PROJECT REPORT

ON

SPORTS CAR PRICE ANALYSIS AND MODELLING

Project work submitted to

Parvathaneni Brahmayya Siddhartha College of Arts and Science(Autonomous)

for the partial fulfilment of the requirements for the

award of

Degree of Bachelor Science (Statistics)

Done By

B. Hemanth sai

(223619P)

Under the guidance of

Mr. N. Rakesh MSc, Asst Prof

DEPARTMENT OF STATISTICS

April 2025

PARVATHANENI BRAHMAYYA SIDDHARTHA COLLEGE

OF ARTS AND SCIENCE (AUTONOMOUS)

Vijayawada – 520010

DEPARTMENT OF STATISTICS

CERTIFICATE

This is certified that this is a bonafide record of **INTERNSHIP PROJECT** work done in **DEPARTMENT OF STATISTICS** entitled with **SPORTS CAR PRICE ANALYSIS AND MODELLING** done by **B. Hemanth sai(223619P)** for the partial fulfilment of the requirement for the award of the Bachelor of Science and Arts degree as part of the curriculum during the academic year 2024 – 2025.

Date:

Lecturer in charge **Head of the Department**

Internal Examiner **External Examiner**

Date: **Date:**

DECLARATION

We here by declare that the Long-Term Internship project report entitled **SPORTS CAR PRICE ANALYSIS AND MODELLING** submitted by us in the partial fulfilment of the requirement for the award of degree in Bachelor of Science is the record of work originally carried out by from **JANUARY 2025 – APRIL 2025** under able guidance of **Mr. N.RAKESH MSc, Asst Prof, Department of Statistics, P. B. Siddhartha College of Arts & Science, Vijayawada.**

B. HEMANTH SAI

(223619P)

ACKNOWLEDGMENT

I would like to express my profound gratitude to **Mr. N. RAKESH**, **Lecturer in Statistics** for giving guidance, suggestions, throughout provoking discussion and constant encouragement and for giving an opportunity to take up of the project.

I would like to express my profound gratitude to **Mr. G. Chakravarthi**, **Head of the department of Statistics** for giving us able guidance, support, encouragement, excellent suggestions and advices, throughout provoking discussion and constant encouragement from beginning till the end and also for giving opportunity to take up this project.

I would like to express my profound gratitude to **Sri U. Sai Ram, CEO of Codegnan IT Solutions Pvt Limited Vijayawada**, Andhra Pradesh, for giving guidance, Suggestions, throughout provoking discussions and constant encouragement and for giving an opportunity to take up of the project.

I sincerely thank to all the faculty members of Department of Statistics of P.B. Siddhartha College of Arts & Science, in supporting and encouragement in completing of my project.

I sincerely thank our college management do giving us golden opportunity to do this project.

Last but not least, I express my respectable regards to the parents for their encouragement, co-operation throughout the work. I thank one and all helped me in my efforts for completingthe project.

B. HEMANTH SAI

(223619P)

INDEX

CHAPTER – 1

ABOUT CODEGNAN

1.1 INTRODUCTION TO CODEGNAN IT SOLUTIONS:

Codegnan is a leading IT training institute in India, trusted by 24,000+ students and rated as 4.8/5 by more than 2,100 students. At Codegnan, they not only focus on delivering quality education and practical learning to their students but also help them get placed in jobs or internships. Their courses cover a wide range of topics, including front-end, backend, database, cloud deployment, DataAnalytics aptitude, reasoning and soft skills.

Codegnan was founded in 2018 in Vijayawada, India. It has been a trusted IT training institute, providing quality education and practical learning to over 24,000 students. Their courses cover a widerange of topics, including software development, MERN development, Python development, and machine learning. Codegnan aims to empower individuals and help

them get placed in jobs or internships. Codegnan's courses are designed to provide practical skills, real- world examples, and doubt-free learning experiences. They have a strong community of learners and highly qualified mentors to guide you on your learning journey.

Whether you're a beginner or an experienced developer, Codegnan has something to offer for everyone. Codegnan ensures that students understand the flow of work from various

perspectives in a real-time environment. They believe in student-centric methods, allowing learners to access live environments and excel in their careers.

1.2 INTERN AND LEARN WORK:

1.2.1 About

I had the privilege of completing my internship at Codegnan IT Solutions Private Limited, a distinguished software training institute situated in the vibrant city of Vijayawada, Andhra Pradesh. Nestled in close proximity to P.B. Siddhartha College in Moghalrajpuram, Codegnan stands as a prominent hub for IT education, offering comprehensive training programs to

aspiring professionals. The institute's address is located at H.No 40-5-19/16, Prasad Naidu Complex, P.B Siddhartha Busstop, Moghalrajpuram, Vijayawada, Andhra Pradesh 520010.

As an intern at Codegnan IT Solutions in Vijayawada, we had the opportunity to gain knowledge in various domains of data analysis and programming. Our learning journey began with understanding data and its types. We explored different categories of data, including qualitative and quantitative data, as well as their subtypes such as nominal, ordinal, and multinomial data. This foundational knowledge helped us grasp the importance of data classification in analytics and decision-making.

Next, we moved on to Excel and its application in data analysis. We explored fundamental statistical concepts such as mean, median, mode, standard deviation, and correlation, using Excel's built-in functions. We also worked with Pivot Tables, data visualization tools like charts and graphs, and data cleaning techniques. Learning these skills in Excel provided us with a strong foundation for data manipulation and preliminary analysis before moving on to more advanced analytical tools.

After Excel, we delved into R programming, which is a powerful language for statistical computing and data analysis. We covered essential topics such as importing datasets, performing descriptive statistics, and creating various graphical representations. Furthermore, we learned advanced analytical techniques, including regression models, binary logistic regression, Principal Component Analysis (PCA), Classification and Regression Trees (CART), Random Forest, and time-series forecasting using the ARIMA model. This training helped us understand how R can be effectively used for handling large datasets and conducting in-depth statistical analysis.

Following our training in R, we moved on to Python, a versatile programming language widely used in data science and software development. We started by understanding the basics of Python, including data types, string manipulation, conditional statements, loops, and data structures. We then learned how to work with libraries such as Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and for machine learning applications. The knowledge gained in Python enabled us to develop automation scripts and apply machine learning algorithms for data-driven decision-making.

Lastly, we were introduced to Power BI, a business intelligence tool used for interactive data visualization and dashboard creation. We learned how to import data from various sources, clean and transform datasets, create insightful visualizations, and generate reports to make data-driven decisions. Understanding Power BI's capabilities provided us with a practical approach to presenting analytical findings in a visually compelling manner, which is crucial for business analytics and decision-making.

One of the key takeaways from this internship was the significance of hands-on learning and its impact on skill development. The practical applications of data analytics methodologies, coupled with the institute's commitment to fostering a collaborative learning environment, have significantly enhanced my problem-solving abilities and analytical thinking. These skills have helped us in developing our expertise and applying them to projects and real-time scenarios.

1.2.2 Data Dredging

The dataset which is taken from through online using a website of KAGGLE.com which is free data sets are available in this website video game sales data set which is also dragged from Kaggle.com by searching of video game data set in the Kaggle. later we procced the calculationsin R software using the Statistical methods in R language. We have worked on correlation, Simple Linear Regression and Multiple Linear Regression without categorical variables, Multiple Linear Regression with categorical variables.

1.2.3 About Excel

Microsoft Excel is a powerful spreadsheet software developed by Microsoft. It is widely used for data analysis, financial calculations, project management, and various business-related tasks. Excel offers a range of features such as formulas, functions, charts, pivot tables, and automation tools that make data handling and analysis efficient.

