Car Price Predictor

using

Machine Learning

Internship Project Report

Submitted to

MSME - Technology Centre

Tool Room & Training Centre, Guwahati

Submitted by

Deven Malla

Department of Information Technology

School of Engineering & Technology - Nagaland University (SET-NU), Kohima

BONAFIDE CERTIFICATE

This serves to certify that the project titled "**Car Price Predictor using Machine Learning**" submitted to **Tool Room & Training Centre**, Guwahati, represents Deven Malla's authentic work from the Department of Information Technology at the School of Engineering & Technology, Nagaland University, under my supervision from 16-07-2024 to 23-07-2024.

Dhruba Baishya

CS/IT Faculty

Place: Guwahati

Date: 24-07-2024

DECLARATION BY AUTHOR

I at this moment affirm that the following report is my original work and does not contain any plagiarised material. Any information obtained from external sources has been appropriately acknowledged. I take full responsibility for any instances of plagiarism that may be discovered within the report.

Deven Malla

Dept. IT (SET-NU)

Place: Guwahati

Date: 24-07-2024

ABSTRACT

This project encompasses the development of a machine learning model utilizing linear regression to predict car prices. The model is trained on data extracted from reputable car listings, encompassing attributes such as car mileage, engine capacity, and year of manufacture. Key steps in the dataset preparation involve handling missing values and scaling features to enhance the model's reliability.

Following rigorous practice and iterative adjustments, the model begins to demonstrate efficacy in predicting car prices. This endeavour underscores the practical application of machine learning in the automotive industry, placing particular emphasis on mitigating bias from stakeholders and the modelling process to facilitate informed decision-making about pricing strategies.

TABLE OF CONTENTS

INTRODUCTION

The process of accurately assessing the value of a used car in today's dynamic automotive market can be quite intricate. Traditional valuation methods often rely on generalized metrics and human expertise, which may not encompass all the nuanced factors that influence a vehicle's worth. However, machine learning has the potential to revolutionize this process through the use of sophisticated, data-driven models.

The "**Car Price Predictor using Machine Learning**" project aims to develop a model for estimating used car prices based on various attributes. By analyzing historical data, the model provides reliable price predictions that reflect current market conditions and individual vehicle characteristics.

1. Dataset:

- The dataset includes attributes such as vehicle name, year, selling price, kilometres driven, fuel type, seller type, transmission, ownership history, mileage, engine capacity, max power, and seating capacity.

2. Pre-Requisite Information:

- Understanding the data and its attributes is crucial for building an effective model.

3. Methodology:

- Data Pre-processing: Cleaning and preparing the dataset to handle missing values and remove duplicates.

- Exploratory Data Analysis (EDA): Gaining insights by analyzing variable distributions, identifying correlations, and detecting anomalies.

- Train-test split: Dividing the dataset into training and testing sets to validate the model.

- Model Selection: Comparing various machine learning algorithms such as Linear Regression and decision trees to choose the most accurate model.

- Model Deployment: Creating a user-friendly application for quick price predictions.

This project illustrates the potential of machine learning in addressing real-world problems, to benefit both sellers and buyers by providing accurate car valuations and fostering a more efficient and equitable used car market.

DATASET

The dataset used for this challenge is a comprehensive collection of information on used automobiles, which incorporates a ramification of functions necessary for predicting car costs. The dataset contains 6,718 entries and 12 columns, protecting exclusive aspects of every car. The capabilities blanketed are:

1. Name:

- The model and variant of the car, e.g., 'Maruti Swift Dzire VDI'.

2. Year:

- The manufacturing year of the car, ranges from 1994 to 2020.

3. Selling Price:

- The selling price of the car, varies significantly across different models and conditions.

4. Kilometers Driven:

- The total distance the car has been driven in kilometres.

5. Fuel Type:

- The type of fuel used by the car, such as Diesel, Petrol, LPG, and CNG.

