MD2201 Data Science
Course Project

S.No	Div	Batch No	Group No	Roll No	Gr.No	Name of Student
1	IT-B	1	1	2	12211665	Pranav Kale
2				13	12210893	Maitrey Katkar
3				17	12210313	Omkar Khanvilkar
4				18	12210610	Janvi Kharat
5				20	12210505	Gita Kolate
6				21	12210697	Babusha Kolhe

1. **Project Title: OLX Car Price Prediction**

2. **Data Set Name: Used Cars Price Prediction**

3. **Data Set Source: Kaggle**

4. **Data set Link:** https://kaggle.com/datasets/avikasliwal/used-cars-price-prediction?select=train-data.csv

5. **Data Set Description:**

   This Dataset consist features or columns like:

   - **Name:** full name of the cars
   - **Location:** Location of the car owner
   - **Year:** Launch Year of particular car model
   - **Kilometers_Driven:** Kilometres Driven by particular car
   - **Fuel_Type:** Fuel Type of the car (Petrol, Diesel, Hybrid)
   - **Transmission:** Type of transmission (manual, auto)
   - **Owner_Type:** Which type of owner (first, second, third)
   - **Mileage:** Mileage of the car
   - **Engine:** Engine capacity of the car
   - **Power:** Power of the car
   - **Seats:** Number of seats inside the car
   - **New_Price:** Market price of new model of same car

- **Price:** Actual Price of the car asked by the owner

Preview of dataset

	Name	Location	Year	Kilometers	Fuel_Type	Transmissi	Owner_Ty	Mileage	Engine	Power	Seats	New_Price	Price
0	Maruti Wa	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5		1.75
1	Hyundai Cr	Pune	2015	41000	Diesel	Manual	First	19.67 kmp	1582 CC	126.2 bhp	5		12.5
2	Honda Jaz:	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5	8.61 Lakh	4.5
3	Maruti Ert	Chennai	2012	87000	Diesel	Manual	First	20.77 kmp	1248 CC	88.76 bhp	7		6
4	Audi A4 Ne	Coimbator	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5		17.74
5	Hyundai E(	Hyderabac	2012	75000	LPG	Manual	First	21.1 km/kg	814 CC	55.2 bhp	5		2.35
6	Nissan Mic	Jaipur	2013	86999	Diesel	Manual	First	23.08 kmp	1461 CC	63.1 bhp	5		3.5
7	Toyota Inn	Mumbai	2016	36000	Diesel	Automatic	First	11.36 kmp	2755 CC	171.5 bhp	8	21 Lakh	17.5
8	Volkswage	Pune	2013	64430	Diesel	Manual	First	20.54 kmp	1598 CC	103.6 bhp	5		5.2
9	Tata Indica	Chennai	2012	65932	Diesel	Manual	Second	22.3 kmpl	1248 CC	74 bhp	5		1.95
10	Maruti Cia	Kochi	2018	25692	Petrol	Manual	First	21.56 kmp	1462 CC	103.25 bhp	5	10.65 Lakh	9.95

- Containing 6019 rows and 14 columns for training and for testing there are 1234 rows and 13 columns the training and testing data are provided by the author separately.
- There are total 1876 unique cars data is available in training dataset
- This data is collected on 1998 to 2019 car models

6. Description of Work Done:

Steps Perform :
Data Accumulation → Data Preprocessing → Model Training → Model Testing → Visualization → Deployment

7. Literature Survey: Give the tabular form of 20 papers with the following columns information.

S. No	Title of paper Authors of paper	Name of journal/conference and date	Data Set name ans link	Data preprocessing techniques done(class imbalance, normalization, missing values handling etc)	Algorithms applied	Findings (Quantitative)
1	Used Car Price Prediction using K-Nearest Neighbor Based Model	International Journal of Innovative Research in Applied Sciences and	https://www.kaggle.com/datasets	The primary data preprocessing technique used here is feature engineering, which involves manipulating and	K-Nearest Neighbor (KNN)	Accuracy : 85%

	K. Samruddhi Dr. R. Ashok Kumar	Engineering (IJIRASE)		transforming raw data into a format suitable for machine learning algorithms.		
2	Predicting the Price of Used Cars using Machine Learning TechniquesSam eerchand Pudaruth	International Journal of Information & Computation Technology.	https://www.kaggle.com/datasets	• Normalization • Handling missing values • Data Reduction	Multiple Linear Regression Analysis K-Nearest Neighbors (KNN) Naive Bayes Decision Trees	Mean Error: Rs51,000
3	Prediction of The Prices of Second-Hand Cars Ozer Celik, U. Omer Osmanoglu	European Journal of Science and Technology	http://ikinciyeni.com/	• Data Collection • Data Labeling	Linear Regression Analysis	R-squared values ranged from 0.71 to 0.92.
4	Used car price prediction using linear regression model Ashutosh Datt Sharma*1, Vibhor Sharma*2	International Research Journal of Modernization in Engineering Technology and Science	http://Kaggle.com	• Null-Entry Removal • One-Hot Encoding • Train-test split • Feature Selection	Linear regression model	R 2 value of 0.86

5	Used Cars Price Prediction and Valuation using Data Mining Techniques Abdulla AlShared	RIT Digital Institutional Repository	Data was collected and Scrapped from a website BuyAnyCar	• Handling missing values • Converting Categorical data • Handling Outliers • Data Normalization	Random Forest Regressor Linear Regression Bagging Regressor	Random Forest Regressor: Accuracy: Achieved an accuracy of 95%. Mean Squared Error (MSE): 0.025. Mean Absolute Error (MAE): 0.0008. Root Mean Squared Error (RMSE): 0.03.
6	Price Prediction for Used Cars Marcus Collard	International Research Journal of Modernization in Engineering Technology and Science	http://kaggle.com/	• Data Cleaning and Normalization • Conversion of categorical variables to numeric	Linear Regression Ridge regression Lasso Regression Random Forest Regression	Random Forest Regression RMSE value of 4799 MAPE value of 37.65%
7	Pre-owned car price prediction by employing machine learning techniques Mauparna Nandan Debolina Ghosh	Journal of Decision Analytics and Intelligent Computing	https://www.google.com/url?	• Label Encoder • Data Normalization	Random Forest	MAE : 0.167132 MSE : 0.078840 RMSE : 0.078840 R2 Score : 0.867691 Accuracy : 86.769137
8	Advancing	2023	Data	• Data cleaning	Linear	Random Forest

