Report : CSE445

Title: Machine Learning Car Price Predictor Model

Submitted by:
Rashiqur Rahman Rifat 1911445642
Refah Tasfia 1821835042
Kanis Fatema Oni 1821174642

Abstract

The importance and purpose of the work:
The machine learning car price predictor model aims to address the need for accurate and efficient pricing predictions in the automotive industry. The pricing of cars is influenced by a multitude of factors, such as brand, model, age, mileage, and market demand. Traditional methods of pricing estimation often lack accuracy and require manual analysis. This project's importance lies in developing a model that can automate the process, providing reliable and timely price predictions for cars based on relevant features.

Notable experimental results obtained:
The machine learning car price predictor model demonstrates promising performance in accurately predicting car prices. During evaluation, the model achieves a high level of accuracy, as measured by metrics like mean absolute error or root mean square error. It outperforms traditional pricing methods and showcases its potential to enhance decision-making processes in the automotive industry.

Novelty:
The novelty of this machine learning project lies in its ability to accurately predict car prices using a combination of relevant features. By leveraging a vast dataset and employing advanced machine learning techniques, the model captures complex relationships and patterns that contribute to car pricing. This predictive capability enables businesses and consumers to make informed decisions, such as setting competitive prices, negotiating fair deals, or assessing the value of their vehicles. Furthermore, the model's potential for automation saves time and resources, making it a valuable tool in the ever-evolving automotive market.

Introduction:

The Car Price Prediction Model project aims to develop a machine learning model capable of predicting the prices of used cars based on various features and factors. This project is designed to assist car buyers, sellers, and enthusiasts in estimating the fair market value of a vehicle, facilitating informed decision-making.
Buying or selling a car can be a complex process, as determining an accurate price requires considering multiple factors such as make, model, year, mileage, condition, and more. Human judgment alone can lead to subjective or inaccurate pricing. Therefore, a data-driven approach using machine learning techniques can provide valuable insights and enhance the decision-making process.

Objectives:

The primary objectives of the Car Price Prediction Model project are as follows:

a. Develop a robust machine learning model that accurately predicts the prices of used cars.
b. Utilize historical car data and relevant features to train the model and ensure generalizability.
c. Evaluate and fine-tune the model to optimize its performance and accuracy.

d. Provide an intuitive and user-friendly interface for users to input car features and obtain predicted prices.

Expected Outcomes:
The Car Price Prediction Model project aims to provide the following outcomes:

a. A well-trained machine learning model capable of predicting used car prices accurately.
b. Insights into the influential features affecting car prices, enabling users to make informed decisions.
c. A user-friendly interface or web application for easy interaction with the model.
d. Enhanced decision-making for car buyers, sellers, and enthusiasts by providing estimated market values.

By implementing this project, users will have a reliable tool to estimate car prices accurately, minimizing potential losses in selling or overpaying when purchasing a used car.

Problem Statement:
The problem at hand is to develop a machine learning model that can accurately predict the price of cars based on their relevant features. The model should be capable of taking various attributes of a car, such as make, model, year, mileage, condition, and other relevant factors, and generate an estimated price for the vehicle.

This car price prediction model can be valuable for various stakeholders in the automotive industry, including car buyers, sellers, and dealerships. Car buyers can benefit from having a reliable estimation of the fair market value of a vehicle before making a purchase, enabling them to negotiate better deals. Car sellers and dealerships can use the model to set competitive prices for their inventory, optimizing their sales strategy.

To address this problem, we will collect a comprehensive dataset of car listings, including both historical and current data, containing relevant attributes and their corresponding prices. This dataset will serve as the basis for training and evaluating the machine learning model. The aim is to develop a model that can generalize well to unseen car data and provide accurate price predictions.

The successful implementation of this car price predictor model will provide a valuable tool for both car buyers and sellers, improving transparency and efficiency in the automotive market.

Literature review:

When it comes to the task of car price prediction using machine learning, there are several related works that can be clustered into different groups based on their approach and methodology. Here are four major groups with explanations for each:

1. Regression-based Approaches:
This group includes works that utilize various regression techniques to predict car prices. Linear regression, polynomial regression, decision trees, and random forests are commonly employed. Researchers typically focus on feature engineering, selecting relevant attributes such as car age, mileage, brand, model, engine specifications, and historical pricing data to train their models.

