Year	2024-25
Semester	6th Semester
Section	B
Group No.	B3
Title of Work	Second Hand Car Price Prediction
Name of Students	Ifra Sheikh Anushka Hedaoo Vansh Shende
USN No.	CS22008 CS22009 CS22010
Guided By	Mr. Sachin Balvir

Abstract:

Accurate prediction of second-hand car prices plays a vital role in assisting both buyers and sellers in making well-informed decisions. This project aims to develop a reliable price prediction system using machine learning techniques, leveraging a dataset that includes features like engine capacity, horsepower, fuel economy, and manufacturer information.To achieve this, we implement several regression algorithms—namely Linear Regression, Decision Tree, Random Forest, and K-Nearest Neighbors—to assess and compare their predictive capabilities. The dataset is carefully preprocessed to address missing data, encode categorical fields, and standardize numerical values for optimal model performance.Model effectiveness is evaluated using the Mean Squared Error (MSE) metric, and performance comparisons are conducted using two train-test split strategies: 80-20 and 70-30. The study offers a comprehensive analysis of which model performs best in forecasting second-hand car prices.

1. Introduction

Buying or selling a second-hand car can be tricky—pricing it just right is a challenge. The price of a used car depends on various factors, such as its brand, model, age, mileage, fuel type, transmission, and overall condition. If the price is too high, buyers may lose interest; if it's too low, sellers might incur losses. That’s why accurate price estimation is crucial for fair deals and informed decision-making in the used car market[1],[2]

Traditionally, car prices were determined by dealers or experts who relied on experience, market trends, and past sales data [3]. However, this method often led to inconsistencies, biases, and inefficiencies, especially when handling a large number of listings. With the rise of technology, machine learning has transformed this process, making car price predictions more accurate, datadriven, and objective[4][1].

Machine learning models analyze vast amounts of historical sales data to identify key patterns and predict car prices based on features like mileage, age, and brand popularity[5]. Unlike traditional methods, these models can process massive datasets efficiently and adjust their predictions as new information comes in[6].
To make this possible, we use different machine learning algorithms, including:

- **K-Nearest Neighbors (KNN):** Compares a car with similar ones in the dataset to estimate its price.
- **Linear Regression:** Establishes a straightforward relationship between car features and price for simple predictions.
- **Decision Tree:** Breaks down price factors into easy-to-follow decision rules.
- **Random Forest:** Combines multiple decision trees to improve accuracy and reduce errors.

By leveraging these machine learning techniques, we can take the guesswork out of pricing second-hand cars, helping buyers and sellers make smarter, fairer, and more confident decisions.

2. Literature Review

Predicting the price of a used car is a valuable task for buyers, sellers, and businesses involved in the automotive market. Machine learning (ML) techniques have gained popularity in this domain as they offer data-driven insights that help estimate vehicle prices more accurately than traditional methods. Various ML algorithms, including Linear Regression, Decision Trees, Random Forest, K-Nearest Neighbors (KNN), and Artificial Neural Networks (ANNs), have been tested for car price prediction. This review explores recent research on these models, comparing their strengths and limitations.

1. Linear Regression-Based Approaches

Linear regression is widely used in predicting car prices due to its simplicity and efficiency. It establishes a relationship between independent factors like mileage, fuel type, transmission type, and the dependent variable, i.e., car price. A study applied linear regression with Apache Spark and PySpark on a dataset of 11,914 used cars and achieved an RMSE of 1.43, indicating high prediction accuracy. The study also explored the sustainability impact of used cars, highlighting how buying used cars reduces carbon emissions and supports green practices [1].

Another study used multiple regression models, finding that mileage negatively impacts car price while brand popularity positively influences resale value. The model demonstrated strong performance with an R^2 close to 1 [7]. However, linear regression struggles to capture complex nonlinear relationships between variables, making it less effective for real-world used car markets.

