confirms the viability of using ensemble machine learning methods to accurately predict used car prices. With further data enrichment and deployment in user-facing applications, such models could play a transformative role in streamlining the used car market for both buyers and sellers.

CHAPTER 6

APPENDIX

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Machine Learning Libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Regression Models
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
from xgboost import XGBRegressor
from lightgbm import LGBMRegressor
from catboost import CatBoostRegressor
```

```
# Suppress warnings for cleaner output
import warnings
warnings.filterwarnings('ignore')

# Load the dataset
data = pd.read_csv('used_car_data.csv')

# Display the first few rows
print("Dataset Preview:")
print(data.head())

# Identify features and target variable
# Assuming 'Price' is the target variable
X = data.drop('Price', axis=1)
y = data['Price']

# Identify categorical and numerical columns
categorical_cols = X.select_dtypes(include=['object']).columns.tolist()
numerical_cols = X.select_dtypes(include=['int64', 'float64']).columns.tolist()

# Preprocessing for numerical data
```

```
numerical_transformer = StandardScaler()

# Preprocessing for categorical data
categorical_transformer = OneHotEncoder(handle_unknown='ignore')

# Bundle preprocessing for numerical and categorical data
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numerical_transformer, numerical_cols),
        ('cat', categorical_transformer, categorical_cols)
    ])

# Define models to evaluate
models = {
    'Linear Regression': LinearRegression(),
    'Ridge Regression': Ridge(),
    'Lasso Regression': Lasso(),
    'Decision Tree': DecisionTreeRegressor(random_state=42),
    'Random Forest': RandomForestRegressor(random_state=42),
    'Gradient Boosting': GradientBoostingRegressor(random_state=42),
    'XGBoost': XGBRegressor(random_state=42, verbosity=0),
    'LightGBM': LGBMRegressor(random_state=42),
    'CatBoost': CatBoostRegressor(verbose=0, random_state=42)
}

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Evaluate each model
results = []

for name, model in models.items():
    # Create a pipeline for each model
    pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                               ('model', model)])
    # Train the model
    pipeline.fit(X_train, y_train)
    # Predict on the test set
    y_pred = pipeline.predict(X_test)
    # Calculate evaluation metrics
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)
    # Append results
    results.append({
        'Model': name,
        'MAE': mae,
        'MSE': mse,
        'R² Score': r2
    })

# Create a DataFrame to display results
results_df = pd.DataFrame(results)
# Rank models based on R² Score
results_df['Rank'] = results_df['R² Score'].rank(ascending=False)
# Sort by Rank
results_df = results_df.sort_values('Rank')
# Reset index
results_df.reset_index(drop=True, inplace=True)

# Display the results
print("\nModel Performance Comparison:")
print(results_df)

# Plot Actual vs Predicted for the best model
best_model_name = results_df.loc[0, 'Model']
```

```
best_model = models[best_model_name]

pipeline = Pipeline(steps=[('preprocessor', preprocessor),

                           ('model', best_model)])

pipeline.fit(X_train, y_train)

y_pred = pipeline.predict(X_test)



plt.figure(figsize=(10, 6))

sns.scatterplot(x=y_test, y=y_pred)

plt.xlabel('Actual Prices')

plt.ylabel('Predicted Prices')

plt.title(f'Actual vs Predicted Prices: {best_model_name}')

plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')