	Used Car Price Prediction in South Africa: An Empirical Examination of Machine Learning Techniques Zenzele Abel Msiza,Pius Adewale Owolawi	International Conference on Artificial Intelligence and its Applications	obtained from Demo automobil es website	and Normalizatio n • Handling Outliers	Regression , Decision Tree, Random Forest, Gradient Boosted Trees Regressor, Artificial Neural Network, and K-Nearest Neighbors. The Random Forest method	(RF): R-squared value: 0.988 RMSE value: 0.019
9	Using Linear Regression For Used Car Price Prediction Sümeyra MUTİ1, Kazım YILDIZ2	International Journal of Computational and Experimental Science and Engineering	http://Kaggle.com	• Handling Missing or incorrect values • Handling outliers • Feature Transformati on	Linear regression model	R-squared value: 0.62

10	Used Car Price Prediction Using Machine Learning VELURU RANJITH	KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES Karunya Nagar, Coimbatore – 641 114. INDIA	https://www.kaggle.com/datasets/lepchenkov/usedcarscatalog	• Removing Outliers	Random Forest	Accuracy: 0.861908
11	Prediction of Used Car Prices using Machine Learning Techniques Eesha Pandit1, Hitanshu Parekh2, Pritam Pashte3, Aakash Natani4	International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 09 Issue: 12 | Dec 2022	http://Kaggle.com	• Feature Renaming • Feature Selection • Exploratory Data Analysis (EDA) • One-Hot Encoding • Correlation Analysis • Feature Allocation	Linear Regression Lasso Regression Ridge Regression Bayesian Ridge Regression Random Forest Regression	Random Forest Regression R-squared (r2) score of 0.95.
12	CAR PRICE PREDICTION USING MACHINE LEARNING TECHNIQUES Abishek R*1	International Research Journal of Modernization in Engineering Technology and Science	http://Kaggle.com	• Applied machine learning techniques to clean and pre-process the dataset. Removed missing values,	1. Simple Linear Regression 2. Multiple Linear Regression 3.Clustering Methods	f random forest model MAE : 1.522771460587 MSE : 10.49007991840 RMSE : 3.2388392856711

.				outliers, and irrelevant features.	(e.g., K-means) 4. Logistic Regression 5. K-nearest Neighbors (KNN) 6. Random Forest 7. Decision Tree	R-squared (r2) : 0.910486881527
13	Prediction of the price of used cars based on machine learning algorithms Yian Zhu	Proceedings of the 3rd International Conference on Signal Processing and Machine Learning	https://tianchi.aliyun.com/dataset/?lang=en-us	• Data cleaning - missing values - outliers - duplicate values • Data dimension reduction - Linear – pca - Non linear - ISOMAP • Feature selections	XGBoost SVM Neural network	R-squared (r2) score of 0.9823.

14	Used Cars Price Prediction using Supervised Learning Techniques Pattabiraman Venkatasubbu, Mukkesh Ganesh	International Journal of Engineering and Advanced Technology (IJEAT)	. The data was collected from the 2005 Central Edition of the Kelly Blue Book	• Applied machine learning techniques to clean and pre-process the dataset. Removed missing values, outliers, and irrelevant features.	Lasso Regression Multiple Regression Regression Tree	Error rate: Multiple regression: 3.468 %
15	Price Prediction of Used Cars Using Linear Regression 1Amit Kewat, 2Nitesh Kanojiya	Journal of Online Engineering Education	https://www.kaggle.com/datasets	• Data Cleaning and Normalization • Handling Outliers • Extracting numeric values	Linear Regression	Accuracy : 89%
16	Car Price Prediction using Machine Learning Techniques Enis Gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric	TEM Journal	autopijaca.ba	• Data Cleaning Skewed Class Removal Normalization Conversion of Continuous Attributes into Categorical	Random Forest (RF) Classifier Artificial Neural Network (ANN) Classifier Support Vector Machine	For the Cheap subset, SVM achieved the highest accuracy at 86.96%. For the Moderate subset, ANN performed better with an accuracy of 86.11%. For the Expensive subset, SVM achieved the highest accuracy at

.				Values Conversion of Regression Prediction Problem into Classification Problem	(SVM) Classifier	90.48%.
17	Used Car Price Prediction Using Random Forest Algorithm Prof. Dipti A. Gaikwad1 , Pratik S. Suwarnakar2 , Yash R. Mahajan3 , Amita U. Petkar4 , Shreyasi G. Theurkar5	International Journal for Multidisciplinary Research (IJFMR)	https://www.kaggle.com/datasets	• Data Cleaning • Feature Engineering • Normalization/Standardization • One-Hot Encoding • Train-Test Split	Linear Regression Lasso Regression Support Vector Machine (SVM) Random Forest	Random Forest R2 Score: 0.8697
18	Used Car Price Prediction Using Machine Learning Techniques Mrs Shyamali Das1 , Mr Ananta Laha2 , Mr Alok Jena3 , Ms Priyadarshini Samal4	International Journal of Research Publication and Reviews	https://www.kaggle.com/datasets	1. Data Cleaning 2. Encoding Categorical Variables 3. Feature Scaling 4. Feature Engineering 5. Handling Skewed Data 6. Train-Test Split	Linear Regression Lasso Regression Support Vector Machine (SVM) Random Forest	Accuracy by Implementation of Random Forest 91. 435 %

