Predicting Used Car Prices with Machine Learning Algorithms

Divyesh Rathod (drathod2@asu.edu)[1],
Meet Shah (mshah72@asu.edu)[1],
Vedant Thaker (vthaker1@asu.edu)[1]
[1] Arizona State University

I. Abstract

In this study, we aim to predict the prices of used cars based on various features using different machine learning algorithms. We used a publicly available dataset of 20,063 used car records with 11 features. After processing the data, we trained and compared linear regression, lasso regression, ridge regression, and polynomial regression algorithms on a local computer, Google Colab, and AWS SageMaker instance. The results and performance of each model are discussed, along with their training times on different platforms.

II. Introduction

The goal of this study is to develop a model that can predict the prices of used cars based on various features such as trim, mileage, year, color, and other factors. This is important because accurate price prediction can help buyers and sellers make informed decisions when purchasing or selling a used car. In this project, we used machine learning algorithms to train a model that can make such predictions with reasonable accuracy. Specifically, we experimented with linear regression, lasso regression, ridge regression, and polynomial regression algorithms, and compared their performance on different platforms.

III. Data Description

The dataset used in this project consists of 20,063 used car records with 11 features: price, trim, isOneOwner, mileage, year, color, displacement, fuel, region, soundSystem, and wheelType. The target variable is the price, while the other 10 features represent various features of the cars. The dataset includes both numerical and categorical features. Table below provides information about the features present in the dataset.

Feature	**Category**	**Description**
price	Numerical	Price of the used car
trim	Categorical	Trim level of the car
isOneOwner	Categorical	Whether the car had only one owner
mileage	Numerical	Mileage of the car
year	Numerical	Year of manufacture
color	Categorical	Color of the car
displacement	Numerical	Engine displacement (in liters)
fuel	Categorical	Type of fuel used by the car
region	Categorical	Geographical region of the car
soundSystem	Categorical	Type of car sound system
wheelType	Categorical	Type of car wheel

Table 1: Description of dataset

IV. Data Visualization

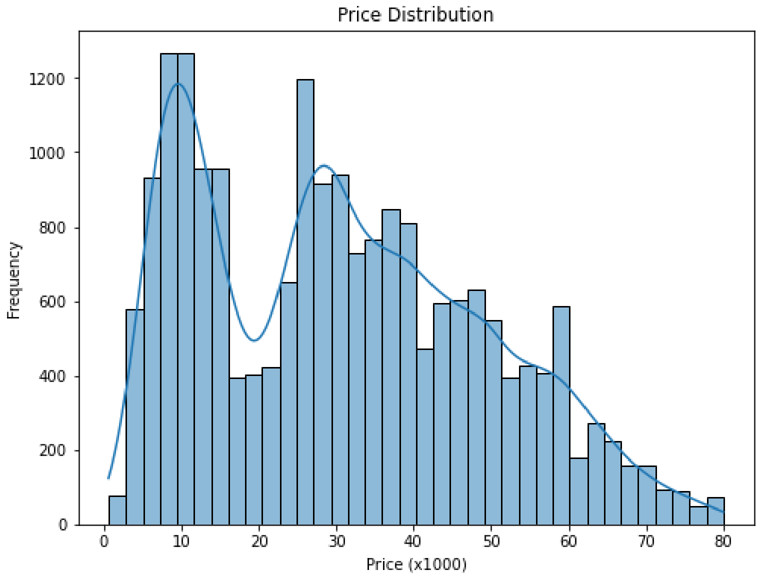

Fig.1: This visualization presents the distribution of car prices in the dataset. It is important to understand the overall distribution of the dependent variable (price) to identify any potential outliers or skewness in the data that might impact our model's performance.

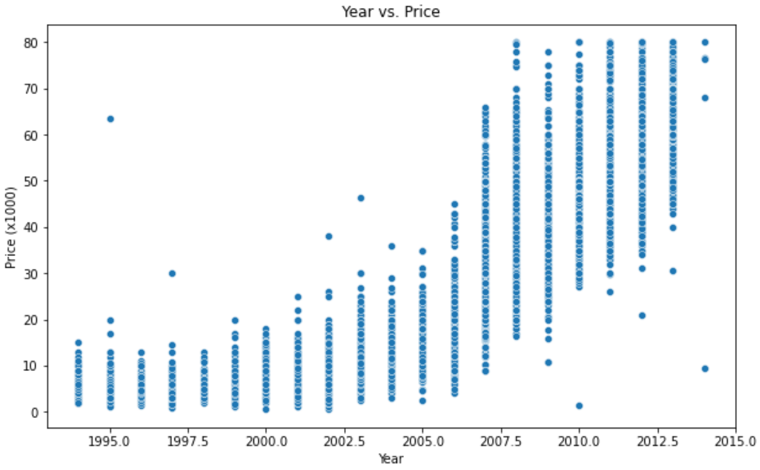

Fig. 2: This scatter plot displays the relationship between the manufacturing year of the cars and their prices. It helps in identifying any trends or patterns between the age of the cars and their selling price, which might be a significant factor in predicting the car prices.