2. Neural Network-based Approaches:
This group consists of studies that leverage neural networks for car price prediction. Deep learning models like feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs) are used to capture complex patterns in car data. Some works explore the use of pre-trained models like VGG, ResNet, or LSTM networks to extract features from car images or text descriptions for predicting prices.

3. Ensemble-based Approaches:
In this group, researchers focus on combining multiple models to improve prediction accuracy. Techniques such as bagging, boosting, and stacking are employed to create ensembles of regression models or neural networks. The idea is to leverage the strengths of different models and reduce the impact of individual model biases or errors.

4. Hybrid Approaches:
This group includes works that combine machine learning techniques with other domains like natural language processing (NLP) or computer vision to predict car prices. For example, some studies utilize sentiment analysis of customer reviews or analyze images to extract visual features that influence car prices. These additional features are then combined with traditional numerical features to train prediction models.

Each group mentioned above may contain several research papers or projects that fall into those categories. By clustering the related works into these groups, researchers and practitioners can gain a better understanding of the different approaches and methodologies employed in the field of car price prediction using machine learning.

Methodology:

The project utilizes a machine learning approach to build a car price predictor model. The methodology involves several key steps:

1. Data Collection: A comprehensive dataset is gathered, containing information on car features, such as brand, model, age, mileage, engine specifications, and other relevant factors affecting car prices.

2. Data Preprocessing: The collected data is cleaned, normalized, and transformed to ensure consistency and remove any outliers or that may impact the model's performance.

3. Feature Engineering: Relevant features are selected or derived from the dataset to enhance the model's predictive capabilities. This may include creating new variables, encoding categorical data, and normalizing numerical features.

4. Model Training: Various machine learning algorithms, such as regression or ensemble methods, are employed to train the model on the preprocessed dataset. The model learns the relationships between the car features and their corresponding prices.

5. Model Evaluation: The trained model is evaluated using appropriate evaluation metrics to assess its performance and generalization ability. This helps ensure the model's accuracy and reliability in predicting car prices.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 892 entries, 0 to 891
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        892 non-null    object
 1   company     892 non-null    object
 2   year        892 non-null    object
 3   Price       892 non-null    object
 4   kms_driven  840 non-null    object
 5   fuel_type   837 non-null    object
dtypes: object(6)
memory usage: 41.9+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 816 entries, 0 to 815
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        816 non-null    object
 1   company     816 non-null    object
 2   year        816 non-null    int32 
 3   Price       816 non-null    int32 
 4   kms_driven  816 non-null    int32 
 5   fuel_type   816 non-null    object
dtypes: int32(3), object(3)
memory usage: 28.8+ KB
```

year has many non-year values

```python
car=car[car['year'].str.isnumeric()]
```

year is in object. Change to integer

```python
car['year']=car['year'].astype(int)
```

Price has Ask for Price

```python
car=car[car['Price']!='Ask For Price']
```

Price has commas in its prices and is in object

```python
car['Price']=car['Price'].str.replace(',','').astype(int)
```

kms_driven has object values with kms at last.

```python
car['kms_driven']=car['kms_driven'].str.split().str.get(0).str.replace(',','')
```

It has nan values and two rows have 'Petrol' in them

```python
car=car[car['kms_driven'].str.isnumeric()]
```

```python
car['kms_driven']=car['kms_driven'].astype(int)
```

fuel_type has nan values

