

**TRIBHUVAN UNIVERSITY**

**National College of Computer Studies (NCCS­­­)**

Paknajol, Kathmandu, Nepal

Project Report

Python Project – Machine Learning

Car Price Regression

Submitted By: Submitted To:

Amodh Nepal Mausam Rajbanshi

BIM 5th semester

Section: B

Roll no.: 2

# Acknowledgment

I would like to acknowledge the use of data regression analysis as a fundamental component of this report on car prices. Data regression allowed us to analyze and model the relationship between various independent variables and the dependent variable, namely car prices, using Python programming language. This analysis was crucial in gaining insights into the factors influencing car prices and ultimately enhancing the quality of this report. The regression analysis provided a robust framework for exploring and quantifying the impact of factors such as car attributes, market conditions, and other relevant variables on car prices. It allowed us to develop predictive models that can aid in decision-making processes and offer valuable information for the automotive industry.

I want to express my gratitude to the Python programming community for providing the necessary tools and libraries, such as NumPy, pandas, and scikit-learn, which made it possible to conduct this analysis efficiently and accurately. Additionally, I would like to acknowledge the contribution of open datasets and data sources, which formed the basis of our analysis. In conclusion, data regression analysis using Python was an essential aspect of this report, and its results have greatly enriched our understanding of the factors driving car prices. It is my hope that this acknowledgment recognizes the significance of data regression in our research and its role in producing the insights presented here.

Table of Contents

[Acknowledgment ii](#_Toc144729743)

[Introduction to Machine Learning 1](#_Toc144729744)

[Machine Learning in Python 2](#_Toc144729745)

[Background 3](#_Toc144729746)

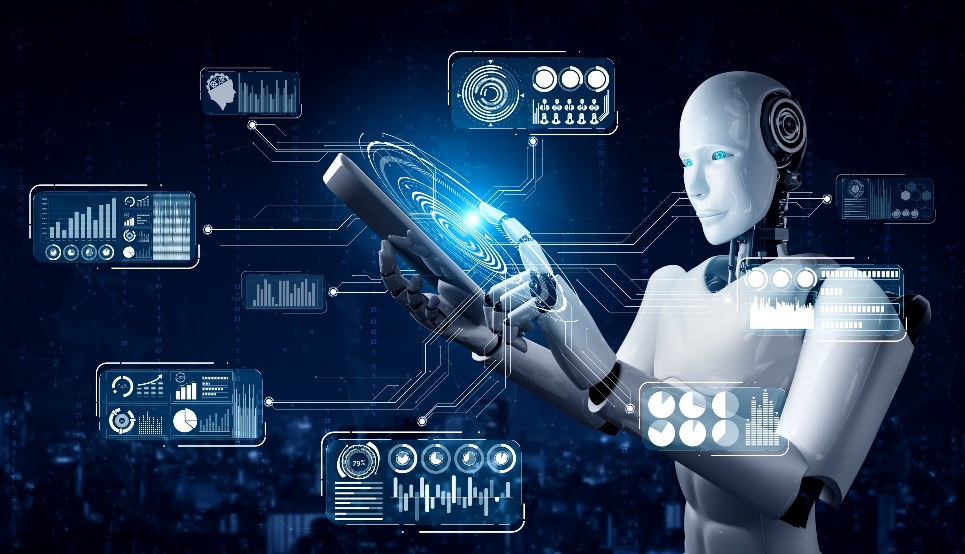
[Objective 5](#_Toc144729747)

[Implementation 6](#_Toc144729748)

[Conclusion 17](#_Toc144729749)

# Introduction to Machine Learning

Machine learning is a subfield of AI that focuses on the development of algorithms and statistical models that allow computer systems to learn and improve their performance on a specific task without being explicitly programmed. It boils down to teaching machines to learn from data and then make predictions or decisions based on that learning.



Here are the key machine learning components and concepts:

* Data: The foundation of machine learning is data. Data is used to teach machine learning models patterns and relationships. This data can be structured (for example, tables and databases) or unstructured (for example, text, images, and audio). The quality and quantity of data frequently have a significant impact on machine learning model performance.
* Training: A machine learning model is trained by exposing it to a dataset containing input data (features) and corresponding target values or labels. By adjusting its internal parameters, the model learns to recognize patterns or relationships in data.
* Features: Features are the variables or attributes that are used to describe the data. The process of selecting, transforming, or creating meaningful features that help the model learn and make accurate predictions is known as feature engineering.
* Model: A machine learning model is a mathematical representation of data patterns or relationships. For various tasks, various models are used, such as linear regression for regression problems, decision trees for classification, and neural networks for complex tasks such as image recognition.
* Algorithm: Machine learning algorithms are mathematical and computational techniques that are used to train models. These algorithms differ according to the type of learning (supervised, unsupervised, or reinforcement) and the task at hand.
* Evaluation: Depending on the task, machine learning models are evaluated using various metrics and techniques such as accuracy, precision, recall, F1-score, or mean squared error.
* Deployment: Machine learning models can be deployed in real-world applications after training and evaluation to make predictions or automate decision-making processes.
* Continuous Learning: Machine learning models can be updated and improved as new data becomes available. This is known as online learning or retraining, and it is critical for maintaining model accuracy over time.