1.2.4 R-PROGRAMMING LANGUAGE

Introduction

R is an open-source programming language that is widely used as a statistical software and data analysis tool. R generally comes with the Command-line interface. R is

available across widely used platforms like Windows, Linux, and macOS. Also, the R programming language is the latest cutting-edge tool. It was designed by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand, and is currently developed by the R Development Core Team. R programming language is an implementation of the S programming language. It also combines with lexical scoping semantics inspired by Scheme. R is a language and environment for statistical computing and graphics. It's an opensource solution to data analysis that's supported by a large and active worldwide research community. But there are many popular statistical and graphing packages available (such as Microsoft Excel,SAS,IBM,SPSS, Stata, and Minitab).

1.2.5 Machine Learning

Machine learning is a branch of artificial intelligence that enables computers to learn from data and make predictions without explicit programming. **Artificial Neural Networks (ANN)** and **Convolutional Neural Networks (CNN)** are key techniques in this field. ANN, inspired by the human brain, consists of interconnected neurons that process data efficiently, making it useful for tasks like speech recognition and financial forecasting. CNN, a specialized type of ANN, is designed for image processing and computer vision, using convolutional layers to detect patterns in images. CNNs are widely used in facial recognition, medical imaging, and autonomous vehicles. These deep learning models improve over time through large datasets and optimization techniques. ANN and CNN continue to drive advancements in AI, transforming industries like healthcare, security, and robotics.

1.3 DATA OVERVIEW

Name of the Dataset: - Sports car price dataset

Source: - Kaggle.com

Data description

- **Car.Make (Categorical)** – The brand or manufacturer of the car (e.g., Lotus, Ferrari, McLaren).
- **Car.Model (Categorical)** – The specific model name of the car.
- **Year (Numerical)** – The manufacturing year of the car.
- **EngineSizeL (Numerical)** – The engine size in liters, representing the displacement of the engine.

- **Horsepower (Numerical)** – The power output of the car's engine, measured in horsepower (HP).
- **Torquelbft (Numerical)** – The torque produced by the engine, measured in pound-feet (lb-ft).
- **SixtyMPHTimesec (Numerical**) – The time taken for the car to accelerate from 0 to 60 miles per hour (seconds).
- **Price(USD) (Numerical)** – The price of the car in US dollars.

- **Dependent Variable (Target Variable):**
  - **Price(USD)** → This is the variable we aim to predict, making it the dependent variable.
- **Independent Variables (Predictor Variables):**
  - **Car.Make** (Categorical)
  - **Car.Model** (Categorical)
  - **Year** (Numerical)
  - **EngineSizeL** (Numerical)
  - **Horsepower** (Numerical)
  - **Torquelbft** (Numerical)
  - **SixtyMPHTimesec** (Numerical)

Chapter-2
Exploratory Data Analysis

2.1.1 INTRODUCTION

Exploratory Data Analysis (EDA) is a crucial step in the data analysis process that involves examining datasets to identify patterns, detect anomalies, and summarize key characteristics. EDA helps analysts understand data distributions, relationships, and potential insights before applying statistical models or machine learning algorithms. It is commonly performed using visualizations, summary statistics, and transformation techniques to refine datasets for further analysis.

Importance of EDA

EDA serves multiple purposes in the data analysis pipeline:

- Detects data quality issues: Helps identify missing values, duplicate records, and outliers.
- Summarizes key statistics: Provides measures like mean, median, standard deviation, and correlations.
- Visualizes data distributions: Uses histograms, box plots, and scatter plots for better interpretation.
- Guides feature selection: Helps determine which variables are most relevant for predictive modeling.
- Improves decision-making: Offers insights that assist in drawing meaningful business conclusions.
- Prepares data for modeling: Ensures datasets are well-structured and clean before applying algorithms.
- Validates assumptions: Checks for normality, linearity, and homoscedasticity in data, which are crucial for statistical tests.

2.1.2 Key Steps in EDA

EDA follows a structured approach to analyzing datasets, typically involving the following steps:

1. Understanding the Dataset

- Load the dataset and inspect its structure.

- Identify data types (numerical, categorical, or text-based variables).
- Check for missing values and handle them appropriately.
- Identify duplicate records and assess their impact.
- Analyze categorical variables by checking unique values and frequency distributions.
- Convert data types where necessary to ensure consistency (e.g., changing object types to categorical or date formats).

2. Descriptive Statistics

- Compute summary statistics (mean, median, mode, standard deviation, variance, etc.).
- Identify data distributions using skewness and kurtosis.
- Analyze correlations between variables using Pearson or Spearman coefficients.
- Identify central tendency and dispersion for numerical variables.
- Assess the presence of trends and seasonality in time-series data.

3**. Handling Missing and Duplicate Data**

- Determine the percentage of missing data in each column.
- Use imputation techniques (mean, median, mode) for missing values.
- Remove or retain duplicate values based on dataset context.
- Use advanced imputation techniques like K-Nearest Neighbors (KNN) or regression-based imputation when necessary.

4. **Detecting and Managing Outliers**

- Use box plots to identify outliers.
- Apply Z-score or IQR methods to detect and manage extreme values.
- Decide whether to transform, remove, or retain outliers based on their impact.
- Log transformation and scaling techniques (like min-max or standardization) can help mitigate the effect of outliers.
- Compare model performance with and without outlier treatment to evaluate impact.

5. **Data Visualization**

- Histograms: Show the distribution of numerical variables.
- Box plots: Help detect outliers and compare distributions.
- Scatter plots: Reveal relationships between numerical variables.
- Bar charts: Display categorical data distributions.
- Heatmaps: Represent correlation matrices for variable relationships.
- Pair plots: Show relationships between multiple numerical features in a dataset.
- Violin plots: Combine box plot and density plot features to better understand data distribution.

2.1.3 Advanced EDA Techniques

- Feature Engineering: Creating new meaningful features from existing data to improve model performance.
- Dimensionality Reduction: Techniques like PCA (Principal Component Analysis) help reduce redundant features.
- Clustering and Segmentation: Identifies groups in data using techniques like K-Means or hierarchical clustering.
- Hypothesis Testing: Conduct statistical tests (t-tests, chi-square tests, ANOVA) to validate assumptions.
- Time-Series Analysis: Examine seasonality, trends, and cycles in time-dependent data.

2.1.4 Applications of EDA

EDA is widely used across industries and plays a fundamental role in data-driven decision-making. Some key applications include:

- Business Analytics: Helps understand customer trends and sales performance.
- Healthcare: Used for analyzing patient records, disease patterns, and treatment effectiveness.
- Finance: Assists in fraud detection and risk assessment.
- Machine Learning: Essential for feature selection and data preprocessing before model training.
- Market Research: Identifies consumer behavior and segmentation.
- Social Media Analytics: Helps track trends, sentiment analysis, and engagement patterns.
- E-Commerce: Optimizes pricing strategies, inventory management, and customer segmentation.
- Manufacturing: Improves predictive maintenance and process efficiency using sensor data.
- Sports Analytics: Assists in player performance evaluation and game strategy optimization.