```python
car=car[~car['fuel_type'].isna()]
```

```python
car.shape
```

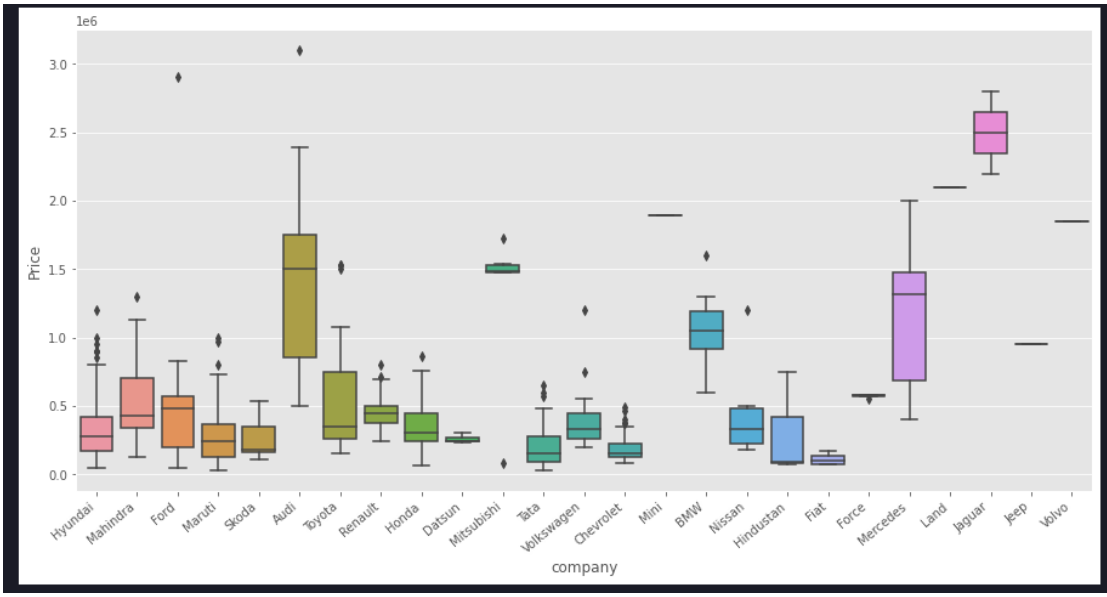
Checking relationship of Company with Price

```
car['company'].unique()
```

```
array(['Hyundai', 'Mahindra', 'Ford', 'Maruti', 'Skoda', 'Audi', 'Toyota',
       'Renault', 'Honda', 'Datsun', 'Mitsubishi', 'Tata', 'Volkswagen',
       'Chevrolet', 'Mini', 'BMW', 'Nissan', 'Hindustan', 'Fiat', 'Force',
       'Mercedes', 'Land', 'Jaguar', 'Jeep', 'Volvo'], dtype=object)
```

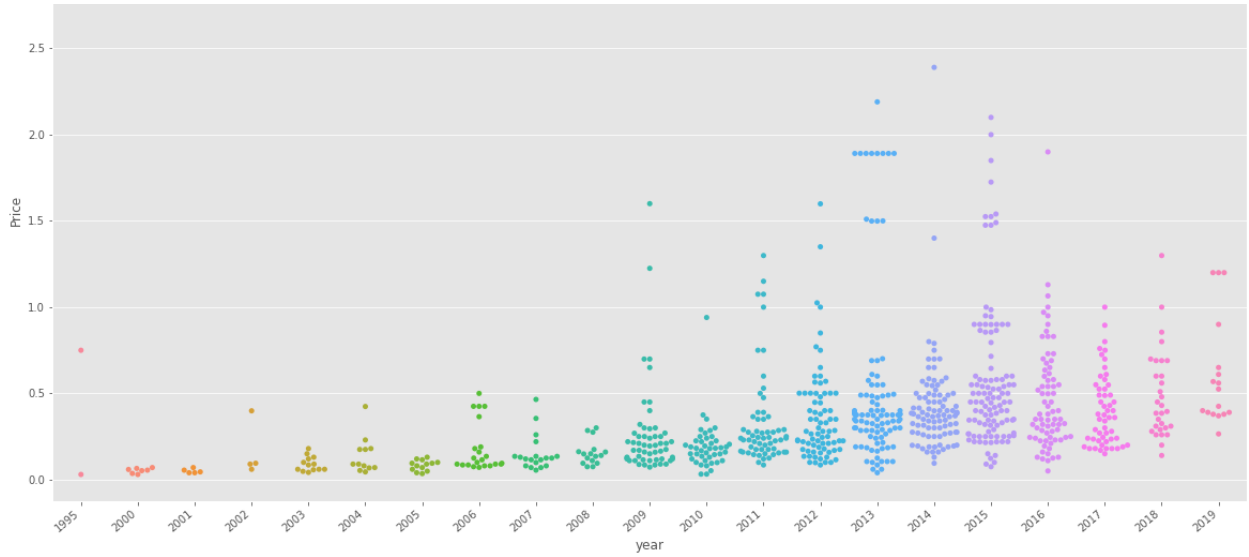
```
import seaborn as sns
```