6. Seller Type:

- The category of the seller, including Individual, Dealer, and Trustmark Dealer.

7. Transmission:

- The type of transmission system in the car, either Manual or Automatic.

8. Owner:

- The number of previous owners of the car.

9. Mileage:

- The mileage of the car in kilometres per litre (km).

10. Engine:

- The engine capacity of the car in cubic centimetres (CC).

11. Max Power:

- The maximum power produced by the car's engine, measured in brake horsepower (bhp).

12. Seats:

- The seating capacity of the car.

PRE-REQUISITE INFORMATION

Before embarking on the creation of a Car Price Predictor using Machine Learning, it is essential to have a foundational understanding of several key concepts and tools:

1. Machine Learning Basics:

- Supervised Learning: Understand the concept of supervised learning, where the model is trained on a labelled dataset, which includes both input features and the corresponding output labels.

- Regression Algorithms: Familiarity with regression algorithms, particularly Linear Regression, as it is used to predict continuous values.

2. Programming Knowledge:

- Python: Proficiency in Python programming is crucial as it is the primary language used for data analysis and machine learning model development.

- Pandas and NumPy: Knowledge of these libraries for data manipulation and numerical operations.

3. Data Preprocessing:

- Handling Missing Values: Techniques to handle missing data, such as imputation or removal, are vital to prepare the dataset for analysis.

- Feature Engineering: Skills in creating new features from existing data to improve model performance.

4. Exploratory Data Analysis (EDA):

- Data Visualization: Ability to use libraries like Matplotlib and Seaborn to visualize data and understand patterns, trends, and outliers.

- Statistical Analysis: Understanding basic statistical concepts to summarize and analyze data distributions.

5. Model Development:

- Train-Test Split: Knowledge of splitting the dataset into training and testing sets to evaluate model performance.

- Model Evaluation Metrics: Understanding metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared to evaluate the accuracy of the model.

6. Model Deployment:

- Serialization Techniques: Familiarity with techniques like Pickle for saving and loading trained models for future use.

- Web Frameworks: Comprehensive understanding of Streamlit for deploying the project.

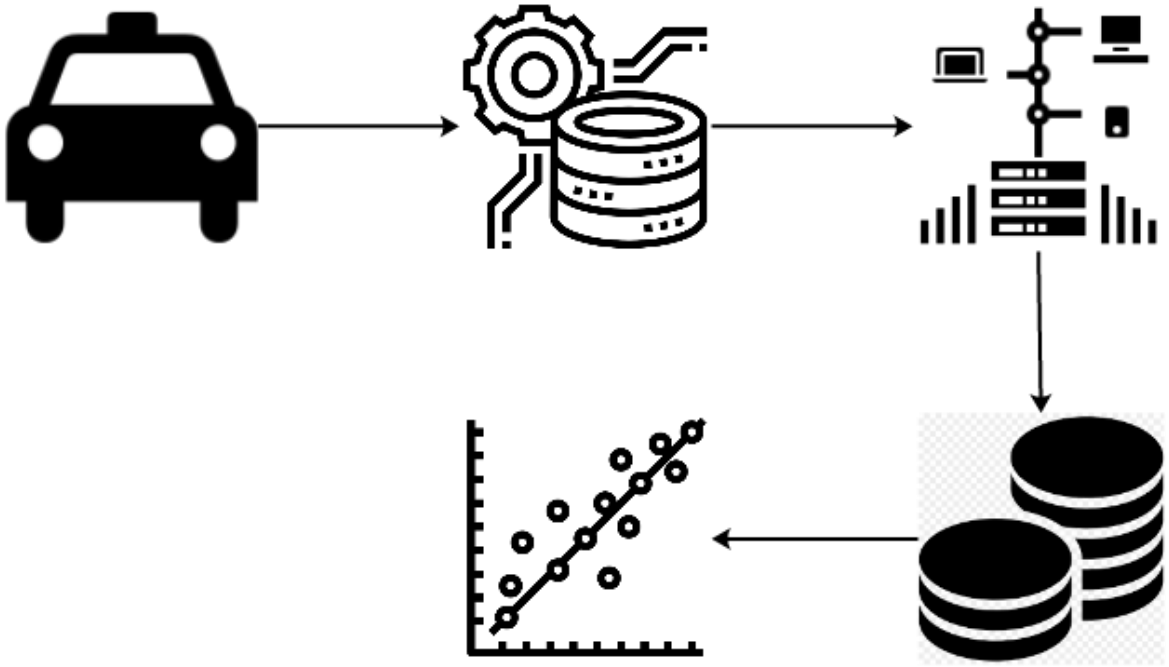
These pre-requisite concepts and tools form the foundation for successfully building and deploying a machine-learning model to predict car prices.