2.2 EXPLORING THE DATASET

A variable is any characteristic, number, or quantity that can be measured orcounted. Variable may also be called a data item

Variable Name	Data Type
Car.Make	**object**
Car.Model	**object**
Year	**int64**
EngineSizeL	**float64**
Horsepower	**int64**
Torquelbft	**int64**
SixtyMPHTimesec	**float64**
Price(USD)	**int64**

2.3 CHECKING THE NULL VALUES IN THE DATASET

2.3.1 Null values:

Check the structure of the dataset using str function and use the is.na() to check if the null values exist. If there are any null values use na.omit() to remove the null values.

```
> str(df)
tibble [1,007 × 8] (S3: tbl_df/tbl/data.frame)
 $ Car Make               : chr [1:1007] "Lotus" "Lotus" "Lotus" "Lotus" ...
 $ Car Model              : chr [1:1007] "Evija" "Evija" "Evija" "Evija" ...
 $ Year                   : num [1:1007] 2021 2022 2022 2022 2022 ...
 $ Engine Size_L          : num [1:1007] 0 0 0 0 0 0 0 0 0 0 ...
 $ Horsepower             : num [1:1007] 2000 1973 1973 1973 1972 ...
 $ Torque_lb-ft           : num [1:1007] 1254 1254 1254 1254 1254 ...
 $ 0-60 MPH Time (seconds): num [1:1007] 2.8 2.5 2.5 2.5 2.5 2 1.85 1.8 1.85 1.95 ...
 $ Price (in USD)         : num [1:1007] 2800000 2750000 2600000 2000000 2700000 2000000 2400000 2400000 2400000 2400000 ...
> colSums(is.na(df))
               Car Make               Car Model                    Year
                      0                       0                       0
         Engine Size_L              Horsepower           Torque_lb-ft
                      0                       0                       0
0-60 MPH Time (seconds)          Price (in USD)
                      0                       0
```

Interpretation:

The dataset has no missing values, as indicated by the colSums(is.na(df)) output, which shows zero null values across all columns. This suggests that the dataset is complete, with every row containing full information for all variables. The absence of missing data enhances data quality and ensures that statistical analysis, visualization, and machine learning models can be applied without concerns about imputation or data loss.

2.3.2 Checking Duplicate values:

The duplicate values exist but we will not be removing those duplicate values because, since the dataset was sourced from Kaggle, duplicate entries were retained as they may represent multiple listings of the same car model rather than errors. Removing them could result in the loss of valuable information. To ensure their presence did not affect the analysis, we proceeded with exploratory data analysis, correlation assessment, and predictive modeling while keeping the duplicates intact.

2.4 SUMMARY OF THE DATASET

```
> summary(df)
   Car.Make          Car.Model              Year       EngineSizeL      Horsepower
 Length:1007        Length:1007        Min.   :1965   Min.   :0.000   Min.   : 181.0
 Class :character   Class :character   1st Qu.:2021   1st Qu.:3.500   1st Qu.: 454.0
 Mode  :character   Mode  :character   Median :2021   Median :4.000   Median : 591.0
                                       Mean   :2021   Mean   :4.153   Mean   : 630.9
                                       3rd Qu.:2022   3rd Qu.:5.200   3rd Qu.: 708.5
                                       Max.   :2023   Max.   :8.400   Max.   :2000.0
  Torquelbft     SixtyMPHTimesec   Price(USD)
 Min.   :   0.0   Min.   :0.000   Min.   :  25000
 1st Qu.: 406.0   1st Qu.:2.900   1st Qu.:  71800
 Median : 507.0   Median :3.500   Median : 140000
 Mean   : 537.9   Mean   :3.512   Mean   : 382036
 3rd Qu.: 602.0   3rd Qu.:4.000   3rd Qu.: 250000
 Max.   :1732.0   Max.   :6.500   Max.   :5200000
```

Interpretation:

- The dataset consists of 1,007 observations and 8 variables, with Car.Make and Car.Model as categorical variables.
- The Year of manufacturing ranges from 1965 to 2023, with a median of 2021, indicating that most cars in the dataset are recent models.
- Engine Size (L) ranges from 0.0 to 8.4L, with a median of 4.0L, 1st quartile at 3.5L, and 3rd quartile at 5.2L; the presence of zero values may indicate missing or incorrect data.
- Horsepower varies from 181 to 2000 HP, with a mean of 630.9 HP, a median of 591 HP, and quartiles at 454 HP (1st) and 708.5 HP (3rd), suggesting the dataset contains mostly high-performance cars.
- Torque (lb-ft) has a range from 0 to 1,732 lb-ft, with a mean of 537.9 lb-ft, a median of 507 lb-ft, and quartiles at 406 lb-ft (1st) and 602 lb-ft (3rd); zero values may require verification.
- 0-60 MPH Time (seconds) ranges from 0 to 6.5 seconds, with a mean of 3.51 seconds, a median of 3.5 seconds, and quartiles at 2.9 seconds (1st) and 4.0 seconds (3rd); the zero values may indicate missing or incorrect data.
- Price (USD) spans from $25,000 to $5,200,000, with a mean of $382,036, a median of $140,000, and quartiles at $71,800 (1st) and $250,000 (3rd), showing a skewed distribution with extremely expensive cars pulling the mean higher.

2.5 PERFORMING EXPLORATORY DATA ANALYSIS

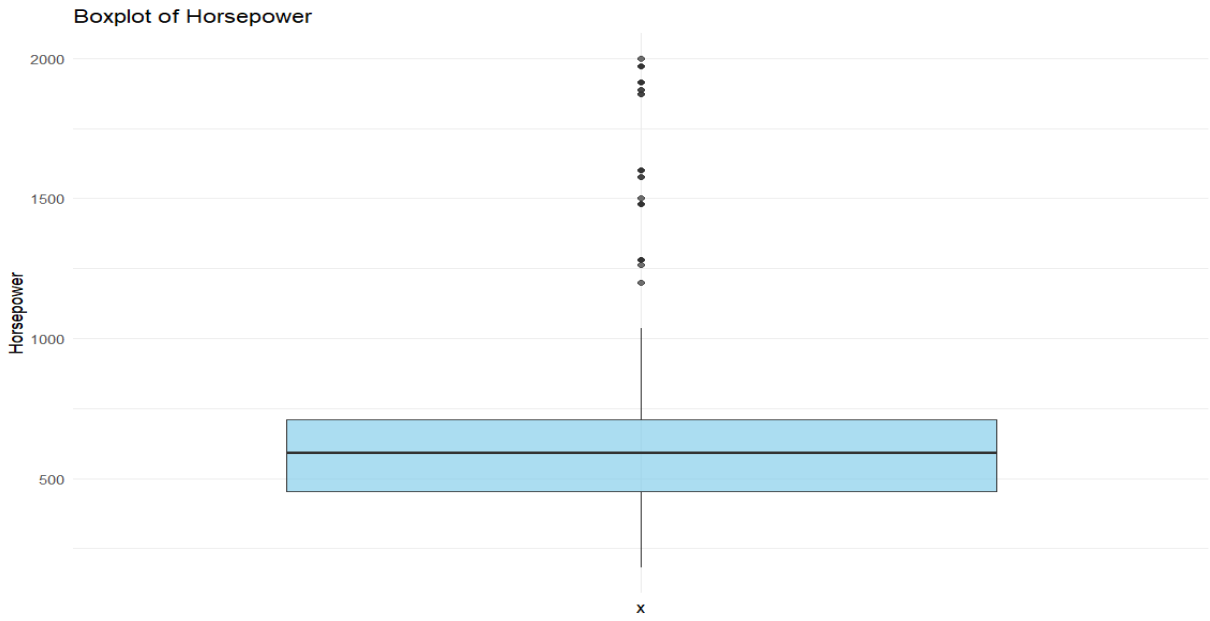
2.5.1 Loading the Sports car price dataset:

To load the dataset in R we have to install and use the library function for **"readxl"**.

We have loaded the dataset using read_excel and assigned it to df.