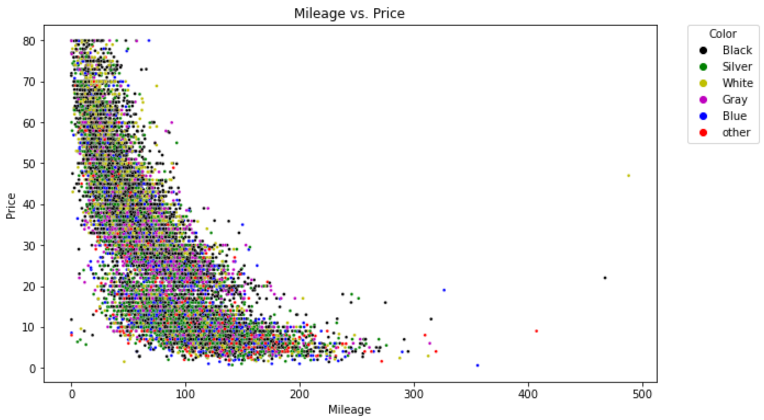

Fig. 3: This scatter plot showcases the relationship between the mileage of the cars and their prices, with each data point colored according to the car's color. It helps in understanding how the mileage affects the car price and whether the color of the car has any impact on the relationship between mileage and price.

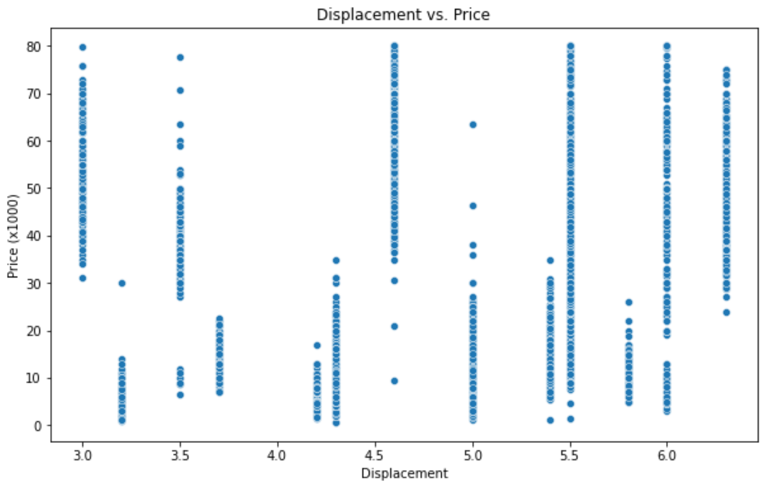

Fig. 4: This scatter plot presents the relationship between the engine displacement of the cars and their prices. It helps in determining if there is a correlation between the engine size and the selling price of the cars, which might be a useful factor in predicting car prices.

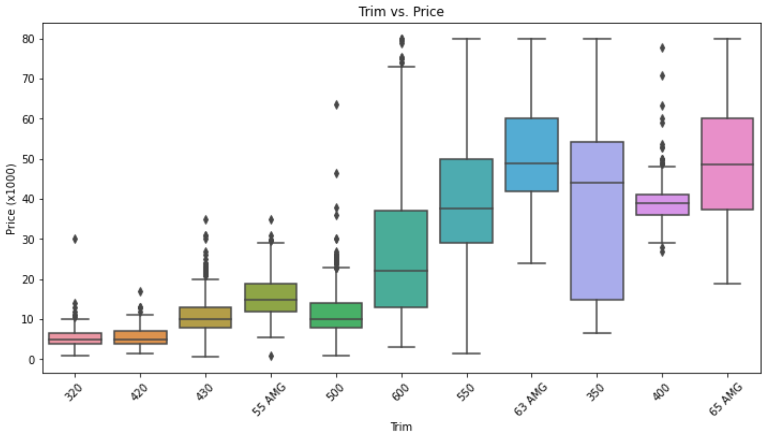

Fig. 5: This box plot displays the distribution of car prices for different colors. It helps in determining if the color of the car has any significant impact on its price, which might be an important factor to consider when predicting car prices.

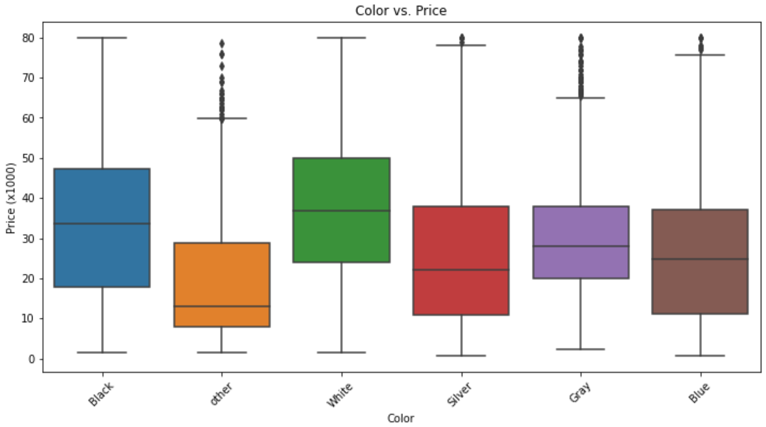

Fig. 6: This box plot displays the distribution of car prices for different colors. It helps in determining if the color of the car has any significant impact on its price, which might be an important factor to consider when predicting car prices.