Installation:

```
# Create a virtual environment (optional but recommended)
python -m venv car_price_env

# Activate the virtual environment
# On Windows
car_price_env\Scripts\activate
# On macOS and Linux
source car_price_env/bin/activate

# Upgrade pip
pip install --upgrade pip

# Install essential libraries
pip install pandas numpy matplotlib seaborn scikit-learn

# Install Streamlit for deployment
pip install streamlit

# Install Pickle (part of Python standard library, no need for installation)
# Import it directly in your script
# import pickle
# Install IPython (name of the Python backend (aka kernel))

# If you need Jupyter Notebook for development and testing
```

```
pip install notebook

# Install other libraries as needed for your project

# Running the Streamlit Application
streamlit run your_script.py
```

METHODOLOGY



I. DATA PRE-PROCESSING

Data pre-processing is a crucial step in preparing the dataset for analysis and modelling. The steps involved in this phase include data cleaning, handling missing values, and transforming data to be suitable for the machine learning model.

1. **Loading the Data:** The dataset is loaded into a pandas DataFrame from a CSV file.

```
import pandas as pd
cars_data = pd.read_csv('Cardetails.csv')
```

2. **Initial Data Inspection:** Inspect the first few rows and the structure of the dataset to understand its content.

```
print(cars_data.head())
print(cars_data.info())
```

3. **Handling Missing Values:** Identify and handle missing values in the dataset. Columns with missing values are dropped if they have a significant number of missing entries.

```
cars_data.isnull().sum()
cars_data.dropna(inplace=True)
```

4. **Removing Unnecessary Columns:** Drop columns that are not necessary for the prediction model, such as 'torque'.

```
cars_data.drop(columns=['torque'], inplace=True)
```

5. **Removing Duplicates:** Check for and remove duplicate entries to ensure data integrity.

```
cars_data.duplicated().sum()
cars_data.drop_duplicates(inplace=True)
```

6. **Data Type Conversion:** Ensure that all columns are of the appropriate data type for analysis and modelling.

```
cars_data['year'] = cars_data['year'].astype(int)
cars_data['selling_price'] = cars_data['selling_price'].astype(int)
```

II. EXPLORATORY DATA ANALYSIS (EDA)

EDA involves analyzing the dataset to summarize its main characteristics, often using visual methods. This step helps in understanding the data distribution, identifying patterns, and spotting anomalies.

1. **Descriptive Statistics:** Compute summary statistics to understand the distribution of numerical variables.

```
print(cars_data.describe())
```

2. **Distribution Analysis:** Plot histograms and box plots for numerical features to visualize their distributions.

```
cars_data.hist(bins=50, figsize=(20,15))
plt.show()
```

3. **Correlation Analysis:** Calculate and visualize the correlation matrix to identify relationships between variables.

```
corr_matrix = cars_data.corr()
sns.heatmap(corr_matrix, annot=True)
plt.show()
```

4. **Category Distribution:** Analyze the distribution of categorical features using bar plots.

```
sns.countplot(x='fuel', data=cars_data)
plt.show()
```

III. TRAIN-TEST SPLIT

Splitting the dataset into training and testing sets is essential to evaluate the performance of the machine learning model on unseen data.