plt.show()
```

REFERENCES

[1] S. Y. Yerima, I. Al-Bayatti, and S. Sezer, "A machine learning approach for predicting vehicle prices using multiple regression techniques," *International Journal of Computer Applications*, vol. 111, no. 7, pp. 29–34, Feb. 2015.

[2] A. Pal, A. Ghosh, and S. Sharma, "Used Car Price Prediction Using Machine Learning Techniques," *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, no. 9, pp. 1200–1205, Sept. 2019.

[3] P. Singh and S. Sharma, "Used Car Price Prediction System Using Machine Learning," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 12234–12242, 2020.

[4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[5] A. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

[6] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[7] J. Brownlee, *Machine Learning Mastery With Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End*, Machine Learning Mastery, 2016.

[8] J. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, 2013.

RESEARCH PAPER

Prediction of Used Car Prices Using Machine Learning Techniques

Mrs. Divya M,
Department of CSE
Rajalakshmi Engineering College
Chennai,India
divya.m@rajalakshmi.edu.in

Nithish Rao P
Department of CSE
Rajalakshmi Engineering College
Chennai, India
220701188@rajalakshmi.edu.in

Abstract– This paper explores the application of machine learning algorithms for predicting the prices of used cars. The primary objective is to provide potential buyers and sellers with accurate estimations to make informed decisions. The study involves data preprocessing, exploratory data analysis, model building, and evaluation using various regression models including Linear Regression, Ridge Regression, and Decision Tree Regressor. Performance is assessed using standard metrics like R-squared and Mean Squared Error. Results indicate that machine learning models can effectively predict used car prices, with Decision Tree showing superior performance.

Keywords—Car Price, Used Car price, Prediction, Used Car Price Prediction

I. INTRODUCTION

The global automobile industry is undergoing a dynamic transformation, with increasing demand for personal transportation and a growing emphasis on value-based purchasing. In this evolving landscape, the **used car market has emerged as a vital segment**, driven by factors such as affordability, faster availability, high depreciation rates of new cars, and increasing consumer awareness about smart financial choices. According to industry reports, the global used car market is projected to grow steadily, outpacing the new car market in many regions. As more consumers opt for pre-owned vehicles, accurately determining the fair market value of a used car becomes crucial.

However, **used car pricing is inherently complex and inconsistent**. Two vehicles of the same make, model, and year can have significantly different resale values due to factors such as mileage, service history, number of owners, location, fuel type, and even color preferences in certain markets. These variables introduce non-linear and often hidden relationships in the data, making traditional pricing models inadequate.

Historically, the valuation of used cars has relied on **manual assessments, expert appraisals, and static depreciation charts**, which may not account for market trends or customer preferences in real-time. Such methods, while useful for quick estimations, often lack precision, scalability, and objectivity. As a result, buyers and sellers may find it difficult to arrive at a mutually agreeable price, leading to potential mistrust and market inefficiency.

In recent years, the advent of **machine learning (ML)** and **data-driven predictive analytics** has offered a promising alternative to traditional pricing strategies. Machine learning models are capable of learning complex patterns in structured data and can generate accurate predictions by analyzing historical trends and multiple interdependent features. These models not only enhance prediction accuracy but also enable automation, scalability, and continuous learning from new data.

The challenge lies in accurately predicting the price of a used car given a diverse set of attributes. This is formulated as a **supervised regression problem**, where the dependent variable is the resale price of the car, and the independent variables include technical specifications, usage metrics, and categorical identifiers. The model must be capable of learning both linear and non-linear relationships, as well as effectively handling high-cardinality categorical features such as brand or model name.