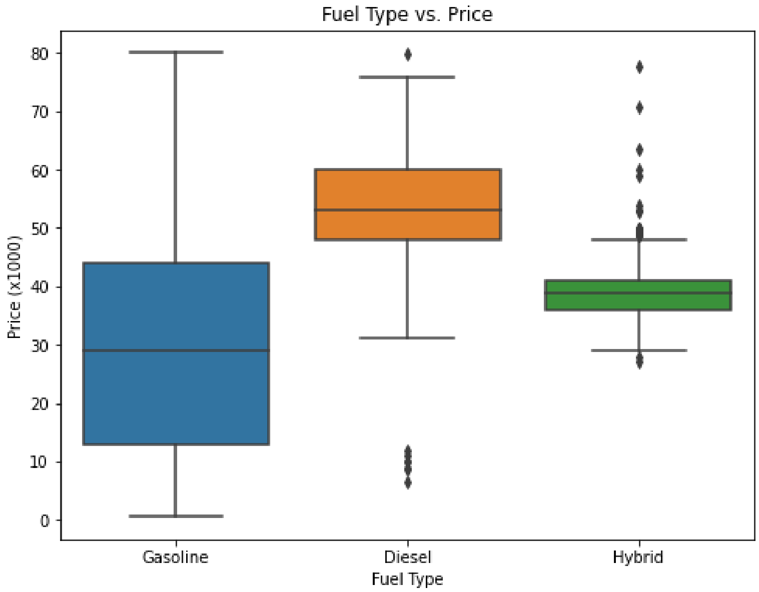

Fig. 7: This box plot shows the distribution of car prices for different fuel types. It helps in understanding how the fuel type affects the car price and whether certain fuel types are more expensive than others, which might be a significant factor in predicting the car prices.

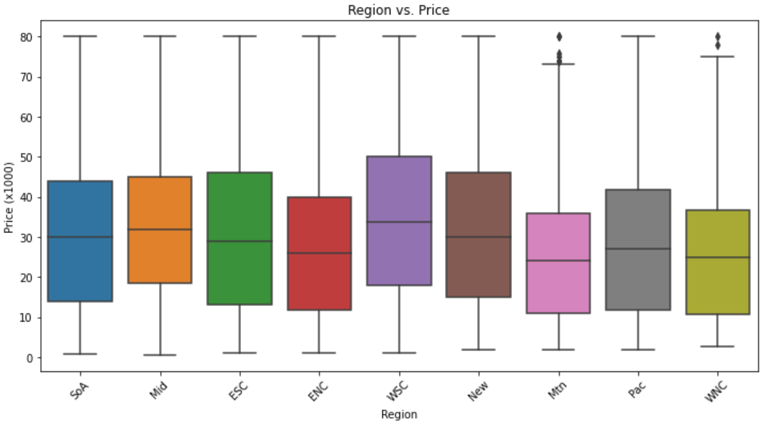

Fig. 8: This box plot presents the distribution of car prices across different regions. It helps in identifying if there are any regional differences in car prices, which might be due to factors such as regional demand, taxes, or other economic factors that can influence the car's value.

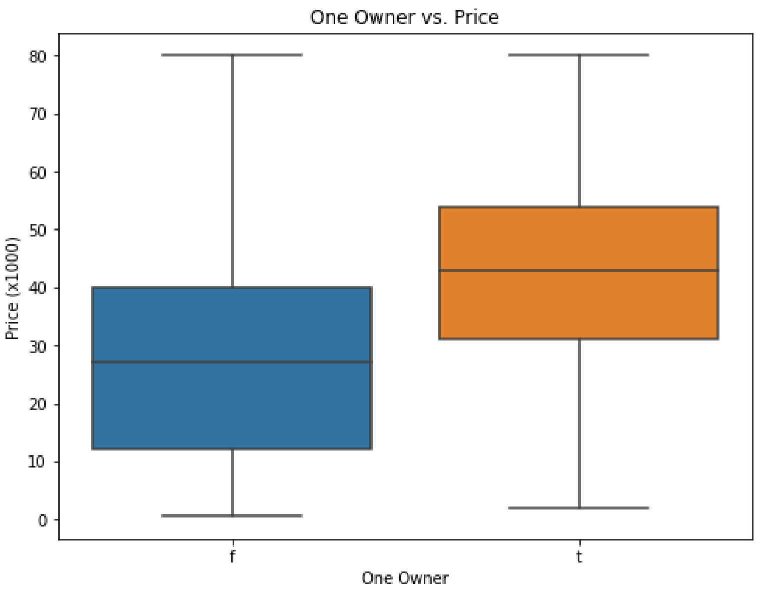

Fig. 9: This box plot shows the distribution of car prices for cars with one owner versus multiple owners. It helps in understanding the impact of car ownership history on its price, as cars with a single owner might be perceived as better maintained or more reliable, which can affect their value.