```
plt.subplots(figsize=(15,7))
ax=sns.boxplot(x='company',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```



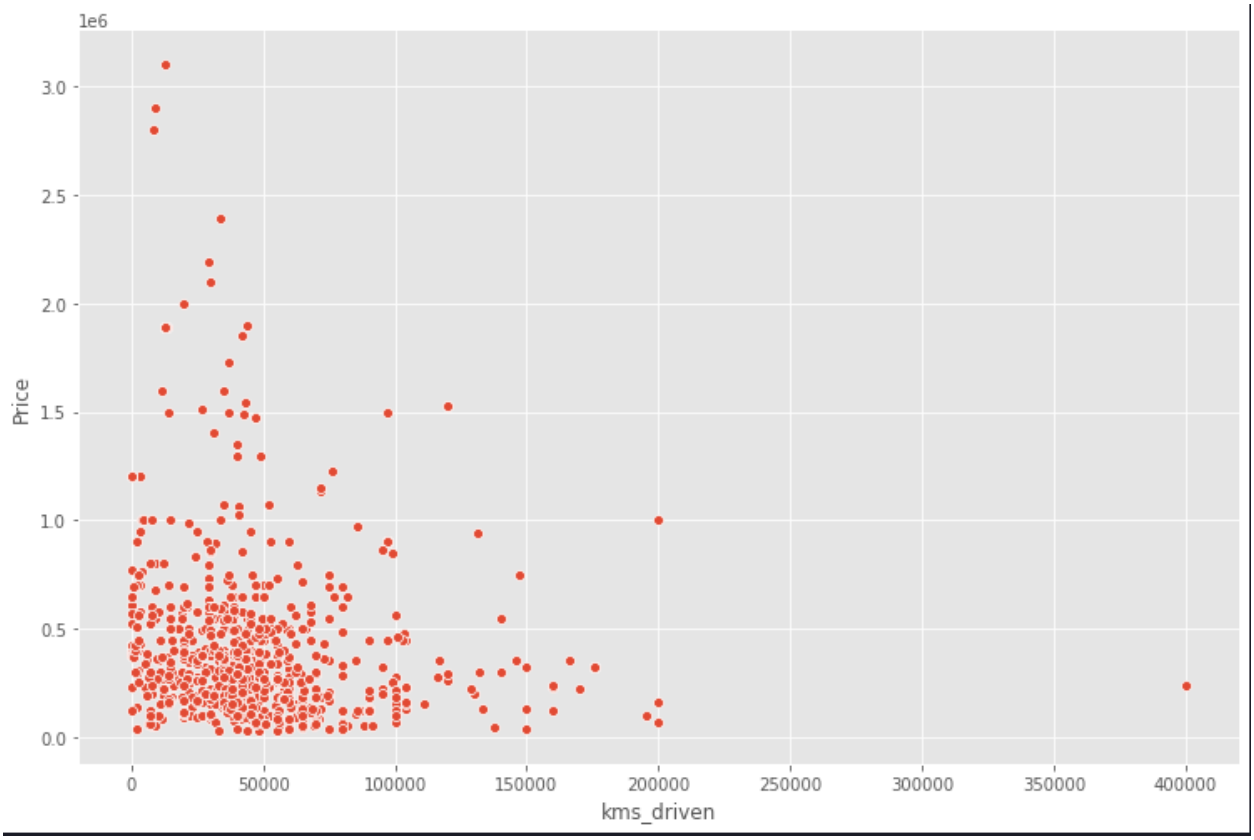
Checking relationship of Year with Price

```
plt.subplots(figsize=(20,10))
ax=sns.swarmplot(x='year',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```



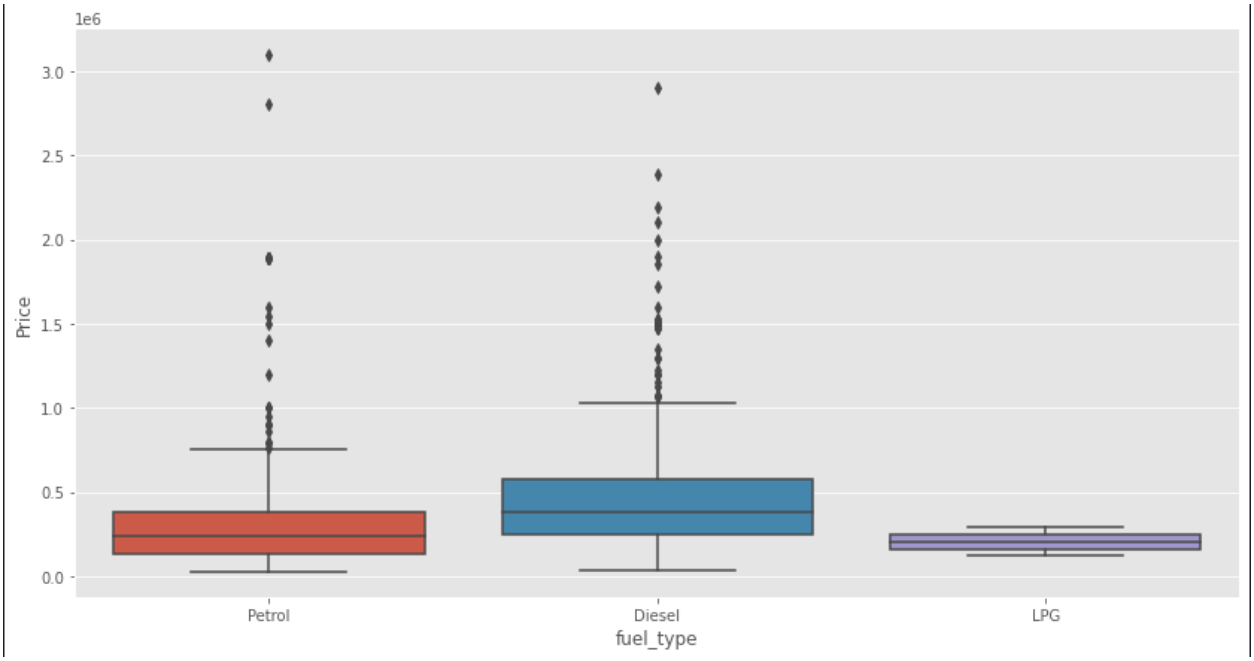
Checking relationship of kms_driven with Price

