Predictive Analysis of Used Car Prices using Machine Learning Algorithms

Submitted by: CS-21146, CS-21145, CS-21019

July 8, 2024

Abstract

This report presents a machine learning approach to predict the prices of used cars using three different regression algorithms: linear regression, k-nearest neighbors (KNN) regressor, and decision tree regressor. The model uses various attributes from the provided dataset to make accurate price predictions, aiming to benefit both buyers and sellers. The dataset was sourced from Kaggle, and after splitting into training and test datasets, the performance of each model was evaluated. The decision tree regressor achieved the highest R^2 value of 0.86, making it the best-performing model among the three.

Contents

Chapter 1

Introduction

The used car market is vast and often challenging to navigate. Accurate price predictions can help buyers find fair deals and sellers price their cars competitively. With the rise of data science and machine learning, leveraging these technologies to predict used car prices offers a data-driven approach to address this challenge.

Predictive analytics, a branch of data science, involves making predictions about future events based on historical data. In the context of used car prices, predictive analytics can help in understanding the factors that influence car prices and predicting future prices based on these factors.

This report details the implementation and evaluation of three regression models—linear regression, KNN regressor, and decision tree regressor—to predict used car prices based on various attributes. The attributes considered include make, model, year of manufacture, mileage, fuel type, transmission, and city, as these significantly impact a car's value. Make and model determine brand reputation, year indicates age, mileage reflects usage, and fuel type and transmission affect efficiency and driving experience.

The objective of this project is to build an accurate machine learning model to predict used car prices, benefiting both buyers and sellers. Buyers can ensure fair deals, while sellers can competitively price their cars. Using historical data from Kaggle, the models learn patterns and relationships between car attributes and prices.

In this report, we present a comprehensive analysis of the dataset, the preprocessing steps taken to clean and prepare the data, the implementation of the three regression models, and the evaluation of their performance. The following sections provide a detailed explanation of each step and the results obtained from the analysis.

Chapter 2

Methodology

2.1 Dataset

The dataset used for this project was sourced from Kaggle, containing various attributes of used cars. Key attributes include make, model, year, mileage, fuel type, transmission, and others relevant for predicting car prices. The dataset was divided into training (80%) and test (20%) sets to train and evaluate the models.

2.2 Data Preprocessing

Data preprocessing included handling missing values, encoding categorical variables, and scaling numerical features where necessary. These steps ensured the dataset was clean and suitable for model training.

2.3 Model Training and Evaluation

Three regression models were implemented and trained:

- **Linear Regression**: This model works under the assumption that there is a linear relationship between the features and the target variable (price).
- **KNN Regressor**: This algorithm predicts the price by finding the 'k' nearest neighbors in the training data and averaging their prices.
- **Decision Tree Regressor**: This model splits the data into subsets based on feature values to minimize variance and capture non-linear relationships.

Models were evaluated using R^2 values and other relevant metrics such as mean squared error (MSE) to assess their performance on the test set.

Chapter 3

Results

The performance metrics for each model on the training and test datasets are summarized below:

3.1 Linear Regression

:

```
LINEAR REGRESSION USING PACKAGES
Mean Squared Error: 20752698855640.71
Root Mean Squared Error: 4555513.017832427
R^2 score: 0.198618580330979
Predicted price for user input: [-2.68714811e+08]
```

Figure 3.1: Linear Regression Statistics (with packages)

```
LINEAR REGRESSION WITHOUT PACKAGES
Linear Regression without sklearn:
Mean Squared Error: 20752698855640.707
Root Mean Squared Error: 4555513.017832427
R^2 score: 0.19861858033097923
Predicted price for user input: 3786600.214624502
```

Figure 3.2: Linear Regression Statistics (without packages)

3.2 KNN Regressor

:

```
KNN REGRESSOR USING PACKAGES
Mean Squared Error: 8857788015373.818
Root Mean Squared Error: 2976203.624648995
Best parameters: {'n_neighbors': 5}
After hyper parameter tuning
Best Mean Squared Error: 10310888886284.047
Best Root Mean Squared Error: 3211057.2848026315
Best R² Score: 0.601837099298826
[3111000.]
```

Figure 3.3: KNN Regressor Statistics (with packages)

```
KNN REGRESSOR WITHOUT PACKAGES
Mean Squared Error: 10313227288797.164
Root Mean Squared Error: 3211421.3813819517
R^2 Score: 0.6017835516918788
Predicted price for user input: [2850000.]
```

Figure 3.4: KNN Regressor Statistics (without packages)