V. Data Processing

The following steps were performed to process the data:

1. Read the data.
2. Replaced 'unsp' with 'NaN' (missing values) in the 'color', 'soundSystem', and 'wheelType' columns.
3. Dropped 'soundSystem' and 'wheelType' columns from the training data, as they had more than 40% of missing data.
4. Filled in the missing values in the 'color' column, which had nearly 4% missing data, using the IterativeImputer technique. To apply this method, we converted the categorical values of the 'color' column to numerical values using the OrdinalEncoder method.
5. Selected the best features using the SelectKBest method. Based on this method, we chose the following five features: 'isOneOwner', 'mileage', 'year', 'displacement', and 'trim'.
6. In addition to SelectKBest, we also used the PCA method for feature selection.
7. Split the data into train (80%) and test (20%) sets for model training and evaluation.

VI. Plan of Attack

We planned to train and compare 7 different regression algorithms to predict used car prices:

1. Decision Tree
2. KNN Regression
3. Lasso Regression
4. Linear Regression
5. Polynomial Regression
6. Random Forest Regression
7. Ridge Regression

These algorithms were chosen because they are commonly used for regression problems and can provide a good baseline for comparison. In addition, we trained the models on three different platforms – local computer, Google Colab, and AWS SageMaker – to compute the training times and compare their performance.

VII. Results

	Mean Squared Error	**Root Mean Squared Error**	**R Square**	**Mean Absolute Error**
Decision Tree Model	18.91	4.35	0.94	3.1
Decision Tree Model (with PCA)	31.11	5.58	0.91	3.7
KNN Regression	21.83	4.67	0.93	3.27
Lasso Regression Model	45.97	6.78	0.86	5.1
Linear Regression Model	45.98	6.78	0.86	5.1
Polynomial Regression Model	17.13	4.14	0.95	2.93
Random Forest Model	23.45	4.84	0.95	3.44
Random Forest Model (with PCA)	4.99	2.23	0.98	1.5
Ridge Regression Model	45.97	6.78	0.86	5.1