Machine learning is widely used in a variety of domains, including healthcare (diagnosis and prognosis), finance (fraud detection and stock prediction), natural language processing (language translation and sentiment analysis), image recognition (object detection and facial recognition), and many others. It continues to play an important role in advancing technology and solving complex problems.

## Machine Learning in Python

Python is a popular data science programming language due to its simplicity, versatility, and a rich ecosystem of libraries and tools designed specifically for data analysis and machine learning. Here's a quick rundown of how Python is used in data science:

* Data Manipulation and Analysis:
* NumPy: Numpy supports arrays and matrices, making it simple to perform numerical operations on large datasets.
* Pandas: Pandas is a powerful data manipulation and analysis library. It provides data structures such as DataFrames for dealing with structured data.
* Data Visualization:
* Matplotlib: Matplotlib is a popular library for creating static, animated, and interactive plots and charts.
* Seaborn: Seaborn is based on Matplotlib and provides a higher-level interface for creating visually appealing statistical graphics.
* Machine Learning:
* scikit-learn: scikit-learn is a large machine learning library that includes tools for classification, regression, clustering, and model selection.

Python's rich library ecosystem, combined with its readability and ease of use, makes it an excellent choice for data scientists looking to effectively explore, analyze, and model data. Depending on the task, you can use a variety of libraries and tools to achieve your data science objectives.

# Background

In today's data-driven world, businesses and individuals are increasingly relying on data-driven insights to make informed decisions. Predicting car prices is a common task in the automotive industry, as well as for consumers looking to buy or sell cars. One powerful tool for this task is data regression, a statistical technique used to model the relationship between a dependent variable and one or more independent variables.

Steps in Car Price Prediction using Data Regression:

1. Data Collection: Gather a dataset containing information about various cars, including their prices and relevant features.

2. Data Preprocessing: Clean and preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features if necessary.

3. Data Exploration: Explore the dataset through summary statistics, visualizations, and correlations to gain insights into the data.

4. Feature Selection: Identify the most relevant features that are likely to influence car prices. Feature selection helps in building more accurate models.

5. Data Splitting: Split the dataset into a training set and a test set. The training set is used to train the regression model, while the test set is used to evaluate its performance.

6. Model Training: Apply a regression algorithm (e.g., Linear Regression) to the training data to learn the relationship between the independent variables and car prices.

7. Model Evaluation: Evaluate the model's performance using appropriate metrics such as Mean Squared Error (MSE), R-squared, or Root Mean Squared Error (RMSE).

8. Prediction: Use the trained model to make predictions on new, unseen data.

9. Interpretation: Interpret the model's coefficients to understand the impact of each independent variable on car prices.

10. Conclusion: Summarize the findings, discuss the model's accuracy, and provide insights into factors affecting car prices.

This report will follow a structured approach to demonstrate how data regression techniques in Python can be utilized to predict car prices effectively. It will include data preprocessing, model selection, evaluation, and interpretation sections, with the goal of providing a reliable car price prediction model.

# Objective

The primary objective of conducting data regression on car price data using Python for this report is to construct a reliable predictive model. This model should be capable of accurately estimating car prices based on a set of independent variables, thereby facilitating better-informed decisions in the automotive market. Through this analysis, we seek to delve into the relationships that exist between car prices and factors such as car age, mileage, brand, model, engine size, and more. By doing so, we aim to provide valuable insights into what drives car pricing trends and fluctuations. Additionally, this report intends to showcase the power of Python as a data analysis and modeling tool, utilizing libraries like NumPy, pandas, scikit-learn, and Matplotlib for data manipulation, exploration, and visualization. Ultimately, our objective is to empower both consumers and industry professionals with a data-driven tool that enhances their ability to accurately assess car values, contributing to more informed buying and selling decisions in the automotive sector.

# 

# Implementation

In this section, we look at how to use sepal and petal measurements to classify Iris flowers into their respective species—specifically, Iris setosa, Iris virginica, and Iris versicolor. To carry out the classification task, we use Python, a versatile programming language for data analysis and machine learning. The following steps outline the key implementation elements:

1. Data Preparation:

- Import necessary libraries (NumPy, pandas, scikit-learn).

- Load the car price dataset (CSV or Excel).