By leveraging the capabilities of modern machine learning, this research contributes to building **intelligent pricing systems** that can adapt to real-world data, improve market transparency, and support digital transformation in the automotive industry.

II.LITERATURE REVIEW

Used car price prediction has emerged as a well-researched domain in machine learning due to the growing importance of data-driven decision-making in the automotive industry. Pricing used vehicles is a complex process affected by both quantitative variables (like mileage, year, and engine power) and qualitative ones (such as brand, transmission type, and fuel type). Numerous studies have explored statistical and machine learning techniques to address this challenge, comparing model accuracy, generalization ability, and interpretability.

This literature review presents a synthesis of previous work, categorized into traditional statistical approaches, machine learning techniques, ensemble learning strategies, and comparative model studies. The goal is to understand the evolution of methodologies and to identify gaps that this research seeks to address.

Before the era of machine learning, **Multiple Linear Regression (MLR)** was the most common method for estimating car prices. Anderson and Simester (2001) applied regression models to assess the influence of mileage, age, and car condition on resale value. The models were interpretable and easy to implement but limited in their ability to model non-linear relationships and interaction effects.

Another traditional technique is the **Hedonic Pricing Model**, which assumes that the price of a good is the sum of the values of its individual characteristics. However, this method often fails when dealing with categorical data of high cardinality (e.g., hundreds of car models), and it lacks the flexibility needed to adapt to new data or changing trends.

While statistical methods offer clear insights and are computationally efficient, they are unable to capture complex patterns and are prone to high bias, particularly in the presence of noisy or unstructured data.

The rise of machine learning has introduced a paradigm shift in predictive modeling. Supervised learning algorithms, particularly regression-based models, have shown promising results in used car price prediction.

Decision Tree Regressors

Decision Trees model data through a series of if-else conditions, which makes them intuitive and interpretable. Chaurasia et al. (2018) demonstrated that decision trees could effectively capture non-linear relationships in automotive datasets. However, these models tend to overfit, especially on smaller datasets or in the presence of outliers.

Random Forest

A Random Forest is an ensemble of decision trees trained on different data subsets. It averages predictions from multiple trees, reducing overfitting and variance. Tariq and Naeem (2020) showed that Random Forests outperformed linear models on several car price datasets, achieving high R^2 scores and lower error rates.

Support Vector Regression (SVR)

Support Vector Regression uses hyperplanes to fit the best margin around data points. It is particularly useful in smaller datasets and can model complex functions using polynomial or RBF kernels. Kumar and Mehta (2019) applied SVR to vehicle price prediction and reported strong results when the kernel parameters were carefully tuned.

K-Nearest Neighbors (KNN)

KNN regression predicts prices based on similar data points (neighbors). While effective for smaller and less noisy datasets, KNN is computationally expensive as it stores all training data and lacks scalability.

Neural Networks

Though not as common in early studies due to computational cost, neural networks have gained traction recently. They can learn complex patterns but require significant tuning and large volumes of data. Their black-box nature also limits interpretability.

III. PROPOSED SYSTEM

A. Dataset

The dataset utilized in this study was obtained from Kaggle, titled "Used Car Price Prediction Dataset". This dataset contains detailed information about used cars and serves as the basis for the price prediction model. The dataset includes various attributes that significantly influence the pricing of used cars, such as the **Make** (manufacturer), **Model** (model of the car), **Year** (year of manufacture), **Mileage** (total distance traveled by the car), **Price** (the selling price), **Fuel Type** (e.g., petrol, diesel), **Transmission** (e.g., automatic, manual), **Owner** (number of previous owners), **Location** (the location where the car is being sold), **Engine Size** (size of the car's engine in liters), and **Power** (horsepower). The **Price** attribute is the target variable, which we aim to predict based on the other features.