Table 2: Model performance comparison

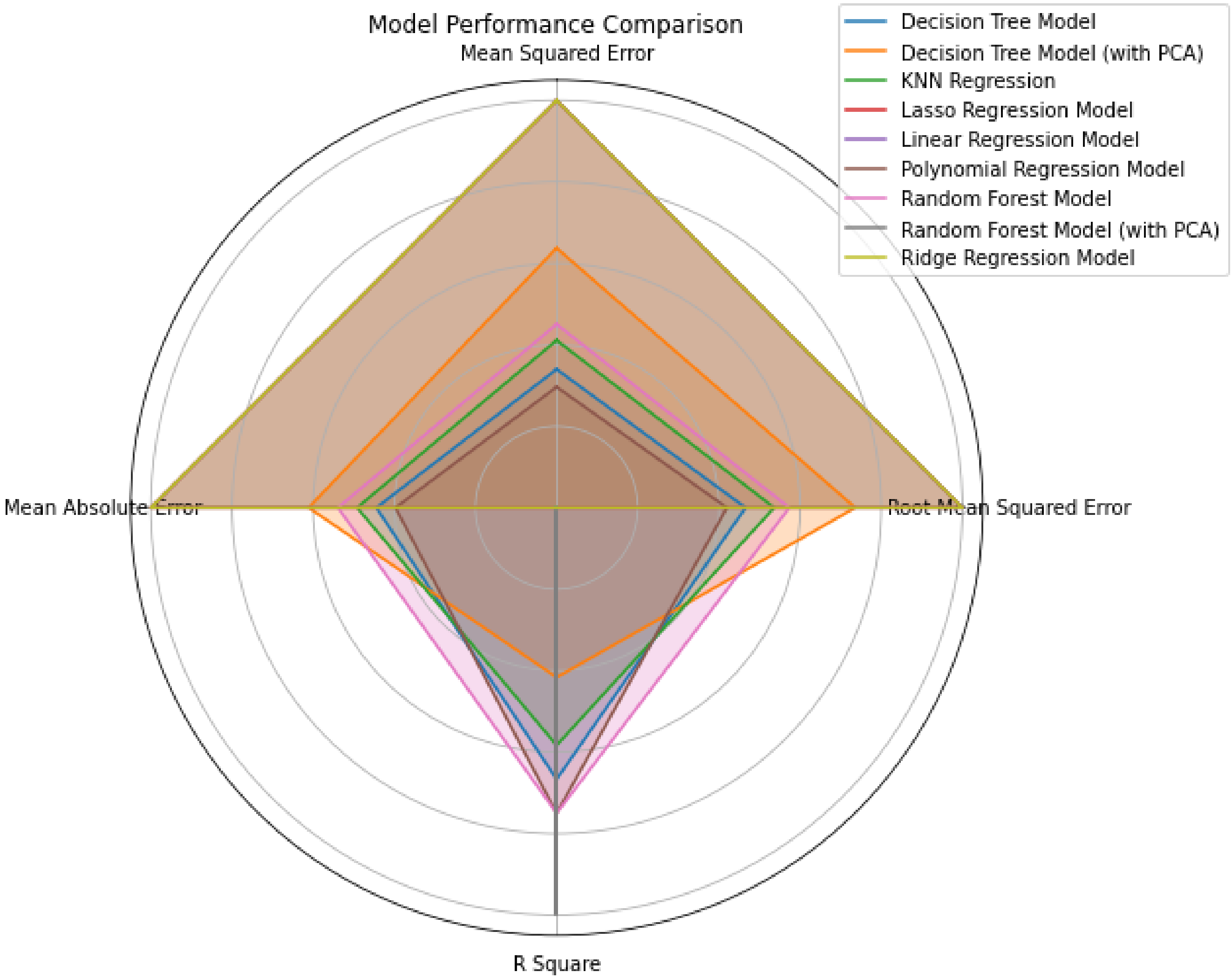

Fig. 10: Graphical representation of model performance

We trained and evaluated various machine learning models to predict used car prices based on the selected features. Here is a detailed discussion of the results for each model:

1. **Decision Tree Model**: This model demonstrated high prediction accuracy with an R Square value of 0.94 and a Mean Absolute Error of 3.1. The training time on AWS SageMaker was the fastest, followed by Google Colab and the traditional computer. The model's performance indicates that it can effectively capture the relationship between the selected features and the used car prices.

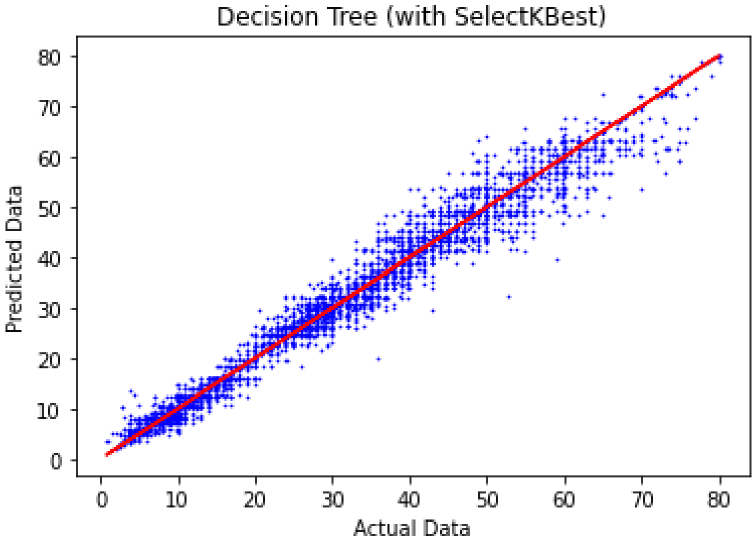

Fig. 11: Plot for Actual Data vs. Predicted Data for Decision Tree Model with feature selection done using SelectKBest Technique

2. **Decision Tree Model (with PCA)**: The model with PCA had slightly lower prediction accuracy compared to the non-PCA model, with an R Square value of 0.91 and a Mean Absolute Error of 3.7. The training time was the fastest on AWS SageMaker, followed by Google Colab and the traditional computer. The results suggest that the use of PCA may not significantly improve the model's performance in this case.

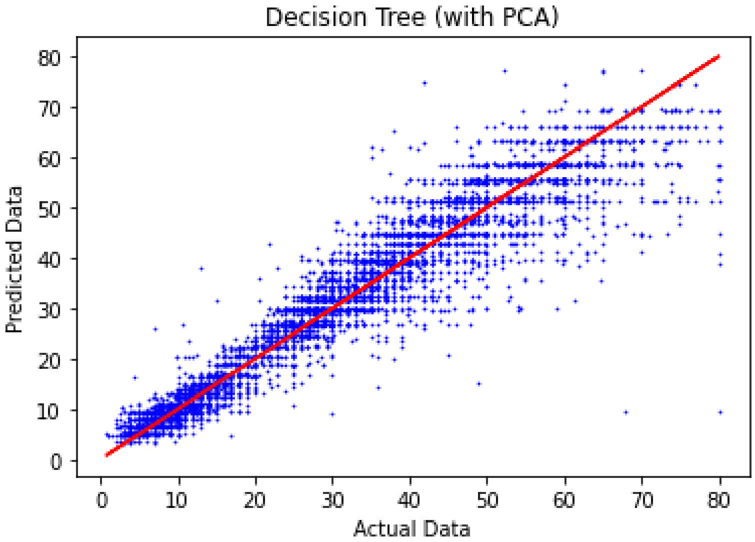

Fig. 12: Plot for Actual Data vs. Predicted Data for Decision Tree Model with feature selection done using PCA Technique

3. **KNN Regression**: This model achieved an R Square value of 0.93 and a Mean Absolute Error of 3.27, indicating good prediction accuracy. The training time was the shortest on AWS SageMaker, followed by Google Colab and the traditional computer. The KNN Regression model is a valuable alternative to tree-based models and can provide accurate predictions for this dataset.