19	Used car price prediction Abhishek Jha, Dr. Ramveer Singh, Manish, Imran Saifi, Shipra Srivastava	International Journal of Advance Research, Ideas and Innovations in Technology	cardekho.com	• Removing Outliers	Random Forest Regression	Accuracy : 91.45%
20	Prediction of prices for used car by using regression models Nitis Monburinon, Prajak Chertchom, T. Kaewkiriya, Suwat Rungpheung, Sabir Buya, Pitchayakit Boonpou	2018 5th International Conference on Business and Industrial Research (ICBIR)	https://www.kaggle.com/datasets	• 1. Data Cleaning 2. Encoding Categorical Variables 3. Feature Scaling 4. Feature Engineering 5. Handling Skewed Data 6. Train-Test Split	Gradient Boosted Regression Trees Random Forest Regression Multiple Linear Regression	Gradient Boosted Regression Trees MSE : 0.28 highest accuracy

8. **Data Preprocessing (if any):**

- While observing the structure of the dataset we observed datatypes of some variables or columns are not suitable for our further processes and also some columns contain NA values.
- And some string type columns contain empty strings. Like New_Price, Mileage, Engine, Power etc
- There where total 42 rows containing "Na" values.
- And there are some columns which should be in datatype Integer or Double but they are present in dataset as a Character or String.

Columns -> Mileage, Engine, Power, New_Price

```
$ Mileage           : chr  "26.6 km/kg" "19.67 kmpl" "18.2 kmpl" "20.77 kmpl"
$ Engine            : chr  "998 CC" "1582 CC" "1199 CC" "1248 CC" ...
$ Power             : chr  "58.16 bhp" "126.2 bhp" "88.7 bhp" "88.76 bhp" ...
$ Seats             : num  5 5 5 7 5 5 5 8 5 5 ...
$ New_Price         : chr  "" "" "8.61 Lakh" "" ...
```

- And above columns also contain some units in suffix which is not very useful for use in further process.

- And the New_Price column consists lots of Empty strings and so after converting it into integer or float the empty string denoted by "Na" and due to high number of Na values in this column we should impute this column.

So above are the some impurities or problems in our dataset and we clean and solve those as follows:

1) We first delete or remove the 42 rows from the dataset which contain "Na" values at start.
2) After that we convert all string type columns like Mileage, Engine, Power into double and also remove the suffix from it.
3) And also Imputed the New_Price values and converted into double datatype Imputation is done using rfImpute() function present in randomforest library uses Proximity Matrix concept to impute values

```
$ Mileage          : num  26.6 19.7 18.2 20.8 15.2 ...
$ Engine           : num  998 1582 1199 1248 1968 ...
$ Power            : num  58.2 126.2 88.7 88.8 140.8 ...
$ Seats            : num  5 5 5 7 5 5 5 8 5 5 ...
$ New_Price        : num  5.06 14.03 8.61 13.03 49.07 ...
```

4) Now our data is ready for further process after data preprocessing there are total 5977 rows are remain in the training data.
5) Similarly we perform same data preprocessing on test data and there are 1192 rows are remain in the testing data.

9. Feature Selection (if any): *Explain the different feature selection techniques you have used in the project*

10. Algorithms Implemented:

- As our problem statement is of Regression type so we are using following models :
  1) Multiple Linear Regression
  2) Decision Tree
  3) Random Forest
  4) SVM (Support Vector Machine as Regressor)

- Before training each model we perform some common processes for better results and accuracy
  1) As our car's names are too long which causes problems while training since we consider them as factors or categories so we first make it short till 2 String.
  2) And as we reduce the car name string we add one more extra feature into the data which is "Brand" which denotes the brand of the car and we found there are 32 different Brands of Cars present in the entire dataset.
  3) We convert some columns like Name, Brand, Owner_Type, Transmission, Location etc into a factor or category.

4) And first separate out the price column from the testing dataset and as the dataset is already divided into 75:25 ratio for training and testing respectively.
5) And after all of this we send the training data to a different model.

- **Multiple Linear Regression:**

Here we use "lm()" function to train the linear model

```
lm_model <- lm(Price ~ ., data = train_data)
```

Here we train our model on all columns for price prediction.
And after training of Model we predict the price of cars present in test data.

```
predicted_price <- predict(lm_model, newdata = test_data_filtered)
```

After Prediction we test model Accuracy using R^2, RMSE & MAE values.

```
[1] "Mean Absolute Error (MAE): 2.34933991485649"
[1] "Root Mean Squared Error (RMSE): 4.24773899332807"
[1] "R-squared-byModel: 0.851410356965969"
[1] "R-squared-byCalculation: 0.839165597575994"
```

- **Decision Tree**

Here we use "rpart()" function to train the decision tree model

```
dt_model <- rpart(Price ~ ., data = train_data, method = "anova")
```

Here we train our model on all columns for price prediction.

The method specified as "anova" indicates that the decision tree will employ analysis of variance to determine the splits at each node during the tree-building process.
And after training of Model we predict the price of cars present in test data.

```
predicted1_price <- predict(dt_model, newdata = test_data_filtered)
```

After Prediction we test model Accuracy using R^2, RMSE & MAE values

```
[1] "Mean Absolute Error (MAE): 2.89320014588467"
[1] "Root Mean Squared Error (RMSE): 4.77760670728731"
[1] "R-squared: 0.796537631947437"
```

- **Random Forest**

Here we use "randomForest()" function to train the random forest model

```
rf_model <- randomForest(Price ~ ., data = train_data,iter=300)
```

Here we train our model on all columns for price prediction.

The parameter iter = 300 specifies the number of trees to be grown in the random forest ensemble, with 300 trees being grown in this case.