2. K-Nearest Neighbors (KNN) Algorithm

The K-Nearest Neighbors (KNN) algorithm is a non-parametric model that predicts car prices based on similarities with past car sales. A study applied KNN regression on a dataset with 14 variables, achieving an 85% accuracy with k=4 as the optimal neighbor count. The model recorded

an RMSE of 4.01 and an MAE of 2.01. The K-Fold cross-validation technique further improved the model's generalizability [8].
Despite its effectiveness for small datasets, KNN is computationally expensive for larger datasets and is highly sensitive to irrelevant features, affecting prediction accuracy [9],[4]

3. Decision Trees for Car Price Prediction

Decision Trees (DT) are widely used for price prediction due to their ability to handle categorical and numerical features. A study comparing Decision Tree Regression with Random Forest found that the Decision Tree model achieved 67.21% accuracy. However, DT models tend to overfit, especially with large datasets, leading to less reliable predictions [10].

Another study suggested that pruning techniques and hyperparameter tuning could improve DT performance, but bias issues remain if the tree depth is not properly controlled [5],[7].

4. Random Forest Algorithm

Random Forest (RF), an ensemble learning method, enhances prediction accuracy by combining multiple decision trees. A study compared Random Forest and Extra Trees Regression, concluding that RF delivered the most stable and accurate predictions, while Extra Trees Regression was three times faster but slightly less accurate [11]

Hyperparameter tuning via RandomizedSearchCV significantly improved performance, making RF one of the most reliable models for large datasets. However, high computational costs limit its applicability in real-time scenarios [5].

5. Gradient Boosting Models

Gradient Boosting methods like XGBoost and LightGBM improve predictions by sequentially correcting errors from previous models. A study combining Gradient Boosting with Decision Trees demonstrated superior accuracy over standalone Decision Trees. The research emphasized that feature selection and proper hyperparameter tuning were essential for performance improvement [6].

While boosting models outperform traditional regression, they require significant computational power and detailed parameter tuning, making them challenging for real-world applications compared to Random Forest [12].

6. Support Vector Machines (SVM)

Support Vector Machines (SVM) are effective for small and medium-sized datasets, capturing complex relationships between features. A study applying SVM regression for car price prediction in Bosnia and Herzegovina found that SVM outperformed linear regression but was less effective

than Random Forest and Neural Networks. The results showed that SVM is useful for structured datasets but does not scale well for large datasets [1].

7. Artificial Neural Networks (ANNs) and Deep Learning

Artificial Neural Networks (ANNs) have been explored for used car price prediction due to their ability to learn nonlinear relationships from large datasets. A study comparing ANNs, Random Forest, and SVM found that ANNs achieved the highest accuracy, especially when trained on large, well-preprocessed datasets. However, deep learning requires significant computational resources and large labeled datasets, making it less practical for small businesses or personal use [4].

8. Time Series Forecasting (ARIMA) for Used Car Prices

Time series models like Auto-Regressive Integrated Moving Average (ARIMA) have been applied to forecast car prices based on historical trends. A study using ARIMA for used car price forecasting found it effective in identifying seasonal price patterns. However, ARIMA models struggle with sudden economic changes or government policy shifts, making them less adaptable for dynamic markets [11].

9. Ensemble Methods and Hybrid Models

Ensemble models, which combine multiple algorithms, often enhance predictive accuracy. A study integrating Decision Trees with Gradient Boosting achieved superior results, demonstrating that combining models can reduce errors. The research highlighted feature selection and preprocessing as critical factors in improving performance[2] .

Another study compared Random Forest, Gradient Boosting, and Neural Networks, concluding that ensemble methods outperformed standalone models. However, these approaches require more computational resources and longer training times.

3.Dataset Information

This study uses the Car Sales Dataset, which contains details about various car models and their pricing. The goal is to analyze sales trends and predict whether a car's price is above or below the median price based on its specifications. The dataset includes 157 records and 15 features, covering details like the car's brand, engine power, size, weight, and fuel efficiency. Since it contains both numbers and text-based categories, some preprocessing is necessary before training machine learning models.

Understanding the Dataset

- **Target Variable :**
    - Price in thousands – Represents the price of a car .

- For easier classification, we categorize prices as "High" (above median) and "Low" (below median) to predict which cars are expensive.