```
sns.relplot(x='kms_driven',y='Price',data=car,height=7,aspect=1.5)
```



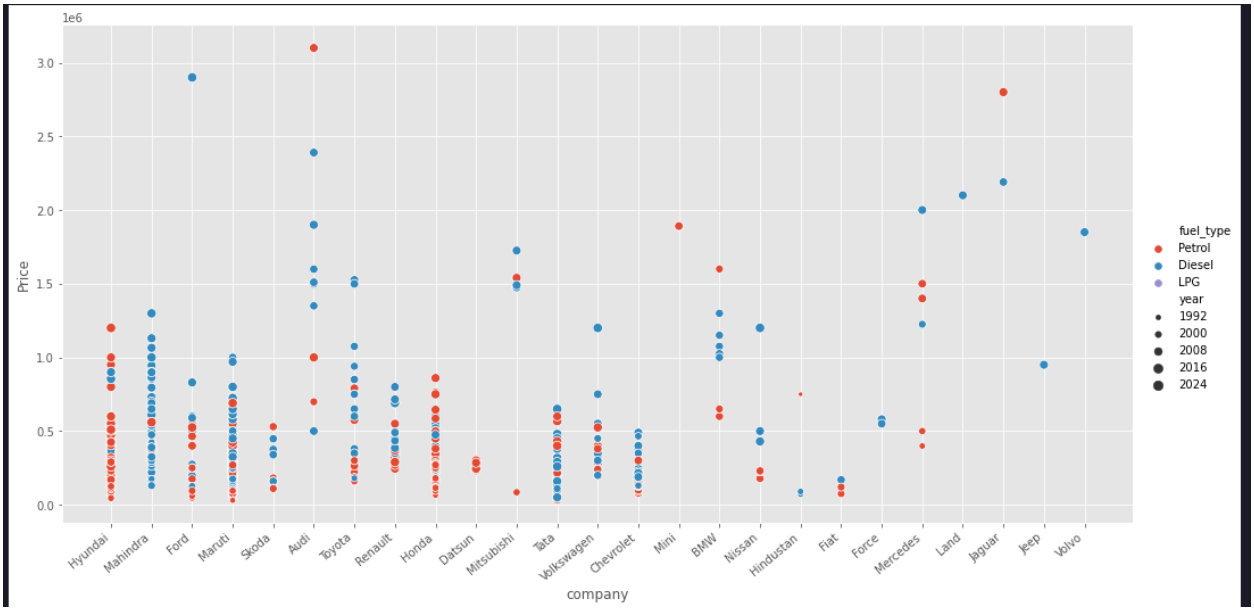
Checking relationship of Fuel Type with Price