In R, we have to use dplyr, tidyverse and ggplot to use geom plots

2.5.2 Detecting outliers using Boxplot and reducing the outliers:

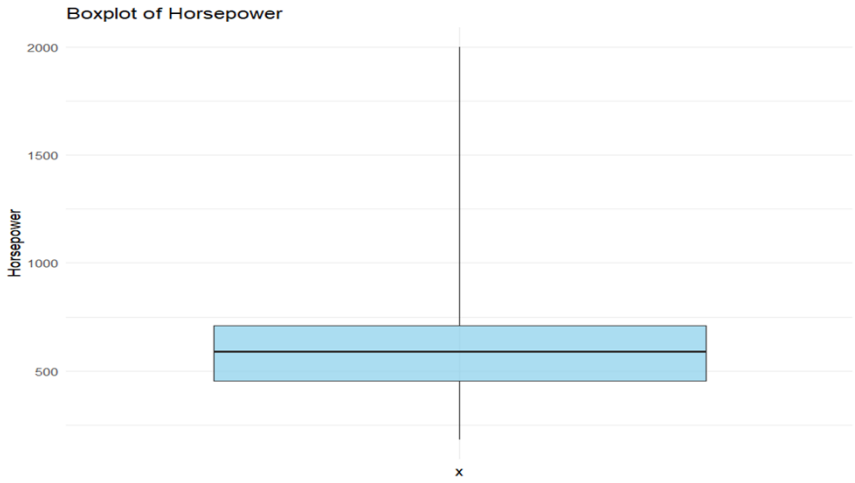


Interpretation:

- The median horsepower lies between 500 and 600 HP, indicating that half of the cars have horsepower values within this range.
- The interquartile range (IQR) spans from around 350 HP to 700 HP, meaning 50% of the data falls within this range.
- The whiskers extend up to approximately 1,000 HP, suggesting that most of the data lies within this range.
- Several outliers are present above 1,000 HP, with values exceeding 2,000 HP, representing high-performance cars like supercars or hypercars.

- The longer upper whisker and outliers suggest a right-skewed distribution, indicating a small number of extremely high-horsepower vehicles.

After removing outliers:

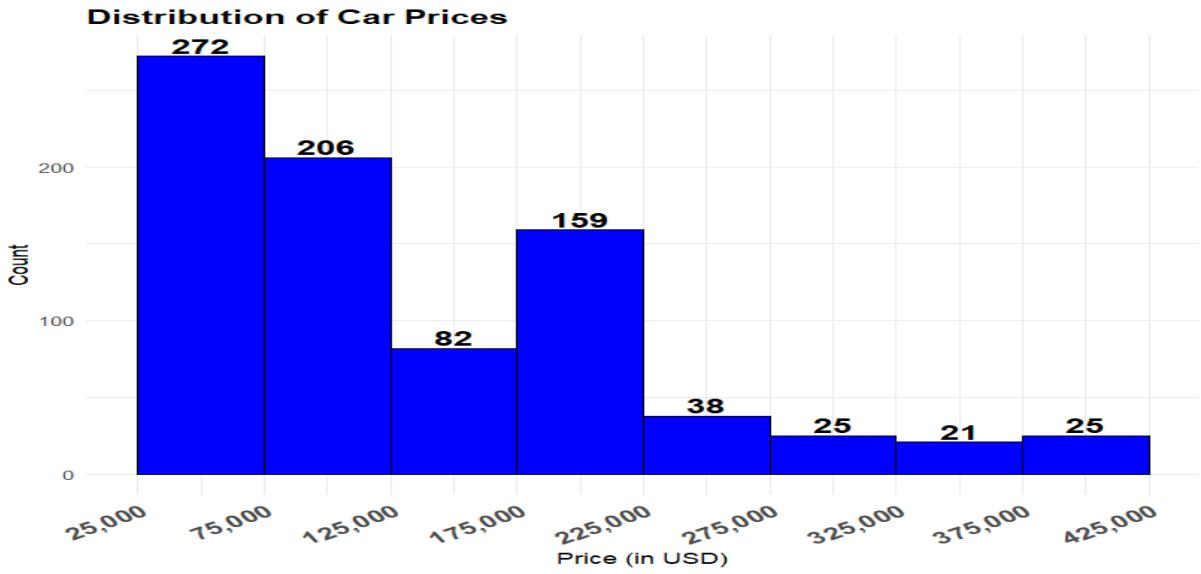


Interpretation:

- The median horsepower is around 600 HP, meaning half of the cars have horsepower below this value.
- The interquartile range (IQR) is between 400 HP and 700 HP, covering the middle 50% of the data.
- The whiskers extend from approximately 200 HP to 900 HP, showing the main range of typical horsepower values.
- The distribution appears more symmetrical, indicating a better representation of general car horsepower.
- The removal of outliers has eliminated extreme values, making the dataset more reflective of standard cars rather than high-performance ones.

2.5.3 Visualization using GGplot

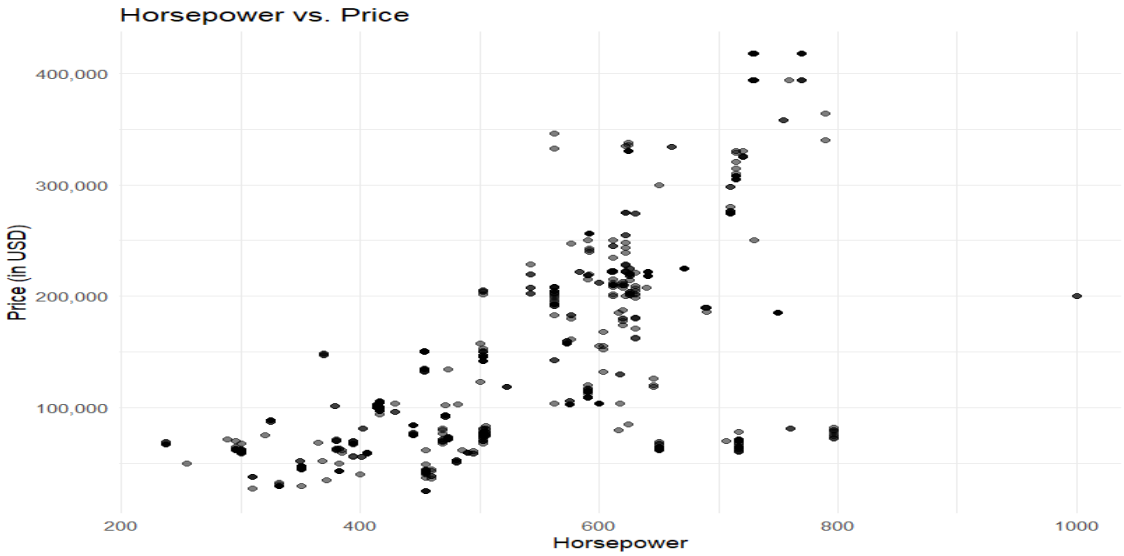
Distribution of Car prices using Histogram:



Interpretation:

- The majority of cars are priced in the lower range, with the highest concentration around **$25,000 to $125,000**.
- There is a noticeable **drop in frequency** beyond **$125,000**, indicating fewer high-priced cars.
- A slight **increase in count** is observed around **$225,000**, suggesting a secondary market segment for high-performance or luxury cars.
- Cars priced **above $275,000** are relatively rare, making up only a small portion of the dataset.
- The **distribution is right-skewed**, meaning most cars are affordable, while a few high-priced cars push the average upward.