- **Independent Features :**
  1. **Manufacturer & Model** – The car brand and model name.
  2. **Vehicle Type** – Whether it's a sedan, SUV, or another category.
  3. **Engine Power** – Includes engine size and horsepower.
  4. **Car Dimensions** – Features like wheelbase, width, length, and curb weight.
  5. **Fuel Information** – Fuel tank capacity and mileage.
  6. **Latest Launch** – The release date of the car, which is converted into a numerical format.

Preprocessing Steps

1. **Handling Missing Values**:
   - If any numbers were missing, we filled them in using the median value of that column.
   - Missing text values (like a missing car type) were replaced with the most common category.
2. **Converting Text to Numbers**:
   - Categorical features (like Manufacturer and Vehicle Type) were label-encoded to make them understandable for machine learning models.
3. **Scaling Numerical Features**:
   - Some models, like Logistic Regression and KNN, perform better when numbers are on the same scale. So, we standardized values like engine size, weight, and fuel efficiency to avoid bias toward larger numbers.

Splitting the Data for Model Training

80%-20% Split:

- 80% of the data is used for training, and 20% for testing.
- This ensures the model learns from more data while still having enough to evaluate accuracy.

70%-30% Split:

- 70% of the data is used for training, and 30% for testing.
- A larger test set helps in better performance evaluation

Link: https://github.com/chandanverma07/DataSets/blob/master/Car_sales.csv

4. Methodology

The goal of this study is to predict car prices based on various vehicle attributes using different machine learning models. The process involves several key steps, including data preprocessing, model selection, training, evaluation, and comparison. Below, we outline each step along with the machine learning algorithms used.

1. Linear Regression

- Linear Regression assumes a linear relationship between features (independent variables) and the target variable (dependent variable).
- It lies on a straight line (or hyperplane in more than three dimensions) that minimizes the sum of squared residuals (predicted minus actual values).
- The model estimates optimal coefficients by the Ordinary Least Squares (OLS) technique.
- It is assessed with metrics such as Mean Squared Error (MSE), R-squared (R^2), and Mean Absolute Error (MAE).

2. Decision Tree Regressor

- Decision Trees partition the dataset into smaller subsets based on feature-based conditions in a tree-like fashion.
- The model chooses the optimal feature for splitting based on variance minimization (for regression).
- The tree grows further until it meets stopping criteria (e.g., maximum depth, minimum samples per split).
- **Overfitting** is managed through pruning methods or limiting the depth.
- Performance is measured by MSE, R^2, and MAE.

3. Random Forest Regressor

- Random Forest is an ensemble algorithm that constructs many Decision Trees and aggregates their predictions.
- It does bagging, where a random subset of data with replacement (bootstrap sampling) is used to train each tree.
- Average predictions are used to minimize variance and enhance generalization.
- It avoids overfitting in contrast to a single Decision Tree.
- Metrics used for evaluation: MSE, R^2, and MAE.

4. Gradient Boosting Regressor

- Gradient Boosting is an ensemble method that constructs models sequentially, and each subsequent tree is designed to correct the mistakes of the previous tree.

- In contrast to Random Forest (which constructs independent trees), it targets errors that were made by earlier models.
- Has a loss function (e.g., squared error) and uses gradient descent to minimize errors in predictions.
- It uses learning rate control to prevent overfitting.
- MSE, R^2, and MAE are used to measure performance.

5 .K-Nearest Neighbors (KNN) Regressor

- A non-parametric algorithm that predicts the target based on the k nearest neighbors in feature space.
- Uses distance metrics (e.g., Euclidean distance) to find the closest points.
- Averages the target values of the k nearest neighbors to make predictions.
- The choice of **k** (number of neighbors) significantly impacts model performance—higher values smooth predictions.
- Performance is evaluated using MSE, R^2 score, and MAE.

Performance Metrics:

1. **Mean Absolute Error (MAE):**
   - MAE calculates the average absolute difference between the actual and predicted values.
   - It gives a straightforward interpretation of the model's average prediction error in the same unit as the target variable.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

   - Interpretation**:** Lower values indicate better model accuracy

2. **Mean Squared Error (MSE)**:
   - MSE measures the average of the squared differences between actual and predicted values.
   - Squaring the errors puts a greater emphasis on bigger errors, so MSE is sensitive to outliers.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

.
   - Interpretation: Lower values represent greater predictive performance.