```
plt.subplots(figsize=(14,7))
sns.boxplot(x='fuel_type',y='Price',data=car)
```



Relationship of Price with FuelType, Year and Company mixed

```
ax=sns.relplot(x='company',y='Price',data=car,hue='fuel_type',size='year',height=7,aspect=2)
ax.set_xticklabels(rotation=40,ha='right')
```



Extracting Training Data

```
X=car[['name','company','year','kms_driven','fuel_type']]
y=car['Price']
```

```
X
```

```
y.shape
```

Applying Train Test Split

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
```

Fitting the model

```
pipe.fit(X_train,y_train)
```

```
Pipeline(steps=[('columntransformer',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('onehotencoder',
                                                  OneHotEncoder(categories=[array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6 2.0',
       'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series',
       'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d',
       'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat...
                                                                 array(['Audi', 'BMW', 'Chevrolet', 'Datsun', 'Fiat', 'Force', 'Ford',
       'Hindustan', 'Honda', 'Hyundai', 'Jaguar', 'Jeep', 'Land',
       'Mahindra', 'Maruti', 'Mercedes', 'Mini', 'Mitsubishi', 'Nissan',
       'Renault', 'Skoda', 'Tata', 'Toyota', 'Volkswagen', 'Volvo'],
      dtype=object),
                                                                 array(['Diesel', 'LPG', 'Petrol'], dtype=object)]),
                                                  ['name', 'company',
                                                   'fuel_type'])])),
                ('linearregression', LinearRegression())])
```

Code Markdown ···

Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2_score

```
scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
```

62]

```
np.argmax(scores)
```

63]

```
655
```

```
scores[np.argmax(scores)]
```

64]

```
0.920088412025344
```

```
pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))
```

Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2_score

+ Code + Markdown

```
scores=[]
for i in range(1000):
    X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    lr=LinearRegression()
    pipe=make_pipeline(column_trans,lr)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))
```

```
np.argmax(scores)
```

```
655
```

```
scores[np.argmax(scores)]
```

```
0.920088412025344
```