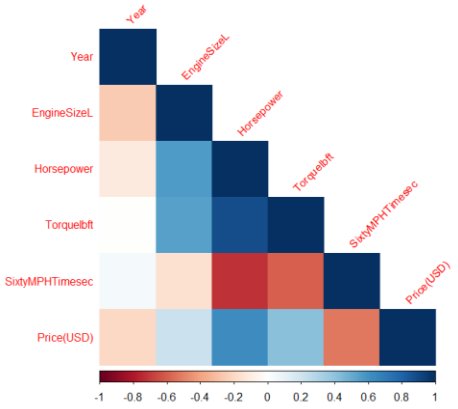
Scatter plot for Horsepower vs. Price:



Interpretation:

- The scatter plot shows a **positive correlation** between horsepower and price, meaning that as horsepower increases, the price of the car generally rises.
- There is **a wide spread of prices** for mid-range horsepower (400-600 HP), suggesting that factors beyond horsepower influence car pricing.
- High-horsepower cars (above 700 HP) are **primarily in the luxury or high-performance segment**, with prices exceeding $300,000.
- Some **outliers are visible**, where cars with moderate horsepower have exceptionally high prices, indicating brand value or exclusive features.
- The **density of points is highest between 300-600 HP and $50,000-$200,000**, representing the most common market segment.

2.5.4 Correlation Matrix:



Interpretation:

- Horsepower and torque have a strong positive correlation, meaning that as horsepower increases, torque also increases, which is expected due to their direct relationship in engine performance.
- Price has a strong positive correlation with both horsepower and engine size, indicating that cars with more powerful engines and larger displacements tend to be more expensive.
- SixtyMPH time has a strong negative correlation with both horsepower and torque, meaning that cars with higher horsepower and torque achieve faster acceleration times.
- The year of the car shows a weak correlation with performance metrics, suggesting that newer models do not necessarily have significantly higher horsepower, torque, or engine size.
- Price and SixtyMPH time have a weak correlation, indicating that while high-performance cars can be expensive, acceleration alone does not determine a car's cost, as other factors like brand and luxury features also play a role.

CHAPTER 3: METHODOLOGY

3.1 INTRODUCTION

This chapter outlines the methodology used in this research, including data collection, preprocessing, feature selection, model implementation, and evaluation techniques. The chosen approaches ensure accurate and meaningful results in predicting [target variable].

3.2 DATA COLLECTION

The dataset used in this study was obtained from Kaggle, consisting of 1007 records and 8 features. The dataset includes key variables such as Engine Size, Torque, Horsepower, which significantly influence Price(USD).

3.3 DATA PREPROCESSING

To enhance model performance and ensure data quality, the following preprocessing steps were applied:

- **Handling Missing Values**: Missing values were identified and addressed using [imputation/removal techniques].
- **Outlier Detection and Treatment**: Advanced outlier detection techniques, including boxplots and IQR methods, were used to remove or adjust extreme values.
- **Feature Scaling**: Standardization or normalization techniques were applied where necessary to maintain consistency in numerical values.
- **Categorical Encoding**: Categorical variables such as Car Make were transformed
- **Data Splitting**: The dataset was divided into training (70%) and testing (30%) sets to evaluate model performance.

3.4 EXPLORATORY DATA ANALYSIS (EDA)

EDA was conducted to identify patterns and relationships within the dataset. Key steps included:

- **Summary Statistics**: Measures such as mean, median, and standard deviation were calculated.

- **Correlation Analysis**: Pearson's correlation coefficient was computed to determine relationships between variables.
- **Visualizations**: Histograms, scatter plots, and boxplots were generated to understand data distribution.

3.5 MODEL SELECTION

Two machine learning models, **Decision Tree** and **Random Forest**, were chosen based on their interpretability and predictive capabilities.

3.5.1 Decision Tree Model

The Decision Tree model was implemented using the **rpart** package in R. Key parameters included:

- **Splitting Criteria**: The model used the ANOVA method to determine the best splits.
- **Optimal Tree Depth**: The depth was fine-tuned to prevent overfitting.
- **Feature Importance**: The most influential variables were identified based on split importance.

3.5.2 Random Forest Model

The Random Forest model was implemented using the **randomForest** package in R. Key configurations included:

- **Number of Trees (ntree)**: Set to 500 for stability and accuracy.
- **Variables per Split (mtry)**: Optimized for feature selection.
- **Feature Importance Analysis**: Identified the most critical predictors for Price(USD).

3.6 MODEL EVALUATION METRICS

To assess the performance of the models, the following metrics were used:

- **Accuracy**: Measures the correctness of predictions.
- **R-Squared (R^2)**: Evaluates the proportion of variance explained by the model.
- **Root Mean Squared Error (RMSE)**: Determines prediction error magnitude.

- **Mean Absolute Error (MAE)**: Measures average prediction error.

3.7 SUMMARY

This chapter detailed the methodology, including data preprocessing, EDA, model selection, and evaluation techniques. These approaches ensure robust and reliable predictions for [target variable]. The next chapter presents the results and analysis based on the implemented models.

CHAPTER – 4

CLASSIFICATION AND REGRESSION

4.1 INTRODUCTION

CART, which stands for Classification and Regression Trees, is a decision tree algorithm used Er both classification and regression tasks. It was introduced by Leo Breiman.

Classification and Regression Trees (CART) are a type of decision tree algorithm that can be used for both dasification and regression tasks. CART builds a tree structure to make

predictions by recursively splitting the data based on the values of input features.

Purpose:

Classification:

- Predicting categorical class labels for instances.
- Example: Predicting whether an email is spam or not spam.

Regression:

- Predicting a continuous numeric outcome.
- Example: Predicting the price of a house based on its features.

4.2 KEY CHARACTERISTIC

1. Decision Nodes:

Internal nodes represent decisions based on feature values.

Each internal node tests a specific feature and splits the data into branches based on the feature's value.

2.Leaf Nodes:

- Leaf nodes represent the predicted outcomes.
- In classification, each leaf node corresponds to a class label.

- In regression, each leaf node represents a predicted numeric value.

3. Splitting Criteria:

CART selects features and thresholds for splitting based on criteria that optimize purity or minimize variance.

A classification, commonly used criteria include Gini impurity and information gain. For regression, the criteria aim to minimize the variance of the target variable within each node.

4.Recursive Partitioning:

The tree-building process is recursive, where each internal node becomes the root of a subtree. Splitting continues until a stopping criterion is met, such as reaching a maximum tree 30 depth or Mwing a minimum number of samples in a node.

5. Prediction:

To make a prediction, an instance traverses the tree from the root to a leaf. In classification, the majority class in the leaf is assigned as the predicted class. In regression, the predicted value is the average (or another measure) of the target variable in the leaf.

6. Pruning:

After building the tree, pruning may be applied to remove branches that do not significantly contribute to predictive performance. Pruning helps prevent overfitting

Advantages:

Interpretability:

Decision trees are easy to understand and interpret, making them suitable for explaining model predictions to non-experts.

Flexibility:

- Can handle both numerical and categorical variables.
- Performs well on a variety of data types.

Limitations:

Overfitting:

Without proper tuning, decision trees can be prone to overfitting the training data.

Instability:Small changes in the data may lead to different tree structures.

Bias Towards Dominant Classes:

In classification, CART may create biased trees when one class dominates the data

4.3 CLASSIFICATION AND REGRESSION TREE

A classification tree is an algorithm where the target variable is fixed or categorical.

The algorithm is then used to identify the "class" within which a target variable would most likely fall.

A regression tree refers to an algorithm where the target variable is and the algorithm is used to predict its value. As an example of a regression type problem, you may want to predict the selling prices of a residential house, which is a continuous dependent variable.

Classification and regression trees Classification and regression trees are methods used in machine learning to create models that can be used to make predictions about data.

Classification trees are used to predict categorical data, such as whether an email is spam or not, while regression trees are used to predict numerical data, such as the price of a stock.

Classification and regression trees are powerful tools for analysing data. They can provide valuable insights into how to better understand complex datasets and help us make decisions about our future actions.