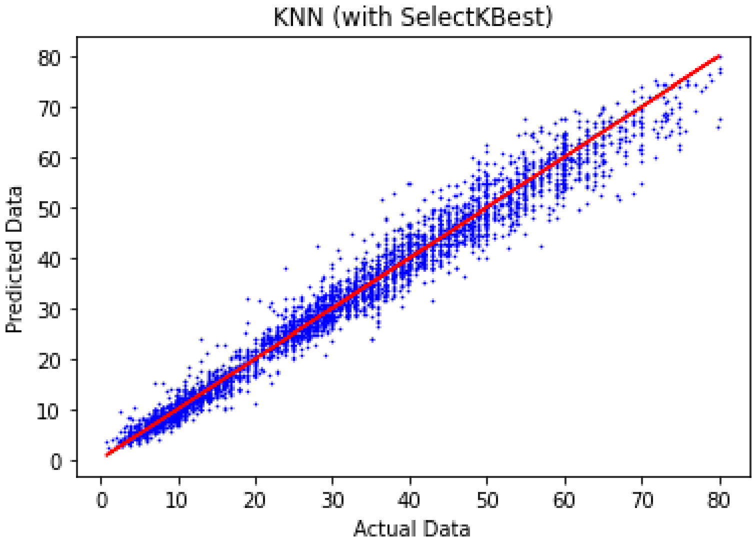

Fig. 13: Plot for Actual Data vs. Predicted Data for Decision Tree Model with feature selection done using PCA Technique

4. **Lasso Regression Model**: This model had a lower prediction accuracy compared to the other models, with an R Square value of 0.86 and a Mean Absolute Error of 5.1. The training time was similar across all platforms, with AWS SageMaker being slightly faster. The results show that the Lasso Regression Model may not be the best choice for this dataset due to its lower accuracy.

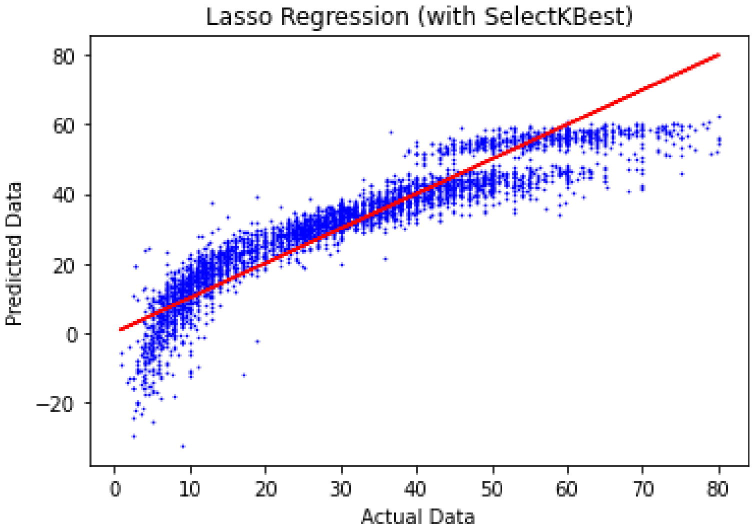

Fig. 14: Plot for Actual Data vs. Predicted Data for Lasso Regression with feature selection done using SelectKBest Technique

5. **Linear Regression Model**: The model exhibited a lower prediction accuracy, with an R Square value of 0.86 and a Mean Absolute Error of 5.1. The training time was the fastest on AWS SageMaker, followed by Google Colab and the traditional computer. The results indicate that the Linear Regression Model may not be the most suitable for predicting used car prices for this dataset.

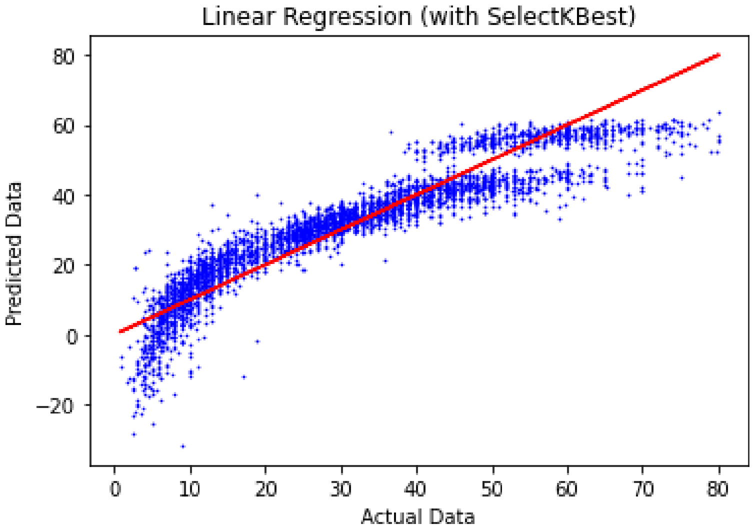

Fig. 15: Plot for Actual Data vs. Predicted Data for Linear Regression with feature selection done using SelectKBest Technique

6. **Polynomial Regression Model**: This model showed high prediction accuracy, with an R Square value of 0.95 and a Mean Absolute Error of 2.93. The training time was the shortest on AWS SageMaker, followed by Google Colab and the traditional computer. The Polynomial Regression Model's performance suggests that it can effectively model the non-linear relationship between the selected features and the used car prices.