And after training of Model we predict the price of cars present in test data.

```
predicted_price_rf <- predict(rf_model, newdata = test_data_filtered)
```

After Prediction we test model Accuracy using R^2, RMSE & MAE values

```
[1] "Mean Absolute Error (MAE) with Random Forest: 1.24104022403177"
[1] "Root Mean Squared Error (RMSE) with Random Forest: 2.4677110045158"
[1] "R-squared with Random Forest: 0.949241942222172"
```

- **SVM**

Here we use "svm()" function to train the SVM model

```
svr_model <- svm(Price ~ ., data = train_data, na.action = na.omit, scale = TRUE,kernel = 'radial')
```

Here we train our model on all columns for price prediction.

The parameter na.action = na.omit specifies the action to take if there are missing values in the data, instructing the model to omit observations with missing values. The parameter scale = TRUE indicates that the predictors should be scaled to have zero mean and unit variance, which is a common preprocessing step in SVM models to ensure that all features are on a similar scale

The kernel function used in this SVM model is specified as 'radial', which denotes a radial basis function (RBF) kernel. RBF kernels are commonly used in SVM models for non-linear regression tasks as they can effectively capture complex relationships between predictors and the target variable.

And after training of Model, we predict the price of cars present in test data.

```
predicted_price_svr <- predict(svr_model, newdata = test_data_filtered)
```

After Prediction, we test model Accuracy using R^2, RMSE & MAE values

```
[1] "Mean Absolute Error (MAE): 1.90599776146066"
[1] "Root Mean Squared Error (RMSE): 4.19691293029233"
[1] "R-squared-byCalculation: 0.842991479190409"
```

Model	MAE	RMSE	R-Squared
Multiple Linear Regression	2.3493399	4.2477389	0.85141035
Decision Tree	2.920059	4.766653	0.8014424
Random Forest	1.24104022	2.46771	0.94924194
SVM (radial-kernel)	1.9045498	4.1931021	0.843276

After Checking Significance of each X-variables we get to know that "Seats" variable have least significance that of others

```
(Intercept)                 0.00175 **
Age                         < 2e-16 ***
Kilometers_Driven           0.00159 **
```

```
Owner_TypeSecond            0.00252 **
Owner_TypeThird             0.15400
Mileage                    1.58e-05 ***
Engine                     1.38e-06 ***
Power                       < 2e-16 ***
Seats                       0.35061
```

But After removing "Seats" the accuracy or R^2 value decreases

Model	MAE	RMSE	R-Squared
Multiple Linear Regression	2.87381	4.97619	0.766461

11. Code:

Model Training code:

```
library(caret)
  library(randomForest)
  library(dplyr)

  data<-read.csv("Final_TrainingDataSet.csv")
```

```
  # Remove unnecessary columns
  data <- data[, !(names(data) %in% c("New_Price", "X","Year"))]

  # Extract only the first string from the Name column
  # data$Name <- sapply(strsplit(data$Name, " "), function(x)
paste(x[1:min(length(x), 2)], collapse=" "))
  data$Brand <- sapply(strsplit(data$Name, " "), function(x)
paste(x[1:min(length(x), 1)], collapse=" "))

  data <- data[, !(names(data) %in% c("Name","X.1"))]


  # Convert required columns to factor
  # This column has more than 53 levels or categories


  data$Location <- as.factor(data$Location)
  data$Fuel_Type <- as.factor(data$Fuel_Type)
  data$Transmission <- as.factor(data$Transmission)
  data$Owner_Type <- as.factor(data$Owner_Type)
  data$Brand<- as.factor(data$Brand)

  # Divide dataset into training and testing (75% train, 25% test)
  set.seed(123)  # for reproducibility
  pd <- sample(2, nrow(data), replace = TRUE, prob = c(0.75,0.25))
  train_data <- data[pd == 1, ]
  test_data <- data[pd == 2, ]

  write.csv(train_data, file = "train_data.csv")

  Price = test_data$Price
  test_data <- test_data[, !(names(data) %in% c("Price"))]

  # Convert categorical variables to factors with levels from training data
  test_data$Location <- factor(test_data$Location, levels =
levels(train_data$Location))
  test_data$Fuel_Type <- factor(test_data$Fuel_Type, levels =
levels(train_data$Fuel_Type))
  test_data$Transmission <- factor(test_data$Transmission, levels =
levels(train_data$Transmission))
  test_data$Owner_Type <- factor(test_data$Owner_Type, levels =
levels(train_data$Owner_Type))
  test_data$Brand <- factor(test_data$Brand, levels = levels(train_data$Brand))


  # names(test_data)
  # is.numeric(test_data$Year)
```

```
# is.numeric(test_data$Age)
# is.numeric(test_data$Kilometers_Driven)
# is.numeric(test_data$Mileage)
# is.numeric(test_data$Engine)
# is.numeric(test_data$Power)
# is.numeric(test_data$Seats)
# is.factor(test_data$Location)
# is.factor(test_data$Fuel_Type)
# is.factor(test_data$Transmission)
# is.factor(test_data$Owner_Type)
# is.factor(test_data$Brand)




# Train a Random Forest model for price prediction
rf_model <- randomForest(Price ~ ., data = train_data,iter=300)

# Predict price using test data
predicted_price_rf <- predict(rf_model, newdata = test_data)

# Evaluate the model
# Calculate Mean Absolute Error (MAE)
MAE_rf <- mean(abs(predicted_price_rf - Price))

# Calculate Root Mean Squared Error (RMSE)
RMSE_rf <- sqrt(mean((predicted_price_rf - Price)^2))

# Calculate R-squared value
R_squared_rf <- cor(predicted_price_rf, Price)^2

print(paste("Mean Absolute Error (MAE) with Random Forest:", MAE_rf))
print(paste("Root Mean Squared Error (RMSE) with Random Forest:", RMSE_rf))

print(paste("R-squared with Random Forest:", R_squared_rf))

saveRDS(rf_model, file = "random_forest_model.rds")
```