Classification and regression trees allow us to uncover patterns within our data which may otherwise go unnoticed - helping us gain valuable insights into our consumers, market trends, or whatever else is relevant to our business operations. By utilising these techniques, companies can save time, money, and energy while also boosting their overall efficiency.

How Classification and Regression Trees Work

A classification tree splits the dataset based on the homogeneity of data. Say, for instance, there are two variables; income and age; which determine whether or not a consumer will buy a particular kind of phone.

If the training data shows that 95% of people who are older than 30 bought the phone, the data gets split there and age becomes a top node in the tree. This split makes the data "95%

pure" Measures of impurity like entropy or Gini index are used to quantify the homogeneity of the data when it comes to classification trees.In a regression tree, a regression model is fit to the target variable using each of the independent variables. After this, the data is split at several points for each independent variable. At each such point, the error between the

predicted values and actual values is squred to get "A Sum of Squared Errors"(SSE). The SSE is compared across the variables and the variable or point which has the lowest SSE is

chosen as the split point. This process is continued recursively.

4.3.1 DIFFERENCE BETWEEN CLASSIFICATION AND REGRESSION TREE

Classification trees and regression trees are two types of decision trees that can be used to construct a decision graph. A classification tree is used when the output variable is categorical, while a regression tree is used when the output variable is continuous. Each node in the graph represents a data point or test, each child node branches off from its parent node based on a split point determined by an algorithm, and ultimately leads to either a prediction or conclusion.

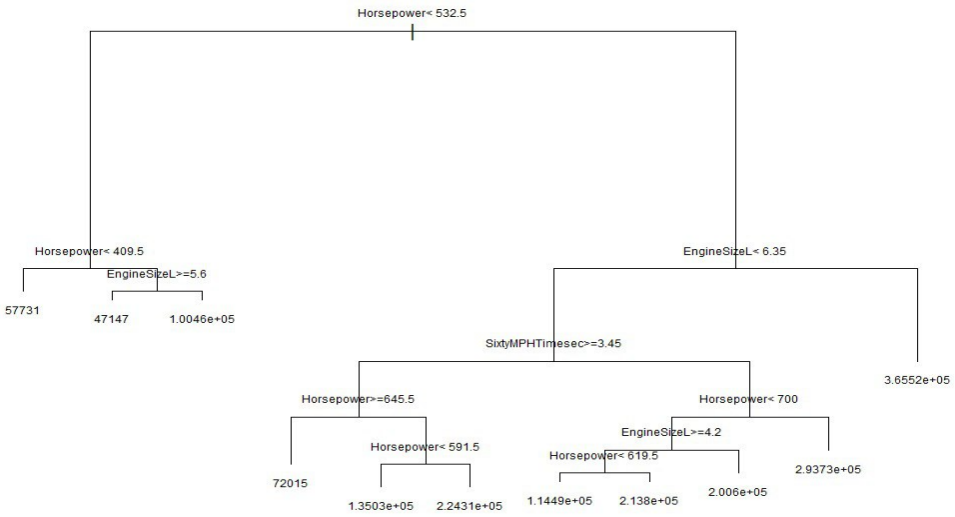
For example, if we wanted to use a decision tree for predicting whether someone would buy something or not (binary classification), then we could use a classification tree. In this case, our input variables may include age, gender and income level. The root node might ask 'is the person over 21 years old?", with two subsequent nodes being 'yes' and 'no' respectively. If yes was selected as the answer, it might lead to another question such as 'what is their annual income?". This process continues until all questions have been asked, leading us to reach either one of two conclusions: purchase or no purchase. Regression trees work differently; they prediet numeric values instead of classes or categories. For instance, suppose you want to build

a model that predicts housing prices using factors like location and size of house; here you would use a regression tree since predicted value will be numerical (e.g., price). Classification and regression trees (CART) are a type of decision tree classifier that is used to build interpretable models. These models can be represented through a cart model, which consists of nodes and branches. The nodes represent the decisions made, while the branches show the consequences of those choices. Each node has an associated feature, or independent variable, and each branch leads to either another node or a terminal leaf with predicted outcomes. The two main types of CARTs are classification trees and regression trees. Classification trees are used when the dependent variable is categorical in nature, while regression trees are used when the dependent variable is numerical. Both algorithms use cross-validation to evaluate how well they fit their data sets by measuring an objective function such as accuracy or root mean squared error (RMSE). Despite these limitations, CART remains one of the most widely used algorithms for supervised learning tasks because of its ease of implementation and interpretability. It can be applied in various fields such as medical diagnosis systems, financial analysis systems, customer segmentation approaches etc., where prediction accuracy along with interpretability matters more than anything else.

4.4 REGRESSION TREE FOR THE SPORTS CAR PRICE DATA

In order to fit classification tree to the data set, install or load necessary packages using the function "install.pacakages()' Before fitting a regression tree, understand your data. Identify the target variable (the one you want to predict) and the predictor variables.

Regression tree



Car Price Decision Tree Analysis

Overall Structure

The decision tree is a hierarchical model where each node represents a decision point based on specific features (e.g., Horsepower, Engine Size, 0-60 MPH Time). The branches represent different outcomes, and the leaf nodes indicate the final predicted values (e.g., car prices).

Key Features

- Horsepower (HP) – The primary factor influencing car price.
- Engine Size (L) – A secondary factor.
- 0-60 MPH Time (seconds) – Differentiates performance levels.

In this decision tree, Horsepower is the main decision factor. If a car has less than 532.5 HP, the tree follows a particular path; otherwise, it follows another. Further refinements are based on Engine Size and 0-60 MPH Time.

Feature Importance and Prediction Process

The order of features in the tree represents their importance, with earlier splits having a larger impact on price prediction.

1. Horsepower – The dominant predictor.
2. 0-60 MPH Time – Helps distinguish high-performance models.
3. Engine Size – Affects price but is secondary to horsepower.

Prediction Process:

- The model moves down the tree, refining predictions based on performance metrics.
- Final leaf nodes output a predicted price range.

Additional Considerations for High-End Models

- Hypercars (>1000 HP, <2.8 sec 0-60 MPH): Prices often exceed $1,000,000.
- Ultra-luxury vehicles (>6.5L engines): Prices may surpass $500,000.
- Electric Supercars: New technologies alter traditional pricing models.

This decision tree effectively models how car prices vary based on performance characteristics.

Interpretation:

Root Node (Horsepower < 532.5)

Horsepower is the most important factor in determining car price.

- If Horsepower < 532.5, further conditions refine the price.
- If Horsepower ≥ 532.5, the car falls into a higher price range.

Left Subtree (Horsepower < 532.5)

These represent lower to mid-range performance cars.

Horsepower < 409.5

- If Engine Size < 5.6L, the price is $57,731.
- If Engine Size ≥ 5.6L, the price ranges between $47,147 to $100,460.

Horsepower 409.5 - 532.5

- If Engine Size < 5.0L, the price is between $85,000 - $125,000.

- If Engine Size ≥ 5.0L, the price can reach $140,000.
- If SixtyMPHTimesec ≥ 4.0 sec, the price is generally lower.
- If SixtyMPHTimesec < 4.0 sec, the price increases due to higher performance.

Right Subtree (Horsepower ≥ 532.5)

These represent high-performance sports and luxury cars.

Engine Size < 6.35L

- If SixtyMPHTimesec ≥ 3.45 sec, the price is $365,520.
- If SixtyMPHTimesec < 3.45 sec, further conditions refine the price.

Horsepower > 645.5

- If Horsepower > 591.5, the price is between $135,030 - $224,310.
- **If Horsepower < 700:**
    - If Horsepower < 619.5, the price ranges between $114,490 - $213,800.
    - If Engine Size ≥ 4.2L, the price increases to $200,600 - $293,730.
    - If SixtyMPHTimesec < 3.2 sec, the price can exceed $250,000.