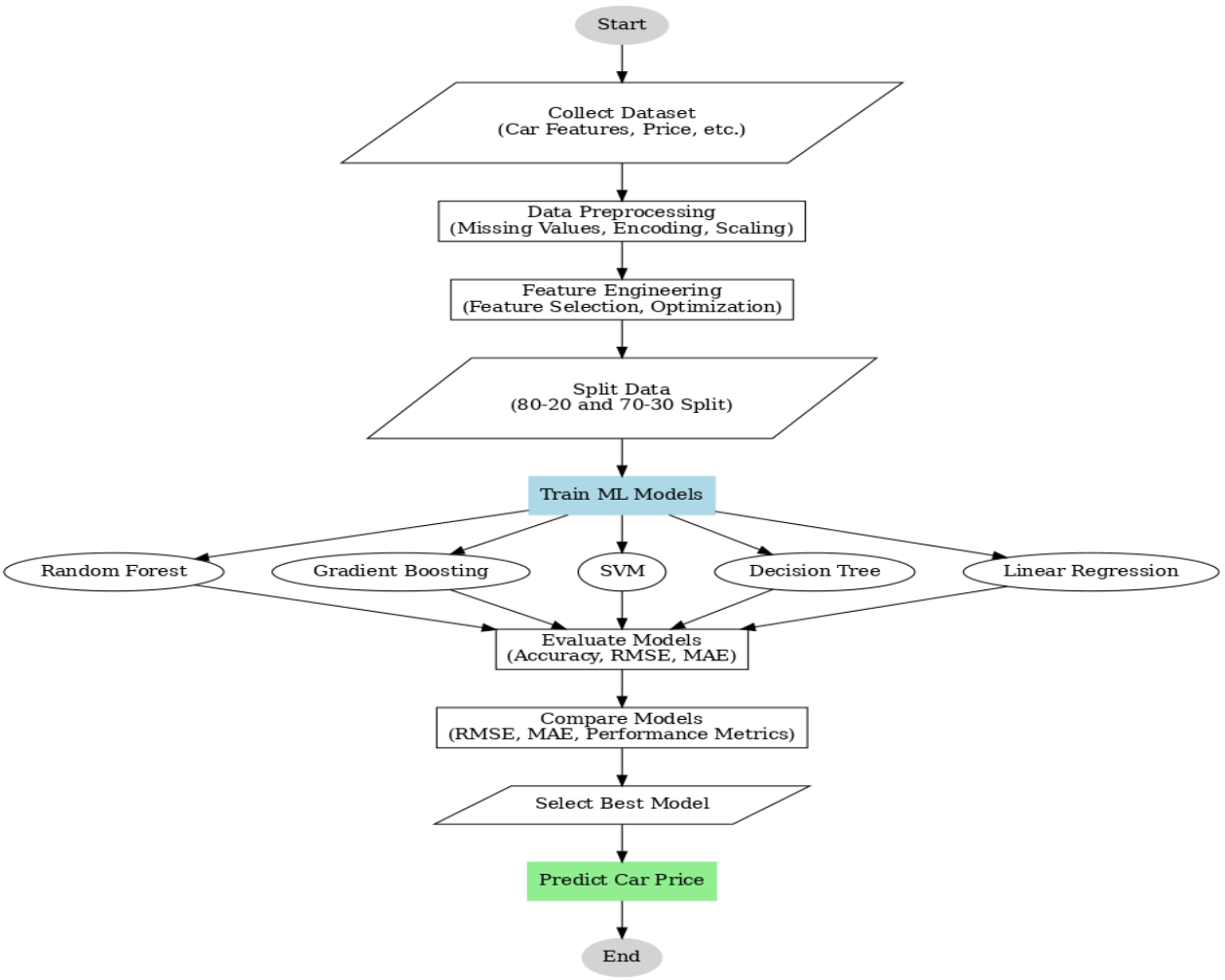
3. **R-Squared (R^2) Score**:

- R^2 measures how well the independent variables explain the variance in the dependent variable.
- It ranges from **0 to 1**, where **1** indicates a perfect fit and **0** means the model does not explain the variance.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

- Interpretation: Higher values (closer to 1) indicate a better fit of the model to the data.