Shiny Training Code :

```
library(shiny)
library(shinydashboard)
library(DT)
library(randomForest)  # Assuming you trained your model using randomForest
```

```
# Load your trained model
price = "price"


car_data <- data.frame(
  car_name = c("Corolla", "Civic", "F-150", "Elantra", "Camaro"))


# Define UI for application
ui <- dashboardPage(
  dashboardHeader(
    title = div(
      "Used Car Price Prediction",
      tags$style(HTML("font-size: 24px;"))
    )
  ),
  dashboardSidebar(
    # Input fields
    width = "30%",
    tags$head(
      tags$style(
        HTML(".sidebar .form-group.shiny-input-container {
          width: 90%;
          align-item:center;
          margin-left:2rem;
        }
        .img
        {
        margin-bottom:3rem;
        }
        "
        )
      )
    ),
    sidebarMenu(
      menuItem("Home", tabName = "home", icon = icon("home")),
      numericInput("year" ,"Year", min = 1900, max = 2019, value = 2015),
      selectInput("brand", "Brand of Car",
                  choices <- c("Maruti", "Hyundai", "Honda", "Audi",
                               "Nissan", "Toyota", "Volkswagen", "Tata", "Land",
                               "Mitsubishi", "Renault", "Mercedes-Benz", "BMW",
                               "Mahindra", "Ford", "Porsche", "Datsun",
"Jaguar",
                               "Volvo", "Chevrolet", "Skoda", "Mini", "Fiat",
"Jeep",
                               "Smart", "Ambassador", "Isuzu", "Force",
"Bentley",
```

```
                              "Lamborghini")),
      selectInput("location", "Location of Car",
                  choices <- c("Mumbai", "Pune", "Chennai", "Coimbatore",
"Hyderabad", "Jaipur", "Kochi", "Kolkata", "Delhi", "Bangalore", "Ahmedabad")),
      numericInput("kilometer", "Kilometer Driven", value = 0),
      selectInput("fuel", "Fuel Type", choices = c("Petrol", "Diesel", "CNG",
"LPG")),
      selectInput("transmission", "Transmission", choices = c("Manual",
"Automatic")),
      selectInput("owner", "Owner Type",
                  choices = c("First", "Second", "Third", "Fourth", "Test Drive
Car")),
      numericInput("mileage", "Mileage (kmpl)", value = 0),
      numericInput("power", "Power (bhp)", value = 0),
      numericInput("engine", "Engine (CC)", value = 0),
      numericInput("seats", "Number of Seats", value = 0),
      selectInput("car_name", "Name of Car", choices = car_data$car_name)
    )
  ),
  dashboardBody(
    # Main panel for displaying results and the car image
    tabItems(
      tabItem(tabName = "home",
              fluidRow(
                box(
                  title = "Prediction",
                  status = "primary",
                  solidHeader = TRUE,
                  width = 12,
                  height = "50%",
                  DTOutput("prediction")
                ),
                box(
                  title = "Car Image",
                  status = "primary",
                  solidHeader = TRUE,
                  width = 12,
                  height = "50%",
                  div(
                    style = "display: flex; justify-content: center; align-
items: flex-start;margin-bottom:6rem;margin-left:10rem;",
                    imageOutput("carImage")
                  )
                ),
                box(
                  title = "Price Prediction",
                  status = "primary",
```

```
                  solidHeader = TRUE,
                  width = 12,
                  height = "50%",
                  textOutput("formatted_price"),


                )
              )
      )
    )
  )
)


# Define server logic
server <- function(input, output) {

  car_data <- data.frame(
    car_name = c("Corolla", "Civic", "F-150", "Elantra", "Camaro"),
    image_file = c("corolla.jpg", "civic.jpg", "f150.jpg", "elantra.jpg",
"camaro.jpg")
  )

  # Server logic for prediction
  output$prediction <- renderDT({
    # Creating a data frame with parameters and values
    data <- data.frame(
      "Parameters" = c("Name of Car", "Year", "Brand", "Kilometer Driven", "Fuel
Type", "Transmission", "Owner Type", "Mileage", "Power", "Engine", "Seats",
"Price"),
      "Values" = c(input$car_name, input$year, input$brand, input$kilometer,
input$fuel, input$transmission, input$owner, input$mileage, input$power,
input$engine, input$seats, "A"),
      stringsAsFactors = FALSE
    )

    # Highlighting the row with "Price" equal to "A"
    data$Parameters <- ifelse(data$Parameters == "Price", price,
data$Parameters)

    # Returning the data frame as a datatable
    datatable(data, rownames = FALSE, options = list(
      columnDefs = list(
        list(targets = "_all", className = "valueColumn")
      )
    ))
  })
```

```
  # Dynamically render car image based on the selected car name
  output$carImage <- renderImage({
    # Get the selected car name
    selected_car <- input$car_name

    # Find the corresponding image file name based on the selected car name
    image_file <- car_data$image_file[car_data$car_name == selected_car]

    # If image file name is found, render the image
    if (!is.na(image_file) && file.exists(paste0("www/", image_file))) {
      list(src = paste0("www/", image_file), width = "70%" )
    } else {
      # If image file is not found, display a placeholder image
      list(src = "www/placeholder.jpg", width = "65%")
    }
  }, deleteFile = FALSE)

  # Predict price using the trained model
  output$formatted_price <- renderText({
    # Prepare input data for prediction
    # Create new_data dataframe with proper data types and levels
    new_data <- data.frame(
      Age = as.integer(2024 - input$year),  # Corrected the calculation of Age
      Location = factor(input$location, levels = levels(train_data$Location)),
      Kilometers_Driven = as.integer(input$kilometer),
      Fuel_Type = factor(input$fuel, levels = levels(train_data$Fuel_Type)),
      Transmission = factor(input$transmission, levels =
levels(train_data$Transmission)),
      Owner_Type = factor(input$owner, levels = levels(train_data$Owner_Type)),
      Mileage = as.numeric(input$mileage),
      Engine = as.integer(input$engine),
      Power = as.numeric(input$power),
      Seats = as.integer(input$seats),
      Brand = factor(input$brand, levels = levels(train_data$Brand))
    )

    cat("\n",names(new_data),"\n")
    cat(is.numeric(new_data$Year))
    cat(is.numeric(new_data$Age))
    cat(is.numeric(new_data$Kilometers_Driven))
    cat(is.numeric(new_data$Mileage))
    cat(is.numeric(new_data$Engine))
    cat(is.numeric(new_data$Power))
    cat(is.numeric(new_data$Seats))
    cat(is.factor(new_data$Location))
    cat(is.factor(new_data$Fuel_Type))
```

```
    cat(is.factor(new_data$Transmission))
    cat(is.factor(new_data$Owner_Type))
    cat(is.factor(new_data$Brand))








    # Make prediction using the loaded model
    predicted_price <- predict(mymodel, new_data)
    predicted_price = round(predicted_price,2)
    # Format the predicted price with a range of +/- 2 Lacs
    formatted_price <- paste(predicted_price, "Lacs")

    # Return the formatted predicted price
    formatted_price
  })
}


# Run the application
shinyApp(ui = ui, server = server)
```