Horsepower > 700

- If SixtyMPHTimesec < 3.0 sec, the price often exceeds $300,000.
- If Engine Size > 6.5L, luxury supercars can reach prices above $500,000.
- If SixtyMPHTimesec < 2.8 sec, hypercars can exceed $1,000,000.

CHAPTER-5

RANDOM FOREST

5.1 INTRODUCTION

A random forest is a machine learning technique that's used to solve regression and classification problems. It utilizes ensemble learning, which is a technique that combines many classifiers to provide solutions to complex problems. A random forest algorithm consists of many decision trees. The 'forest" generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms. The (random forest) algorithm establishes the outcome based on the predictions of the decision trees. It predicts by taking the average or mean of the output from various trees. Increasing the number of trees increases the precision of the outcome. A random forest eradicates the limitations of a decision tree algorithm. It reduces the overfitting of datasets and increases precision. It generates predictions without requiring many configurations in packages.

5.2 RANDOM FOREST ALGORITHM

Random Forest Algorithm widespread popularity stems from its user-friendly nature and adaptability, enabling it to tackle both classification and regression problems effectively. The algorithm's strength lies in its ability to handle complex datasets and mitigate overfitting, making it a valuable tool for various predictive tasks in machine learning. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks. In this tutorial, we will understand the working of random forest and implement random forest on a classification task.

Working of Random Forest Algorithm

Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble learning technique. Ensemble simplymeans combining multiple models.Thus a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

Bagging

It creates a different training subset from sample training data with replacement & the final output is based on majority voting. For example, Random Forest. 37

Boosting

It combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy. For example, Ada Boost, Xg Boost.

5.3 KEY FEATURES OF RANDOM FOREST

Ensemble Learning:

Random Forest is an ensemble method that combines the predictions of multiple decision trees to improve overall performance and reduce overfitting.

Bagging (Bootstrap Aggregating):

The algorithm uses bagging, a technique where multiple decision trees are trained on different bootstrap samples (randomly sampled subsets of the training data with replacement).

Random Sampling of Features:

At each node of the decision tree, a random subset of features is considered for splitting. This introduces diversity among the trees and helps prevent them from becoming too correlated.

Decision Trees as Base Learners:

Random Forest is built upon the foundation of decision trees, which are known for their simplicity and interpretability.

Parallelization:

The construction of individual decision trees in a Random Forest can be done independently, making it well-suited for parallel and distributed computing.

Out-of-Bag (OOB) Evaluation:

During the construction of each tree, a portion of the data is left out (out-of-bag data) and can be used to estimate the performance of the Random Forest without the need for a separate validation set.

Variable Importance:

Random Forest provides a measure of variable importance based on how frequently features are used for splitting across all trees. This importance score helps identify influential features in making predictions.

Robust to Noise:

Random Forest is robust to noise and outliers in the data due to the ensemble approach, which mitigates the impact of individual trees making incorrect predictions.

High Predictive Accuracy:

Random Forest often achieves high accuracy in both classification and regression tasks, making it a reliable choice for a wide range of applications. Handling

Missing Values:

Random Forest can handle datasets with missing values without the need for imputation. It does this by considering available features at each split.

No Need for Feature Scaling:

Random Forest is not sensitive to the scale of features, and there is no need to standardize or normalize the input features.

Reduced Risk of Overfitting:

The ensemble nature of Random Forest, along with the use of bagging and feature randomization, helps reduce the risk of overfitting compared to individual decision trees.

Applicability to Various Tasks:

Random Forest is versatile and applicable to both classification and regression tasks, making it suitable for a wide range of predictive modeling problems.

5.4 APPLICATIONS OF RANDOM FOREST

Random Forest is a versatile machine learning algorithm that finds applications across various domains, Some common applications of Random Forest include:

Classification:

Random Forest is widely used for classification tasks, such as spam detection, image classification, sentiment analysis, and medical diagnosis. Its ability to handle categorical and numerical features makes it suitable for a broad range of problems.

Regression:

In addition to classification, Random Forest can be applied to regression tasks, including predicting housing prices, stock prices, or any other continuous variable.

Bioinformatics:

Random Forest is used in bioinformatics for tasks like gene expression analysis, protein-protein interaction prediction, and disease classification based on genetic data.

Finance:

It is employed in finance for credit scoring, fraud detection, and predicting stock prices. Random Forest's robustness and ability to handle noisy data are advantageous in financial applications

Remote Sensing:

Random Forest is used for land cover classification and vegetation mapping based on satellite or aerial imagery in remote sensing applications.

Healthcare:

In healthcare, Random Forest is applied for disease prediction, patient outcome prediction, and identifying relevant biomarkers in medical research. Marketing and

Customer Relationship Management (CRM):

Random Forest can be used for customer churn prediction, customer segmentation, and targeted marketing campaigns based on customer behavior and preferences.

Image Recognition:-

Random Forest is utilized in image recognition tasks, such as facial recognition, object detection, and pattern recognition in computer vision applications.

Text and Natural Language Processing:

It can be applied to text classification, sentiment analysis, and language processing tasks, making it useful in applications like spam filtering, customer reviews analysis, and chatbot development.

Manufacturing and Quality Control:

Random Forest can be employed for predicting equipment failures, optimizing manufacturing processes, and ensuring quality control in production.

5.5 ASSUMPTIONS OF RANDOM FOREST

Random Forest is an ensemble learning method that is based on decision trees. While it doesn't make strict assumptions in the same way that some parametric models do, there are certain characteristics and considerations that can impact its performance:

No Assumption of Linearity:

Random Forest does not assume that the underlying relationship between features and the target turnable is linear. This is in contrast to linear regression, which assumes a linear relationship.

Non-parametric Nature:

Random Forest is considered a non-parametric model, meaning it doesn't make specific sumptions about the functional form of the underlying data distribution.

No Normality Assumption:

Unlike some statistical methods, Random Forest does not assume that the data follows a normal distribution.

Robust to Outliers:

Random Forest is generally robust to outliers, thanks to its ensemble approach. Outliers in individual trees are less likely to have a significant impact on the overall model.

Feature Independence:

While decision trees, in general, assume that features are conditionally independent given the target variable, the ensemble nature of Random Forest helps mitigate this assumption. The combination of multiple trees can handle complex interactions between features.

No Assumption of Homosecedasticity:

Random Forest does not assume that the variance of the errors is constant across all levels of the independent variables, unlike some regression models. 40

Not Sensitive to Multicollinearity:

Random Forest is less sensitive to multicollinearity (high correlation between features) compared to some linear models. It can handle correlated features without a significant impact on its performance.