1. **Defining Features and Target:** Select the features (independent variables) and the target (dependent variable).

```
X = cars_data.drop(columns=['selling_price'])
y = cars_data['selling_price']
```

2. **Encoding Categorical Variables:** Convert categorical variables into numeric format using one-hot encoding.

```
X = pd.get_dummies(X, drop_first=True)
```

3. **Splitting the Data:** Split the data into training and testing sets.

```
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,
random_state=42)
```

IV. MODEL SELECTION

Selecting and training the appropriate machine learning model to predict car prices is a critical step. Various models can be evaluated based on their performance metrics.

1. **Model Selection:** Start with simple models like Linear Regression and then explore more complex models like Random Forest and Gradient Boosting.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

2. **Model Evaluation:** Evaluate the model's performance using metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
y_pred = model.predict(X_test)
print('MAE:', mean_absolute_error(y_test, y_pred))
print('MSE:', mean_squared_error(y_test, y_pred))
print('R-squared:', r2_score(y_test, y_pred))
```

3. **Hyperparameter Tuning:** Use techniques like GridSearchCV or RandomizedSearchCV to optimize hyperparameters of more complex models.

```
from sklearn.model_selection import GridSearchCV
```

```
param_grid = {'n_estimators': [100, 200], 'max_features': ['auto',
'sqrt']}
grid_search = GridSearchCV(estimator=RandomForestRegressor(),
param_grid=param_grid, cv=5)
grid_search.fit(X_train, y_train)
```

V. DEPLOYMENT

The final step is to deploy the trained model for real-world usage. This involves saving the model and creating an interface for users to interact with it.

1. **Model Serialization:** Save the trained model using joblib or pickle.

```
import joblib
joblib.dump(model, 'car_price_predictor.pkl')
```

2. **Deployment:** Using Streamlit, we deployed the model as a user-friendly web application, making real-time predictions accessible.

CONCLUSION

The "Car Price Predictor" project demonstrates a comprehensive approach to developing a machine learning model for predicting car prices based on various attributes.

Data pre-processing involved handling missing values, removing duplicates, and transforming data, ensuring the dataset was clean and consistent. This step was crucial for accurate and reliable predictions.

EDA provided valuable insights into the dataset, helping us understand the distribution of variables and identify key patterns. This guided feature selection, enhancing the model's predictive power.

Splitting the data into training and testing sets allowed us to evaluate the model's performance effectively. This approach ensured the model's generalizability to unseen data and prevented overfitting.

We evaluated several machine learning algorithms, including Linear Regression and Random Forest. By optimizing hyperparameters, we improved model accuracy and robustness. The best-performing model was selected based on metrics like Mean Absolute Error (MAE) and R-squared.

The final model was deployed using Flask, creating an API for real-time price predictions. This demonstrated the

model's practical applicability and provided a user-friendly interface for end-users.

KEY TAKEAWAYS

1. Data Quality: Clean, well-prepared data is essential for accurate models.
2. Feature Selection: EDA helps in selecting relevant features, and improving performance.
3. Model Evaluation: Proper data splitting ensures reliable performance evaluation.
4. Hyperparameter Tuning: Optimizing parameters enhances model accuracy.
5. Deployment: Accessible deployment ensures practical utility.

The "Car Price Predictor" project showcases a structured approach to building, evaluating, and deploying a robust machine learning model ready for real-world application.

REFERENCE

1. Dataset
   https://drive.google.com/file/d/1UIj8rOmDJn4UgqDTJezMNfzr7jh_behL/view?pli=1
2. Requirements.txt
   https://stackoverflow.com/questions/7225900/how-can-i-install-packages-using-pip-according-to-the-requirements-txt-file-from
3. Streamlit
   https://docs.kanaries.net/topics/Streamlit/streamlit-vscode
4. Github
   https://github.com/deven-malla/car_price_prediction_project_using_machine_learning.git