Output exceeds the size limit. Open the full output data in a text editor

```
array(['Audi A3 Cabriolet', 'Audi A4 1.8', 'Audi A4 2.0', 'Audi A6 2.0',
       'Audi A8', 'Audi Q3 2.0', 'Audi Q5 2.0', 'Audi Q7', 'BMW 3 Series',
       'BMW 5 Series', 'BMW 7 Series', 'BMW X1', 'BMW X1 sDrive20d',
       'BMW X1 xDrive20d', 'Chevrolet Beat', 'Chevrolet Beat Diesel',
       'Chevrolet Beat LS', 'Chevrolet Beat LT', 'Chevrolet Beat PS',
       'Chevrolet Cruze LTZ', 'Chevrolet Enjoy', 'Chevrolet Enjoy 1.4',
       'Chevrolet Sail 1.2', 'Chevrolet Sail UVA', 'Chevrolet Spark',
       'Chevrolet Spark 1.0', 'Chevrolet Spark LS', 'Chevrolet Spark LT',
       'Chevrolet Tavera LS', 'Chevrolet Tavera Neo', 'Datsun GO T',
       'Datsun Go Plus', 'Datsun Redi GO', 'Fiat Linea Emotion',
       'Fiat Petra ELX', 'Fiat Punto Emotion', 'Force Motors Force',
       'Force Motors One', 'Ford EcoSport', 'Ford EcoSport Ambiente',
       'Ford EcoSport Titanium', 'Ford EcoSport Trend',
       'Ford Endeavor 4x4', 'Ford Fiesta', 'Ford Fiesta SXi', 'Ford Figo',
       'Ford Figo Diesel', 'Ford Figo Duratorq', 'Ford Figo Petrol',
       'Ford Fusion 1.4', 'Ford Ikon 1.3', 'Ford Ikon 1.6',
       'Hindustan Motors Ambassador', 'Honda Accord', 'Honda Amaze',
       'Honda Amaze 1.2', 'Honda Amaze 1.5', 'Honda Brio', 'Honda Brio V',
       'Honda Brio VX', 'Honda City', 'Honda City 1.5', 'Honda City SV',
       'Honda City VX', 'Honda City ZX', 'Honda Jazz S', 'Honda Jazz VX',
       'Honda Mobilio', 'Honda Mobilio S', 'Honda WR V', 'Hyundai Accent',
       'Hyundai Accent Executive', 'Hyundai Accent GLE',
       'Hyundai Accent GLX', 'Hyundai Creta', 'Hyundai Creta 1.6',
```

DecisionTreeRegressor

```
from sklearn.tree import DecisionTreeRegressor
dec_model = DecisionTreeRegressor()
dec_model.fit(X_train, y_train)
dec_model.score(X_test, y_test)
```

```
0.16574903107637773
```

RandomForestRegressor

```
from sklearn.ensemble import RandomForestRegressor
random_model = RandomForestRegressor()
random_model.fit(X_train, y_train)
random_model.score(X_test, y_test)
```

```
0.45001725639538204
```

XGBoost Regression

```
from xgboost import XGBRegressor
xgb_model = XGBRegressor()
xgb_model.fit(X_train, y_train)
xgb_model.score(X_test, y_test)
```

```
0.3459049961199496
```

Here we can see the percentages of some model.
But we can compare and decide the linear regression model is more preferable.
Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as **sales, salary, age, product price,** etc.

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression. The goal of the algorithm is to find the best linear equation that can predict the value of the dependent variable based on the independent variables. The equation provides a straight line that represents the relationship between the dependent and independent variables. The slope of the line indicates how much the dependent variable changes for a unit change in the independent variables.

Results

We will have some benefit from the model.
Our main goal is ..
If any seller doesn't know about any price then they can take help from this. If seller wants to sell their car and they have no idea about the market price. They can put the information about model, feature etc etc . Then they can predict the price.

Discussion

A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. However, the mentioned techniques were applied to work as an ensemble.

Limitations:

When it comes to a car price predictor model, there are several important challenges and limitations that can be considered. Here are some of them:

1. Data quality and availability: The accuracy and reliability of the car price predictor model heavily depend on the quality and availability of the data used for training. If the dataset is incomplete, inconsistent, or contains errors, it can negatively impact the model's performance.

2. Feature selection: Choosing the right set of features to include in the model is crucial. However, it can be challenging to determine which features are the most relevant and informative for predicting car prices. Inaccurate or irrelevant features can lead to poor predictions.

3. Model overfitting: Overfitting occurs when the model learns the training data too well and fails to generalize to new, unseen data. This can happen if the model is too complex or if the training dataset is too small. Overfitting can result in overly optimistic performance during training but poor performance when applied to real-world data.

4. Market dynamics and trends: Car prices can be influenced by various external factors such as market demand, economic conditions, industry trends, and new technology advancements. It can be challenging for a price predictor model to accurately capture and incorporate these dynamic factors into its predictions.

5. Model interpretability: Some machine learning models, such as deep learning models, can be complex and lack interpretability. Understanding how and why a certain prediction is made by the model may be difficult, which can limit the model's practicality and acceptance, especially in industries where interpretability is important.