12. Shiny App

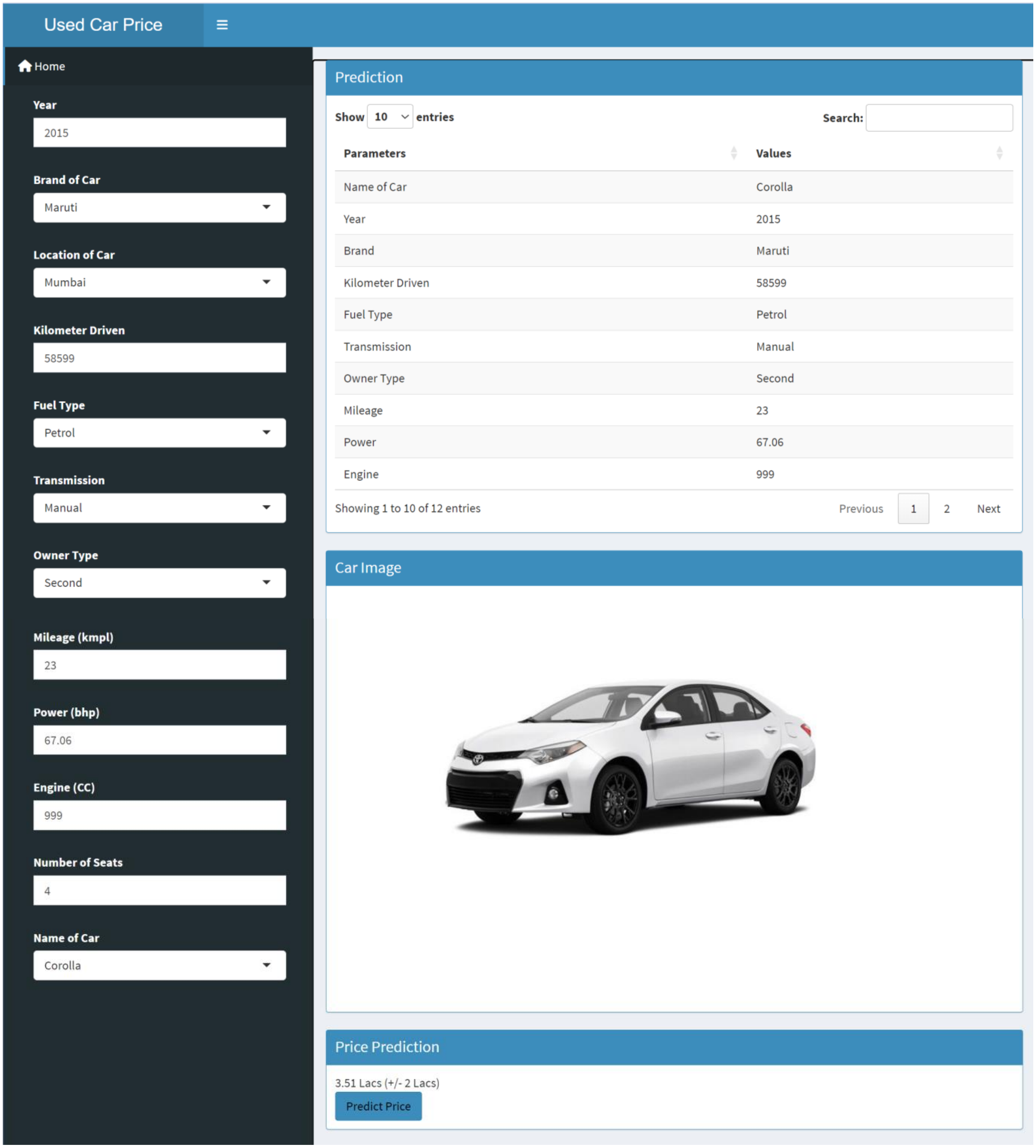


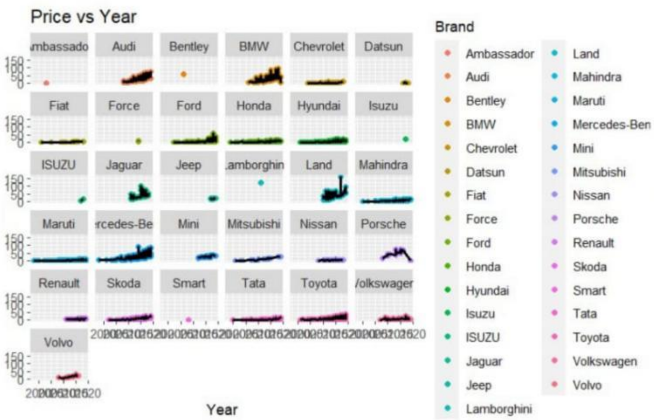
Fig ; Interface of Shiny app

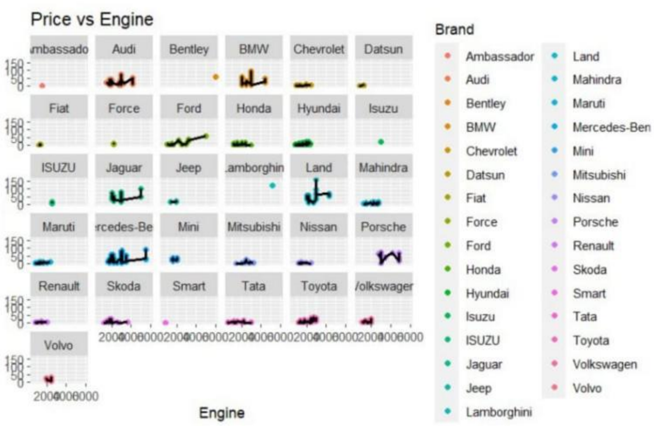
13. Evaluation Parameters : *Explain which evaluation parameters you have used in your project***.**