- Check for and handle missing values, duplicates, and inconsistent data types.

2. Exploratory Data Analysis (EDA):

- Perform EDA to understand the dataset.

- Create summary statistics, visualizations, and correlation matrices.

- Identify influential features that likely affect car prices.

3. Data Splitting:

- Split the dataset into training and testing subsets (e.g., 80/20 or 70/30).

- Ensure randomization to avoid bias.

4. Model Selection:

- Choose an appropriate regression model (e.g., Linear Regression, Ridge, Lasso).

- Initialize the selected regression model.

5. Model Training:

- Fit the regression model to the training data.

- Tune model hyperparameters if needed for better performance.

6. Model Evaluation:

- Assess the model's performance using metrics like MSE, or RMSE.

- Evaluate its generalization on the testing dataset.

7. Predictions:

- Use the trained model to predict car prices on the testing dataset.

- Compare predicted prices to actual prices to gauge model accuracy.

8. Interpretation:

- Analyze feature coefficients to understand their impact on car prices.

- Identify the most influential features affecting price predictions.

9. Model Visualization:

- Create visualizations (e.g., scatter plots, residual plots) to visualize model performance and assumptions.

10. Reporting and Conclusion:

- Summarize findings, including model accuracy and feature insights.

- Discuss practical implications for the automotive industry.

- Address any limitations and suggest areas for future research.

The codes used for the projects are given below:

#Import modules

from sklearn.linear\_model import LinearRegression

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.preprocessing import LabelEncoder

#Read the data set

car\_price=pd.read\_csv("CarPrice\_Assignment.csv")

car\_price.head()

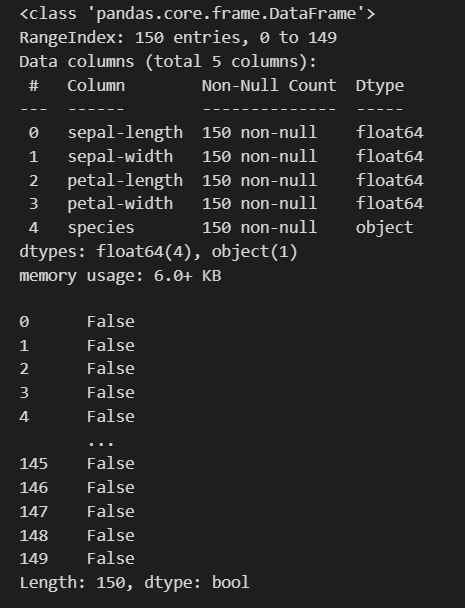
A screenshot of a computer

Description automatically generated

#Describing data set

data.info()

data.columns



data.columns

A computer screen shot of white text

Description automatically generated

#Identify unique column

print(data['CarName'].unique())

print(data['fueltype'].unique())

print(data['aspiration'].unique())

print(data['doornumber'].unique())

print(data['carbody'].unique())

print(data['drivewheel'].unique())

print(data['enginelocation'].unique())

print(data['enginetype'].unique())

print(data['cylindernumber'].unique())

print(data['fuelsystem'].unique())

A screen shot of a computer code

Description automatically generated

#Scatterplot

import matplotlib.pyplot as plt

plt.scatter(default\_arr,y\_test,marker='o',label="Ground truth")

plt.scatter(default\_arr,y\_pred,marker='x',label="Prediction")

plt.legend()

plt.grid()

A screen shot of a graph

Description automatically generated

#Data pre-processing



#Coorelation Matrix

data.corr()