3.3 Decision Tree Regressor

:

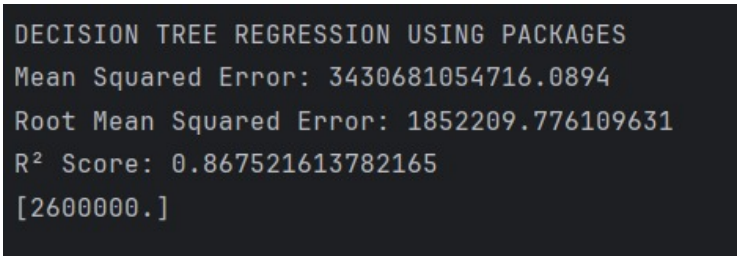

Figure 3.5: Decision Tree Regressor Statistics (with packages)

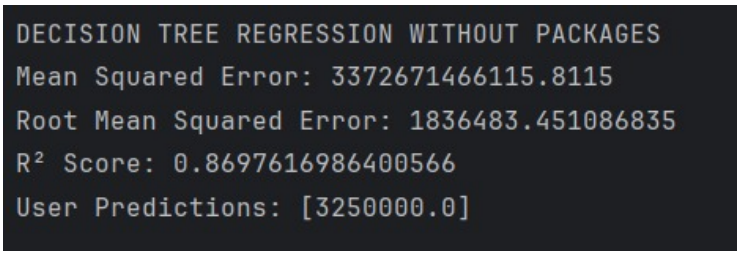

Figure 3.6: Decision Tree Regressor Statistics (without packages)

Chapter 4

Discussion

The decision tree regressor's ability to handle non-linear relationships and mixed data types contributed to its high performance. Linear regression, despite its simplicity and computational efficiency, struggled with capturing the complex patterns in the data, resulting in lower R^2 values. The KNN regressor performed better than linear regression but required significant computational resources due to its instance-based learning approach.

4.1 Comparison of Models

The screenshot below shows the Comparisons of MSE and R^2 scores.

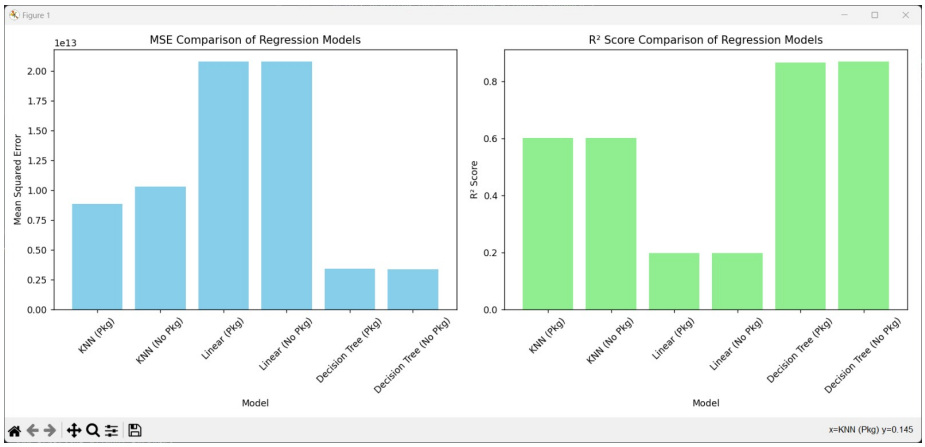

Figure 4.1: Comparison of Models (MSE and R^2)

Chapter 5

Conclusion

This study demonstrates the effectiveness of using machine learning algorithms to predict used car prices. Among the models tested, the decision tree regressor emerged as the best performer with an R^2 value of 0.86 on the test data. Future work could explore ensemble methods and further feature engineering to enhance model performance.

Chapter 6

References

- GeeksforGeeks, *Machine Learning - Linear Regression*, GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/ml-linear-regression/. [Accessed: July 8, 2024].
- GeeksforGeeks, *Python - Decision Tree Regression using sklearn*, GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/python-decision-tree-regression-using-sklearn/. [Accessed: July 8, 2024].
- scikit-learn, *Plot the decision tree regressor*, scikit-learn. [Online]. Available: https://scikit-learn.org/stable/auto$_e$*xamples*/*tree*/*plot*$_t$*ree*$_r$*egression.html*.[*Accessed* : *July*8, 2024].
- scikit-learn, *sklearn.neighbors.KNeighborsRegressor*, scikit-learn. [Online]. Available: https://scikit-learn.org/stable/api/sklearn.neighbors.html. [Accessed: July 8, 2024].