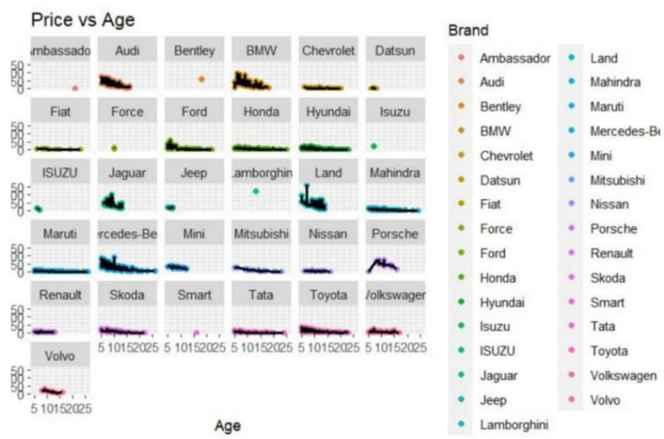
14. Results and Discussions:

Data Visualization :

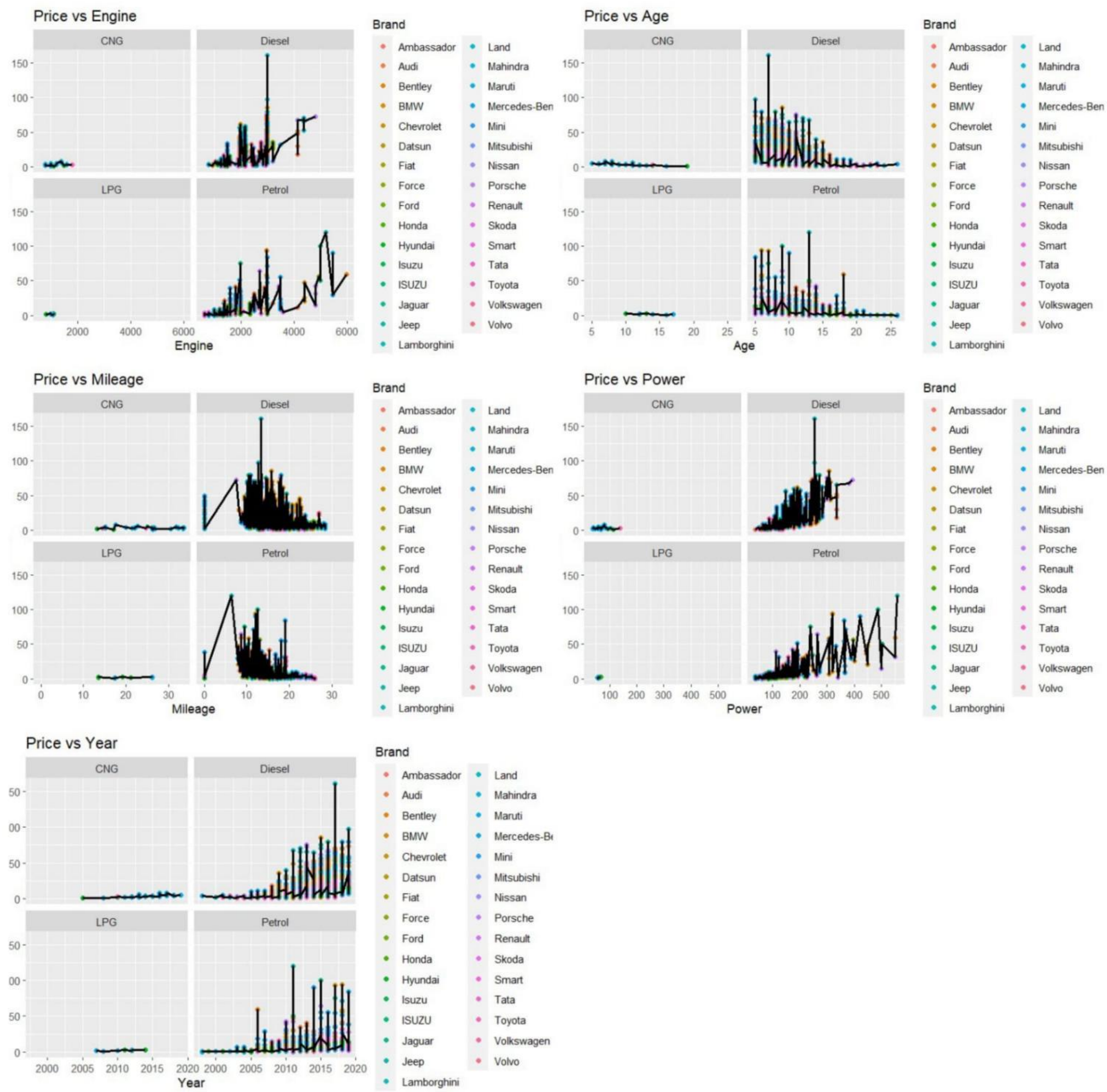
1.