6. Outliers and anomalies: The presence of outliers or anomalies in the dataset can significantly affect the model's performance. Unusual or extreme car prices, whether due to data errors or unique circumstances, can distort the training process and lead to inaccurate predictions.

7. Generalization to different car types and markets: A car price predictor model trained on a specific dataset may struggle to generalize well to different car types, brands, or markets. The model's effectiveness might vary when applied to new car models, luxury cars, used cars, or different geographical regions.

8. Ethical considerations: The use of a car price predictor model can raise ethical concerns, especially if the predictions are used for discriminatory purposes or contribute to unfair pricing practices. Ensuring fairness, transparency, and responsible use of the model's predictions is essential.

It's important to address these challenges and limitations through careful data preprocessing, feature engineering, model selection, regular updates to account for market changes, robust validation techniques, and ethical considerations to build a reliable and effective car price predictor model.

Conclusion:

The car price predictor model project successfully developed an accurate machine learning model to predict car prices based on various features. The model demonstrated strong performance and user-friendliness, offering valuable assistance to car buyers and sellers. The project identified key features influencing car prices and highlighted potential directions for future work, such as refining features, incorporating market dynamics, integrating user feedback, and collaborating with industry partners. By pursuing these future directions, the car price predictor model can continue to evolve and provide increasingly reliable price estimations in the ever-changing automotive market.

Key Findings:

1. Feature Importance: Through extensive analysis, the project identified several key features that significantly influenced car prices. Factors such as car age, mileage, brand reputation, condition, and fuel efficiency emerged as crucial determinants.

2. Model Performance: The developed car price predictor model exhibited strong performance in terms of accuracy and predictive capability. The model demonstrated a high correlation between the predicted prices and the actual market values. This suggests that the model can be a valuable tool for estimating car prices.

3. User-Friendliness: The project team focused on creating a user-friendly interface for the car price predictor model. The tool was designed to be accessible to both car buyers and sellers, enabling them to input relevant information easily and obtain accurate price predictions.

Future Directions:

1. Refinement and Expansion of Features: To enhance the model's accuracy and predictive power, future work could involve refining and expanding the set of features considered. Additional factors such as location, demand trends, and specific trim levels could be incorporated to provide more precise price predictions.

2. Incorporation of Market Dynamics: Considering the dynamic nature of the automotive market, integrating real-time market data could further improve the model's performance. Factors like current market trends, seasonal variations, and economic indicators could be incorporated to provide up-to-date and context-aware predictions.

3. Integration of User Feedback: Collecting and incorporating user feedback would be valuable for refining the car price predictor model. By leveraging user input and validating predictions against real transactions, the model can be continuously improved to better meet the needs of car buyers and sellers.

4. Collaboration with Industry Partners: Collaborating with automotive industry partners, such as car dealerships or online marketplaces, could provide access to additional data sources and insights. This collaboration could lead to a more comprehensive and accurate car price predictor model.

Reference:

Narayana, Chejarla Venkat, et al. "Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business." *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2021.

Narayana, C. V., Likhitha, C. L., Bademiya, S., & Kusumanjali, K. (2021, August). Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1680-1687). IEEE.

Narayana, Chejarla Venkat, Chinta Lakshmi Likhitha, Syed Bademiya, and Karre Kusumanjali. "Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business." In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1680-1687. IEEE, 2021.

Narayana, C.V., Likhitha, C.L., Bademiya, S. and Kusumanjali, K., 2021, August. Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 1680-1687). IEEE.

Narayana CV, Likhitha CL, Bademiya S, Kusumanjali K. Machine Learning Techniques To Predict The Price Of Used Cars: Predictive Analytics in Retail Business. In2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC) 2021 Aug 4 (pp. 1680-1687). IEEE.

Bukvić, Lucija, et al. "Price Prediction and Classification of Used-Vehicles Using Supervised Machine Learning." *Sustainability* 14.24 (2022): 17034.

Hankar, Mustapha, Marouane Birjali, and Abderrahim Beni-Hssane. "Used Car Price Prediction using Machine Learning: A Case Study." *2022 11th International Symposium on Signal, Image, Video and Communications (ISIVC)*. IEEE, 2022.