Flowchart:



5. Results and Analysis

Model	MSE ↓ (70-30)	MAE ↓ (70-30)	R^2 Score ↑ (70-30)	MSE ↓ (80-20)	MAE ↓ (80-20)	R^2 Score ↑ (80-20)
Linear Regression	25.05	3.839	0.888	27.71	4.003	0.8711
Decision Tree	31.29	4.033	0.8601	27.41	3.829	0.8725
Random Forest	29.45	3.594	0.8684	30.60	3.582	0.8577
Gradient Boosting	28.20	3.643	0.8739	25.04	3.483	0.8835
KNN	65.09	5.134	0.709	59.12	4.762	0.725
SVM	181.52	7.039	0.1886	148.98	5.907	0.3069

1. Best Performing Model

- Gradient Boosting achieved the lowest MSE and highest R^2 in the 80-20 split, making it the best-performing model.
- Linear Regression performed well in the 70-30 split, with the highest R^2 score (0.8880).
- Random Forest and Gradient Boosting performed consistently well in both splits.

Best Model	70-30 Split	80-20 Split
Lowest MSE	Linear Regression (25.05)	Gradient Boosting (25.04)
Lowest MAE	Random Forest (3.594)	Gradient Boosting (3.483)

2. Model Comparisons

- Linear Regression performed well but may struggle with complex, non-linear patterns.
- Decision Tree had moderate performance, but its higher MSE suggests potential overfitting.
- Random Forest performed better than a single Decision Tree, indicating ensemble learning improves stability.
- Gradient Boosting outperformed all models in the 80-20 split, demonstrating its effectiveness in learning from errors.
- KNN had the worst performance, with the highest MSE and lowest R^2, indicating it may not be well-suited for this dataset
- Support Vector Machine (SVM) performed significantly worse than all other models, with extremely high MSE and low R^2, indicating poor generalization on this dataset.

Ranking	Model (70-30)	R^2 Score (70-30)	Model (80-20)	R^2 Score (80-20)
1st	Linear Regression	0.8880	Gradient Boosting	0.8835
2nd	Gradient Boosting	0.8739	Decision Tree	0.8725
3rd	Random Forest	0.8684	Linear Regression	0.8711
4th	Decision Tree	0.8601	Random Forest	0.8577
5th	KNN	0.7090	KNN	0.7250
6th	Support Vector Machine	0.1886	Support Vector Machine	0.3069

3. 70-30 vs. 80-20 Split Performance

- Gradient Boosting had better performance in the 80-20 split with an R^2 of 0.8835, indicating it generalizes well with more training data.
- Linear Regression performed slightly better in the 70-30 split**,** possibly due to a better fit with more training data.
- Decision Tree and Random Forest showed similar trends in both splits, but Random Forest consistently reduced MSE.
- SVM had the worst performance in both cases, showing extremely high MSE and low R^2, confirming it is not suitable for this dataset.

4. Data Splitting Analysis (80-20 vs. 70-30 Split)

Impact of Data Splitting on Model Performance:

- Linear Regression: The 80-20 split has a slightly higher MSE and MAE, but it shows a marginally better R^2 than the 70-30 split. This suggests that having more training data helps generalize better.
- Decision Tree: Performance significantly worsens in the 70-30 split, with MSE increasing and R^2 dropping. This indicates that Decision Trees overfit on a smaller training dataset.
- Random Forest: Shows a similar trend as Decision Trees but is more stable across both splits. It maintains a competitive R^2, suggesting that ensemble methods reduce overfitting.
- Gradient Boosting: Performs consistently well across both splits, showing that boosting techniques generalize well even with different train-test ratios.
- KNN: Performs poorly in both cases, with significantly high MSE and low R^2, indicating that KNN struggles with high-dimensional data.

- SVM: Had the worst performance across all models, with an extremely high MSE and low R^2 in both splits, making it unsuitable for this dataset.