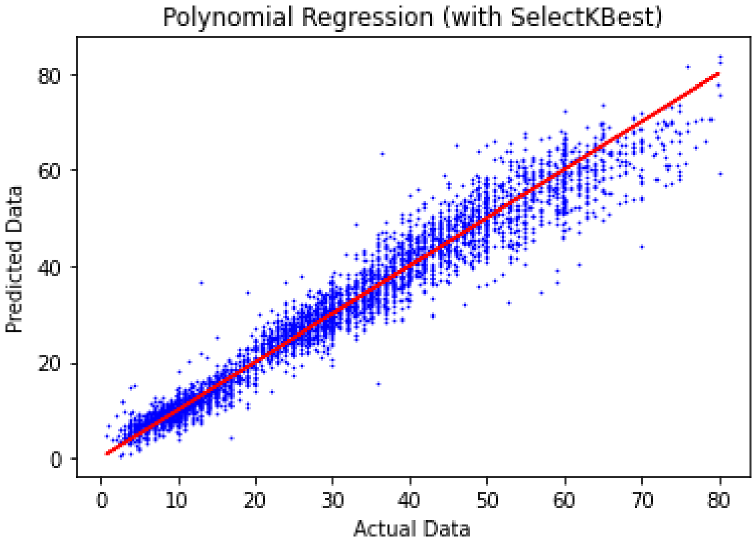

Fig. 16: Plot for Actual Data vs. Predicted Data for Polynomial Regression with feature selection done using SelectKBest Technique

7. **Random Forest Model**: This model demonstrated high prediction accuracy, with an R Square value of 0.95 and a Mean Absolute Error of 3.44. The training time was the fastest on AWS SageMaker, followed by Google Colab and the traditional computer. The Random Forest Model's strong performance makes it a viable option for predicting used car prices in this dataset.

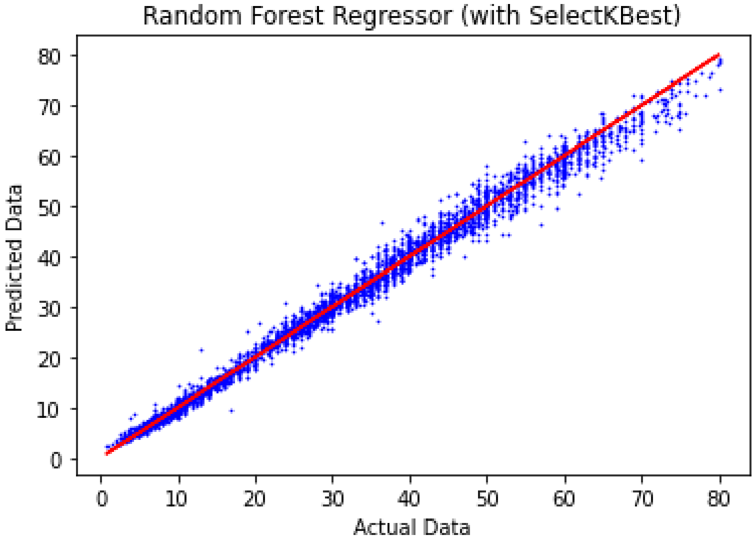

Fig. 17: Plot for Actual Data vs. Predicted Data for Random Forest Regression with feature selection done using SelectKBest Technique

8. **Random Forest Model (with PCA)**: The model had the highest prediction accuracy among all models, with an R Square value of 0.98 and a Mean Absolute Error of 1.5. However, the training time was longest on Google Colab, followed by AWS SageMaker and the traditional computer. The results show that the Random Forest Model with PCA can provide the most accurate predictions, but it requires more computational resources on some platforms.