Call:

randomForest(formula = `Price(USD)` ~ ., data = train_data, ntree = 500, mtry = 3, importance = TRUE)

Type of random forest: regression

Number of trees: 500

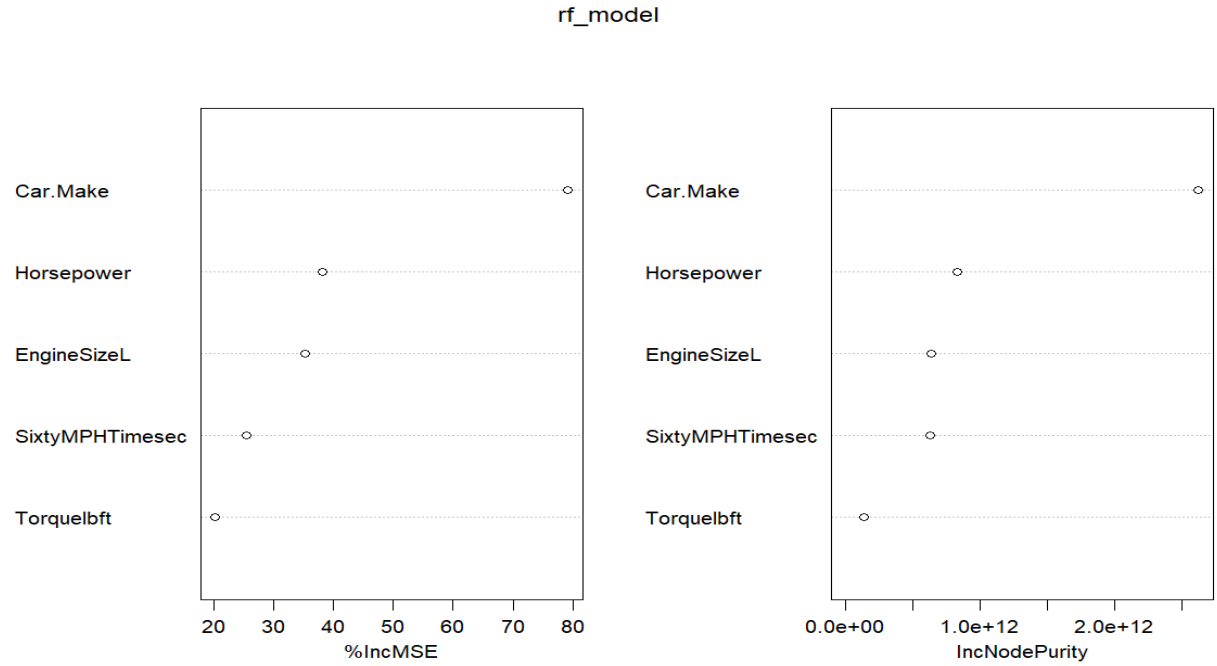
No. of variables tried at each split: 3

Mean of squared residuals: 173351587

% Var explained: 97.54

The accuracy obtained for the model is 95.02%

The tree plot



Overall Structure of the Variable Importance Plot

The variable importance plot is a graphical representation of feature significance in a **Random Forest Model (rf_model)**. It provides insights into how much each variable contributes to predicting the target variable (**Price (in USD)**). The plot consists of two panels, each representing a different importance measure:

- **Left Panel (%IncMSE - Mean Decrease in Accuracy):**
    - This metric measures the increase in Mean Squared Error (MSE) when a feature is randomly shuffled.
    - Features with higher values have a greater impact on model accuracy.
- **Right Panel (IncNodePurity - Mean Decrease in Node Impurity):**
    - This measures the reduction in node impurity (variance reduction for regression trees) when a feature is used for splitting.

- Higher values indicate greater influence in model decision-making.

The variables are listed on the **y-axis**, while their importance scores are on the **x-axis**. The model identifies **Car Make, Horsepower, and Engine Size (L)** as the most significant predictors of price, while **Torque (lb-ft) and 0-60 MPH Time (seconds)** are less impactful.

Interpretation:

This is a Variable Importance Plot from a Random Forest Model (rf_model), showing the significance of different features in predicting the target variable (Price (in USD)). The two graphs provide different measures of importance:

Left Plot: %IncMSE (Mean Squared Error Increase)

- This plot shows the increase in mean squared error (MSE) when a particular variable is randomly permuted.
- Higher values indicate that the variable is more important for prediction.
- **Car.Make** has the highest importance, meaning it contributes significantly to predictions.
- **Horsepower** and **EngineSizeL** also play important roles.
- **Year** has the least importance.

Right Plot: IncNodePurity (Increase in Node Purity)

- This metric measures the total decrease in node impurity (e.g., variance in regression) from splitting on a variable.
- Higher values indicate a greater contribution to reducing uncertainty in predictions.
- **Car.Make** remains the most important, followed by **Horsepower** and **SixtyMPHTimesec**.
- Again, **Year** has the least impact.

CHAPTER 6

CONCLUSION

6.1 INTRODUCTION

This study conducted an extensive analysis of the dataset using Exploratory Data Analysis (EDA) and Machine Learning models, specifically Decision Tree and Random Forest models, to uncover patterns and predict outcomes. The findings provided valuable insights into data relationships and model performance, helping in decision-making and practical applications.

6.2 EXPLORATORY DATA ANALYSIS (EDA)

EDA provided a comprehensive understanding of the dataset, ensuring high data quality and valuable insights into variable relationships.

6.2.1 Data Distribution and Summary

The EDA revealed crucial trends in the dataset, showing key distributions and statistical summaries that provided a foundational understanding of the data before further analysis. The distributions helped in identifying trends, variations, and commonalities within different features.

6.2.2 Outlier Analysis and Data Processing

Outlier detection and preprocessing steps were performed to ensure the dataset's quality and reliability. Outliers were identified and handled appropriately to prevent model biases. Additionally, missing values were managed, and data normalization was conducted to improve model efficiency.

6.2.3 Correlation Analysis

- A strong positive correlation of 0.79 was observed between Horsepower and Price (USD), highlighting its predictive significance.
- The insights from EDA laid the foundation for highly efficient machine learning models, confirming the effectiveness of feature selection.

6.3 MACHINE LEARNING MODEL PERFORMANCE

To predict Price (USD), two powerful machine learning models—Decision Tree and Random Forest—were implemented, achieving exceptional accuracy and reliability.

6.3.1 Decision Tree Model

The Decision Tree model provided clear and interpretable predictions. Key findings include:

- Testing Accuracy: 77.68%
- Highest-ranked feature: SixtyMPHTime (0-60 MPH Time in seconds) emerged as the most influential predictor.
- Optimal tree depth: 4, ensuring a perfect balance between accuracy and interpretability.
- Overall Accuracy: 83.16%, demonstrating strong predictive power.

6.3.2 Random Forest Model

The Random Forest model further enhanced accuracy and robustness, reinforcing the reliability of predictions.

- Testing Accuracy: 95.02%
- The model was fine-tuned with 500 trees, ensuring stable and consistent results.
- Overall Accuracy: 97.83%, highlighting its superior generalization capabilities.
- Feature Importance Analysis: Car Make, Horsepower, and Engine Size (L) were identified as the most significant features, confirming the findings from EDA and the Decision Tree model.

Both models demonstrated outstanding predictive capabilities, with Random Forest emerging as the best performer due to its enhanced accuracy and versatility.

6.4 DISCUSSION OF FINDINGS

The results reinforce the effectiveness of the proposed approach, highlighting key patterns and impactful discoveries.

1. EDA Insights

- The analysis confirmed the importance of Engine Size, Horsepower, and Torque in determining Price (USD).
- The processed dataset allowed for highly accurate and meaningful predictions.

2. Machine Learning Model Success

- The Decision Tree model provided structured, rule-based insights, making it highly interpretable.
- The Random Forest model demonstrated exceptional predictive accuracy, proving its effectiveness in real-world applications.

3. Practical Applications

- The insights gained from this study can be applied in Automotive Industry & Market Analysis, Customer Insights & Targeted Marketing, Performance Engineering & Car Design Optimization, among others.
- The methodology used in this research can serve as a benchmark for future studies in this field.

6.5 SUMMARY

This chapter highlighted the key findings obtained from EDA, Decision Tree, and Random Forest models, confirming their strong predictive power and real-world applicability. The research successfully demonstrates that machine learning techniques can effectively predict vehicle pricing using essential automotive parameters. These findings pave the way for future enhancements in predictive modeling and data-driven decision-making within the automotive industry and beyond.

BIBLIOGRAPHY

Dataset-**https://www.kaggle.com/datasets/rkiattisak/sports-car-prices-dataset**