5. Key Observations from Data Splitting:

- Most models perform slightly better in the 80-20 split because they get more training data for learning patterns.
- Decision Trees are highly sensitive to data splitting, as seen from their drastic drop in R^2 with a 70-30 split.
- Ensemble methods (Random Forest & Gradient Boosting) are more stable across different splits, indicating their robustness in handling varying data proportions.
- KNN fails in both cases, likely due to high variance and sensitivity to irrelevant features.
- SVM is the worst-performing model, struggling with high MSE and low R^2, showing that it does not work well on this dataset.

6.Conclusion: Gradient Boosting is the best model, achieving the highest R^2 and lowest errors, especially in the 80-20 split. Linear Regression performs well but may struggle with non-linearity. Random Forest shows stability, while Decision Tree risks overfitting. KNN performs poorly, indicating its limitations with high-dimensional data and complex relationships.

7.References:

[1] K. Samruddhi and R. Ashok Kumar, "Used Car Price Prediction using K-Nearest Neighbor Based Model," *International Journal of Innovative Research in Applied Sciences and Engineering*, vol. 4, no. 2, pp. 629–632, Aug. 2020, doi: 10.29027/ijirase.v4.i2.2020.629-632.

[2] A. Tijjani Amshi, "VEHICLE PRICE PREDICTION BY AGGREGATING DECISION TREE MODEL WITH BOOSTING MODEL."

[3] A. Pandey, V. Rastogi, and S. Singh, "'Car's Selling Price Prediction using Random Forest Machine Learning Algorithm.'" [Online]. Available: https://ssrn.com/abstract=3702236

[4] A. Chandak, P. Ganorkar, S. Sharma, A. Bagmar, and S. Tiwari, "Car Price Prediction Using Machine Learning," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 444–450, May 2019, doi: 10.26438/ijcse/v7i5.444450.

[5] J. Matematika Dan Ilmu Pengetahuan Alam LLDikti Wilayah, P. Hasan Putra, B. Purba, Y. Agustina Dalimunthe, and J. Teknik, "Random forest and decision tree algorithms for car price prediction," 2023.

[6] S. Muti and K. Yildiz, "Using Linear Regression For Used Car Price Prediction," *International Journal of Computational and Experimental Science and Engineering*, vol. 9, no. 1, pp. 11–16, Mar. 2023, doi: 10.22399/ijcesen.1070505.

[7] M. Özçalıcı, "Karar Ağaçları ve Genetik Algoritmalar ile İkinci El Otomobil Satış Fiyat Tahmini," *Alphanumeric Journal*, vol. 5, no. 1, pp. 103–103, Jun. 2017, doi: 10.17093/alphanumeric.323836.

[8] D. Budilaksana, M. Sukarsa, A. Agung, K. Agung, and C. Wiranatha, "Implementing kNearest Neighbor Methods to Predict Car Prices."

[9] P. Venkatasubbu and M. Ganesh, "Used Cars Price Prediction using Supervised Learning Techniques," *Int J Eng Adv Technol*, vol. 9, no. 1s3, pp. 216–223, Dec. 2019, doi: 10.35940/ijeat.A1042.1291S319.

[10] K. Samruddhi and R. Ashok Kumar, "Used Car Price Prediction using K-Nearest Neighbor Based Model," *International Journal of Innovative Research in Applied Sciences and Engineering*, vol. 4, no. 3, pp. 686–689, Sep. 2020, doi: 10.29027/ijirase.v4.i3.2020.686-689.

[11] A. Alhakamy, A. Alhowaity, A. A. Alatawi, and H. Alsaadi, "Are Used Cars More Sustainable? Price Prediction Based on Linear Regression," *Sustainability*, vol. 15, no. 2, p. 911, Jan. 2023, doi: 10.3390/su15020911.

[12] K. Akishev *et al.*, "DEVELOPMENT OF AN INTELLIGENT SYSTEM AUTOMATING MANAGERIAL DECISION-MAKING USING BIG DATA," *Eastern-European Journal of Enterprise Technologies*, vol. 6, no. 3(126), pp. 27–35, 2023, doi: 10.15587/1729-4061.2023.289395.