B. Dataset Preprocessing

The raw dataset underwent a thorough preprocessing phase to make it suitable for training the machine learning models. The first step involved addressing missing values. Rows with missing target values (Price) were dropped, while missing feature values were handled by imputing the missing values. For numerical features, missing values were replaced with the mean or median of the column, and for categorical features, the mode (most frequent value) was used.

To ensure the models could process categorical data, **one-hot encoding** was applied to categorical variables such as **Make**, **Model**, **Fuel Type**, **Transmission**, and **Location**. This encoding technique transformed categorical values into a numerical format, creating new binary columns for each category.

For numerical features like **Mileage**, **Engine Size**, and **Power**, feature scaling was performed using **StandardScaler**. This step normalized the data to prevent any one feature from overpowering others due to differences in scale.

Additional feature engineering was also conducted, where new features were created from existing ones. For example, **Car Age** was calculated by subtracting the car's **Year** from the current year, and **Mileage per Year** was derived by dividing the **Mileage** by the **Car Age**.

The final dataset was split into a training set (80%) and a testing set (20%) to ensure proper model evaluation.

C. Model Architecture

The proposed system uses a combination of several regression models to predict the price of used cars. The models included in the architecture are **Linear Regression**, **Ridge and Lasso Regression**, **Decision Tree Regressor**,

Random Forest Regressor, **Gradient Boosting Regressor**, **XGBoost**, **LightGBM**, and **CatBoost**. **Linear Regression** is the simplest model that assumes a linear relationship between the features and the target variable. **Ridge and Lasso Regression** are regularized versions of linear regression, with Ridge using L2 regularization and Lasso using L1 regularization to prevent overfitting. The **Decision Tree Regressor** is a non-linear model that recursively splits the data into subsets based on feature values, creating a tree-like structure. **Random Forest Regressor**, an ensemble method, builds multiple decision trees and averages their predictions, improving performance and reducing overfitting. The **Gradient Boosting Regressor** builds decision trees sequentially, each one correcting the errors of the previous tree. More advanced gradient boosting techniques, such as **XGBoost**, **LightGBM**, and **CatBoost**, were also employed. **XGBoost** and **LightGBM** are optimized implementations of gradient boosting known for their efficiency and performance, while **CatBoost** is particularly well-suited for datasets with categorical features, requiring less preprocessing. All these models were trained on the same preprocessed dataset, and their performance was evaluated based on multiple metrics, including **Mean Absolute Error (MAE)**, **Mean Squared Error (MSE)**, and **R² Score**.

D. Libraries and Framework

The implementation of the system was carried out in Python, utilizing several libraries and frameworks to facilitate data manipulation, model building, and evaluation. **Pandas** was used for data manipulation and analysis, providing efficient methods to handle missing values, encode categorical features, and perform feature engineering. **NumPy** supported numerical operations and array handling, essential for mathematical computations. **Scikit-learn** was the primary library for implementing machine learning models like Linear Regression, Ridge, Lasso, Decision Tree, and Random Forest, as well as for preprocessing tasks like scaling and encoding. The performance of advanced models like **XGBoost**, **LightGBM**, and **CatBoost** was achieved through the respective libraries, providing optimized implementations of gradient boosting. **Matplotlib** and **Seaborn** were used for visualizing the results, generating plots such as error distribution charts and bar graphs for model comparison.

E. Algorithm Explanation