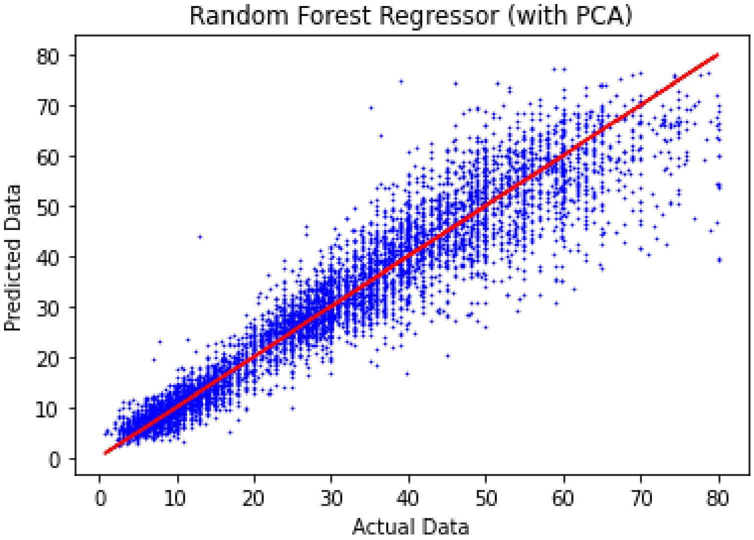

Fig. 18: Plot for Actual Data vs. Predicted Data for Random Forest Regression with feature selection done using PCA Technique

9. **Ridge Regression Model**: This model had a lower prediction accuracy compared to the other models, with an R Square value of 0.86 and a Mean Absolute Error of 5.1. The training time was the fastest on AWS SageMaker, followed by Google Colab and the traditional computer. The Ridge Regression Model's performance suggests that it may not be the best choice for this dataset.

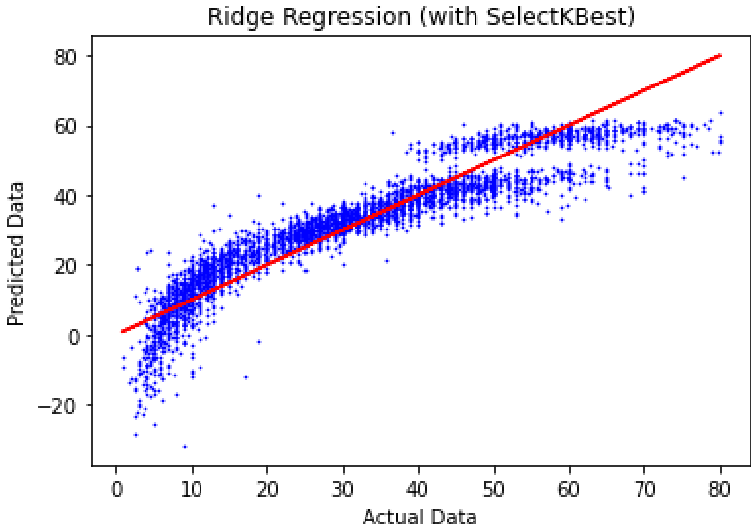

Fig. 19: Plot for Actual Data vs. Predicted Data for Random Forest Regression with feature selection done using PCA Technique

	Traditional Computer (RAM - 8.00 GB, vCPU - 8)	Google Colab (RAM - 12.7 GB, vCPU - 2)	AWS SageMaker (RAM - 32.00 GB, vCPU - 8)
Decision Tree Model	0.050864	0.034027	0.019665
Decision Tree Model (with PCA)	0.080812	0.022827	0.029848
KNN Regression	0.066254	0.016832	0.027677
Lasso Regression Model	0.012966	0.012229	0.009077
Linear Regression Model	0.075797	0.050673	0.006553
Polynomial Regression Model	0.175228	0.162937	0.089919
Random Forest Model	5.846378	2.206671	2.976913
Random Forest Model (with PCA)	5.520597	5.133787	3.127575
Ridge Regression Model	0.006937	0.012000	0.004187

Table 3: Computation timings (in seconds) on different platforms

VIII. Conclusion

Based on the results obtained, the Random Forest Model with PCA had the highest prediction accuracy, with an R Square value of 0.98 and a Mean Absolute Error of 1.5. However, its training time was the longest on Google Colab, while AWS SageMaker and the traditional computer had comparatively shorter training times.

The Polynomial Regression Model and Random Forest Model also demonstrated high prediction accuracy, with R Square values of 0.95 and Mean Absolute Errors of 2.93 and 3.44, respectively. Their training times were the shortest on AWS SageMaker, followed by Google Colab and the traditional computer.

The remaining models, including Decision Tree Models, KNN Regression, Lasso Regression Model, Linear Regression Model, and Ridge Regression Model, exhibited varying levels of prediction accuracy and training times across platforms.

In conclusion, the choice of the best model for predicting used car prices depends on the desired trade-off between prediction accuracy and training time. The Random Forest Model with PCA provides the highest accuracy but requires more computational resources on some platforms. Other models, such as the Polynomial Regression Model and Random Forest Model, offer a balance between accuracy and training time, making them suitable for situations with limited computational resources. The remaining models may still be viable options when training time is a critical factor, although they have lower prediction accuracies.

It is essential to consider the specific requirements of a given project or application when selecting the most appropriate model. In some cases, it may be necessary to fine-tune hyperparameters or experiment with different feature selection techniques to improve model performance further. Regardless, the results presented here provide valuable insights into the relative strengths and weaknesses of each model and can serve as a starting point for further exploration and optimization.

IX. Resources

Find the complete code to the above project here - https://github.com/mshah72/MachineLearningAndDeepLearning-Project