A screen shot of a black and white screen

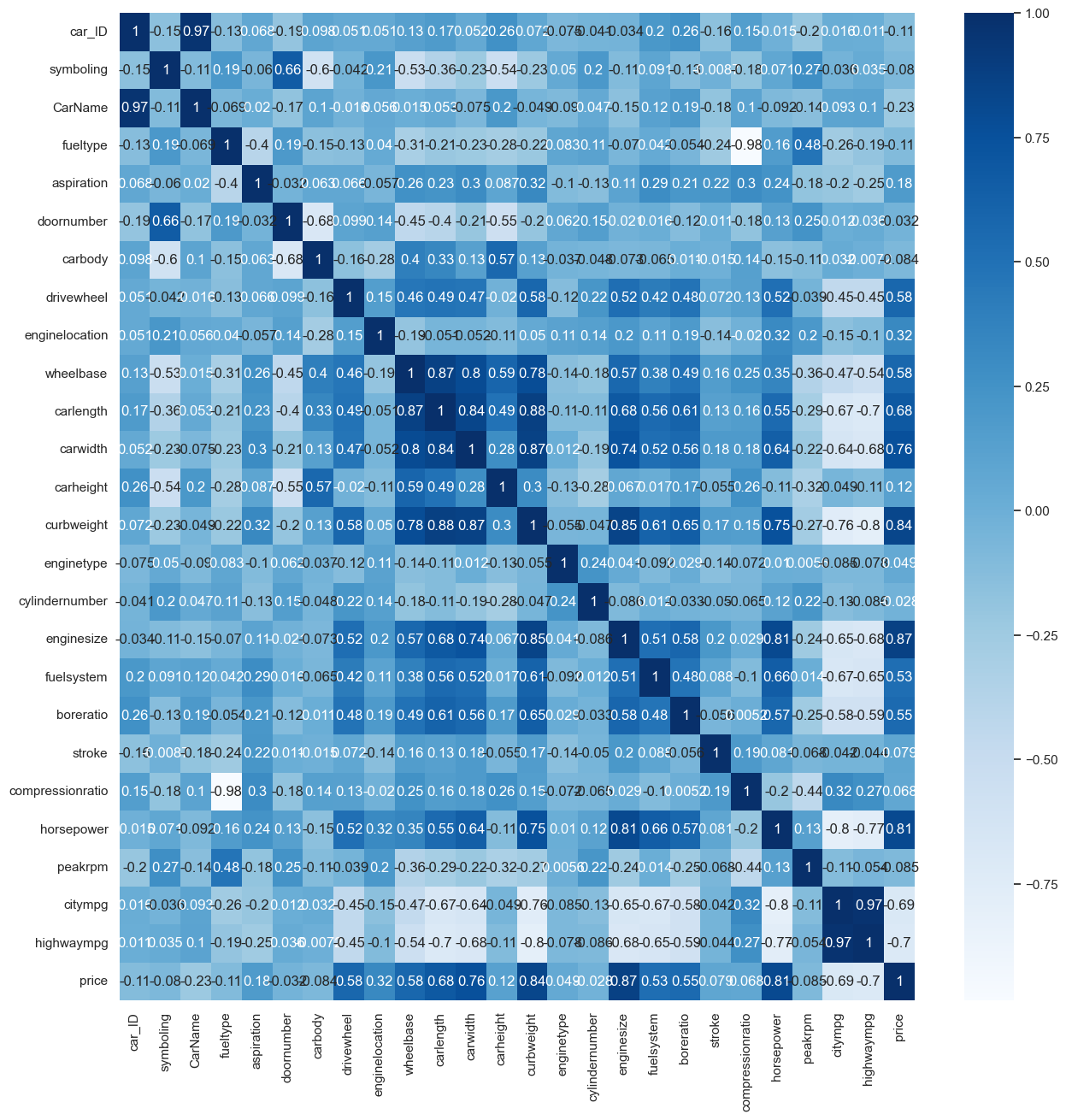
Description automatically generated

A screenshot of a computer screen

Description automatically generated

sns.heatmap(data.corr(), cmap="Blues", annot=True)

sns.set(rc={'figure.figsize':(15,15)})



#Model Training

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=10

)

#train - 70

#test – 30

lr\_model = LinearRegression()

lr\_model.fit(X\_train,y\_train)

A screenshot of a computer

Description automatically generated

# Conclusion

In conclusion, I have found that the use of Python for car price prediction is a powerful and valuable tool in the field of data science and machine learning. Through this report, I have explored various aspects of building a car price prediction model using Python, and I can draw several key takeaways:

* Data Preparation is Crucial: Data preprocessing and feature engineering play a vital role in building an accurate car price prediction model. Cleaning the dataset, handling missing values, and selecting relevant features are essential steps to ensure the model's success.
* Model Selection Matters: Python offers a wide range of machine learning libraries and frameworks, allowing me to choose the most suitable algorithm for my specific dataset. Experimenting with different models and hyperparameters can lead to better predictive performance.
* Evaluation and Validation: Proper evaluation metrics such as Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are critical for assessing the model's accuracy. Additionally, cross-validation helps ensure that the model generalizes well to unseen data.
* Feature Importance: Understanding which features contribute the most to the car price prediction is essential for both model interpretability and potentially improving the model's performance.
* Deployment: After building and fine-tuning the model, it can be deployed into production, allowing me to automate car price predictions and make informed decisions.
* Continuous Improvement: Machine learning models should be monitored over time, and retraining may be necessary as new data becomes available or business requirements change.

In conclusion, Python provides a versatile and accessible platform for developing sophisticated car price prediction models, making it a valuable tool for businesses in the automotive industry and beyond. The successful implementation of such models can lead to improved pricing strategies, enhanced customer experiences, and ultimately, increased profitability. However, it's important to note that the effectiveness of the model heavily depends on the quality of the data and the thoroughness of the data preprocessing and modeling processes. Therefore, a data-centric approach and ongoing model maintenance are critical for achieving accurate and reliable car price predictions