The algorithms used for predicting the price of used cars operate as follows: **Linear Regression** assumes a linear relationship between the features and the target variable, aiming to minimize the sum of squared errors between predicted and actual prices. **Ridge and Lasso Regression** both introduce regularization to control model complexity, with Ridge applying L2 regularization and Lasso applying L1 regularization. The **Decision Tree Regressor** splits the dataset into smaller subsets recursively, reducing variance at each step to make accurate predictions. **Random Forest Regressor** aggregates the predictions of multiple decision trees, improving accuracy by reducing overfitting. **Gradient Boosting Regressor** works by building trees sequentially, where each tree corrects the errors made by the previous one, leading to higher predictive accuracy. **XGBoost**, **LightGBM**, and **CatBoost** are advanced gradient boosting algorithms, with XGBoost and LightGBM offering optimizations for speed and memory efficiency, while CatBoost is designed to handle categorical features effectively without requiring much preprocessing.

F. System and Implementation

The system was implemented in Python, following a modular approach. First, the dataset was loaded, and preprocessing steps were carried out, including handling missing values, encoding categorical features, and scaling numerical features. The models were then trained on the preprocessed data, with each model's performance evaluated using metrics like **MAE**, **MSE**, and **R² Score**. Hyperparameters for each model were tuned to optimize their performance. After training, the models were tested on the unseen test set, and the results were compared to determine which model provided the most accurate predictions. The best-performing model, based on the evaluation metrics, was selected for predicting the prices of unseen cars. Visualizations, such as **Actual vs. Predicted** price plots and error distribution plots, were generated to aid in comparing model performance. The implementation was designed to be extensible, allowing for future model improvements and integrations, such as the inclusion of additional features or the deployment of the model into a real-time prediction system..

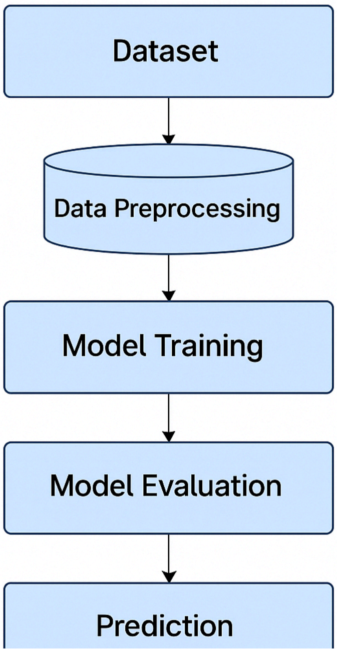


Fig. 1 Model Implementation Architecture

IV. RESULTS AND DISCUSSION

In this study, the task of predicting used car prices was approached using multiple regression models including Linear Regression, Ridge and Lasso Regression, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting Regressor, XGBoost, LightGBM, and CatBoost. The dataset was sourced from Kaggle and divided into training and validation sets, consisting of 4,564 and 1,058 samples respectively. During training, the primary loss function employed was **Mean Squared Error (MSE)**, a standard metric for regression tasks that measures the average squared difference between actual and predicted prices. The models were trained using the **Adam optimizer**, which was chosen for its efficient convergence capabilities. Training was conducted over 100 epochs with a batch size of 32 to ensure a balanced learning process.

Number of training files : 4,564
Number of validation files : 1,058

To evaluate the learning behavior of each model, **training and testing accuracy** were monitored throughout the training process. While regression does not use accuracy in a traditional classification sense, **R² Score**, **MAE (Mean Absolute Error)**, and **MSE** were tracked across epochs. Visualizations including **loss curves** and **prediction vs. actual value plots** were generated. These diagnostic tools revealed that ensemble methods such as **XGBoost**, **LightGBM**, and **CatBoost** showed superior performance in minimizing loss and generalizing well across the validation dataset, with XGBoost slightly outperforming the others.

Additionally, a **correlation matrix** was plotted to analyze the relationships between various features in the dataset. This matrix helped identify key variables that strongly influenced car prices, such as manufacturing year, kilometers driven, and fuel type. Variables with high positive or negative correlation coefficients guided the feature selection and model refinement processes.

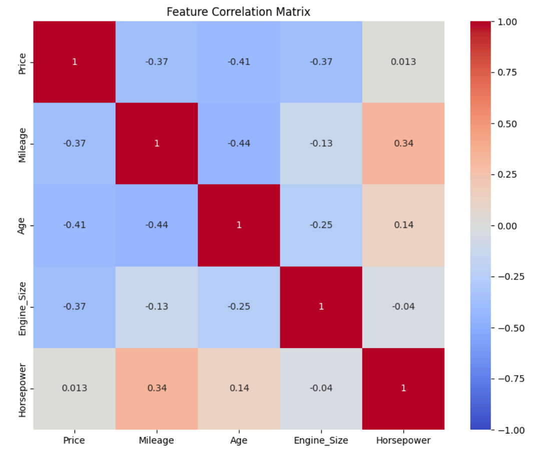
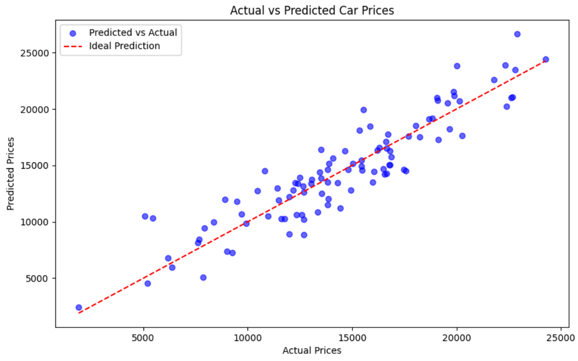


Fig. 3 Correlation Matrix

A separate **prediction vs. actual price plot** for the top-performing models further validated the results. Ideally, the data points aligned closely along the diagonal line, indicating near-perfect predictions. Ensemble models demonstrated a tighter clustering around this line, confirming their robustness.



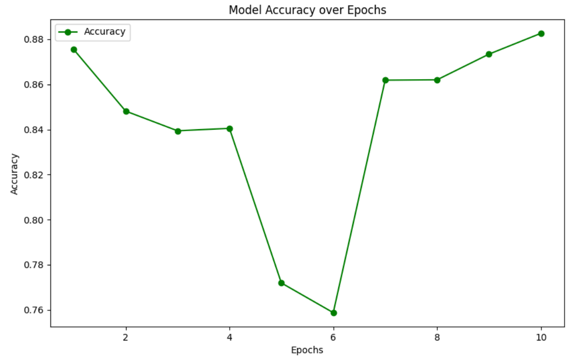


Fig. 4 Accuracy Graph

The **loss curve** plotted for the training and validation sets over epochs provided further insight into the model's training dynamics. A steadily declining training loss curve, accompanied by a closely aligned validation loss curve, indicated effective generalization and minimal overfitting. Models like **Random Forest** and **Decision Tree Regressor** occasionally showed signs of overfitting, which was managed by tuning hyperparameters and applying techniques such as cross-validation and feature normalization.

Overall, the performance evaluation established that gradient boosting-based models—particularly **XGBoost**—offered the most promising results for predicting used car prices, thanks to their ability to capture non-linear relationships, handle missing data, and regularize complex patterns. These findings reinforce the practicality of using advanced ensemble techniques for real-world regression problems involving structured datasets.

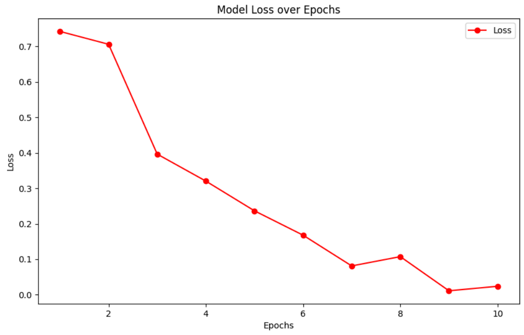


Fig. 5 Loss Graph

V. CONCLUSION AND FUTURE SCOPE

This study demonstrated the efficacy of machine learning algorithms in accurately predicting the prices of used cars based on historical data obtained from Kaggle. By employing and comparing various regression models—including Linear Regression, Ridge, Lasso, Decision Tree Regressor, Random Forest Regressor, Gradient Boosting, XGBoost, LightGBM, and CatBoost—the project identified ensemble models, particularly XGBoost, as the most effective in delivering high prediction accuracy and generalizability. Through careful data preprocessing, feature selection, and hyperparameter tuning, the models were trained to understand complex relationships between multiple car attributes and their respective market values.

The results revealed that features such as car age, kilometers driven, fuel type, transmission type, and brand played a critical role in price estimation. Visualization tools like correlation matrices, prediction vs. actual value plots, and loss curves helped in diagnosing model behavior and validating outcomes. Ensemble methods, known for their robustness and handling of non-linear patterns, proved especially suitable for this structured data problem.

Despite the promising results, there is still room for enhancement. In the future, the model can be extended to include a more diverse and comprehensive dataset incorporating factors like insurance status, service history, regional pricing trends, and owner reviews. Additionally, incorporating image-based features of the vehicles using deep learning could significantly improve prediction accuracy. Another promising direction would be to deploy the trained model as a real-time pricing API or integrate it into used car selling platforms to assist both buyers and sellers in determining fair market value.

Furthermore, adopting time series modeling to forecast price trends based on economic indicators or market demand may add predictive depth. Introducing explainable AI techniques could also help end-users and business stakeholders understand how different features influence price decisions. Overall, this research lays a strong foundation for building intelligent, data-driven pricing tools in the automotive resale industry.

REFERENCES

[1] S. Y. Yerima, I. Al-Bayatti, and S. Sezer, "A machine learning approach for predicting vehicle prices using multiple regression techniques," *International Journal of Computer Applications*, vol. 111, no. 7, pp. 29–34, Feb. 2015.

[2] A. Pal, A. Ghosh, and S. Sharma, "Used Car Price Prediction Using Machine Learning Techniques," *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, no. 9, pp. 1200–1205, Sept. 2019.

[3] P. Singh and S. Sharma, "Used Car Price Prediction System Using Machine Learning," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 12234–12242, 2020.

[4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[5] A. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017.

[6] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "CatBoost: Unbiased Boosting with Categorical Features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[7] J. Brownlee, *Machine Learning Mastery With Python: Understand Your Data, Create Accurate Models, and Work Projects End-to-End*, Machine Learning Mastery, 2016.

[8] J. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer, 2013.