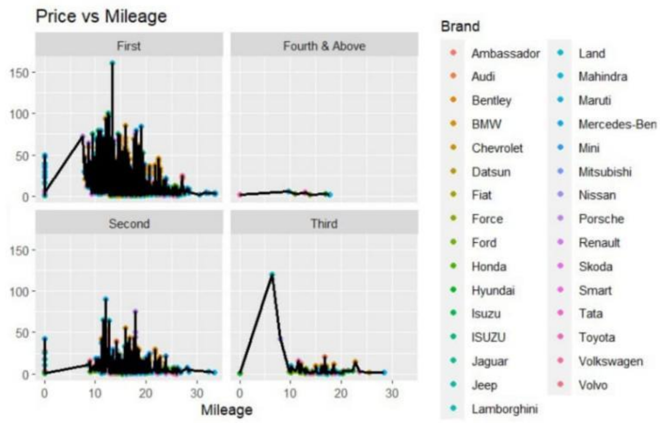


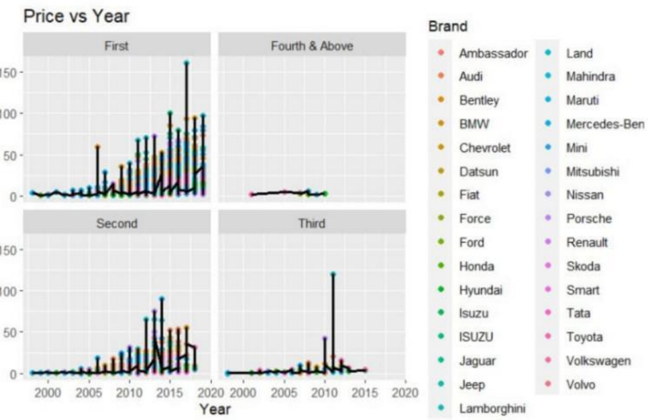


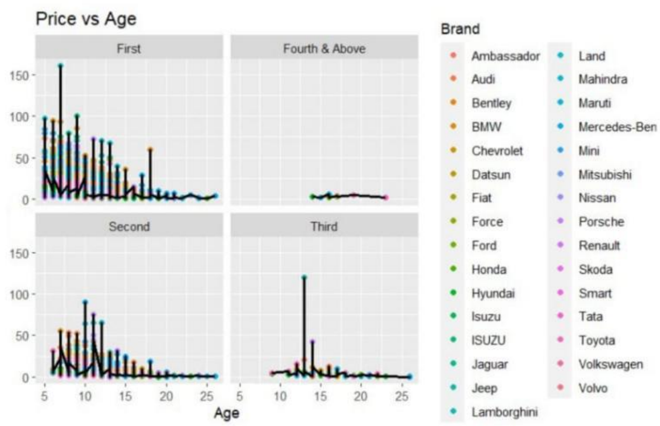
2.

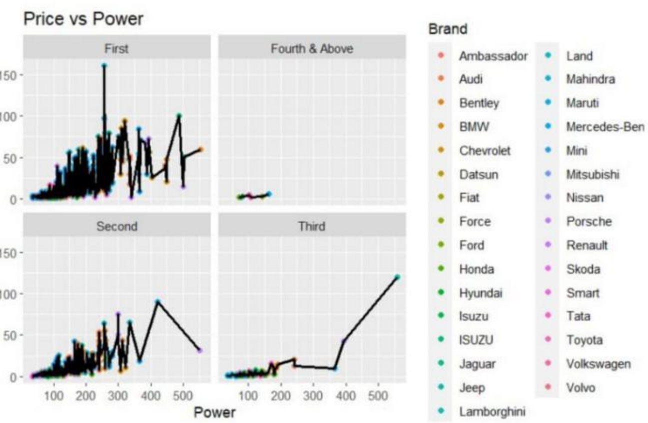

Price vs Engine
Price vs Age
Price vs Mileage
Price vs Power
Price vs Year
CNG
Diesel
LPG
Petrol
Engine
Age
Mileage
Power
Year
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Ben
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

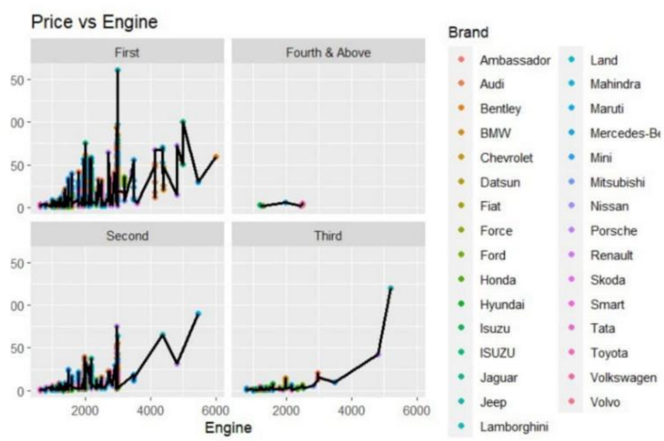
3.


Price vs Mileage
First
Fourth & Above
Second
Third
Mileage
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Ben
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

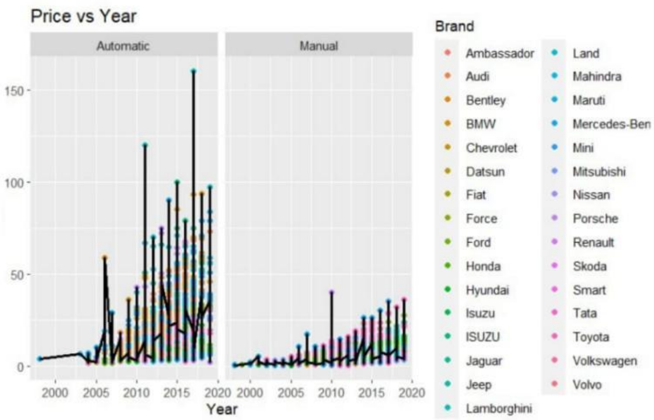

Price vs Year
First
Fourth & Above
Second
Third
Year
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Ben
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

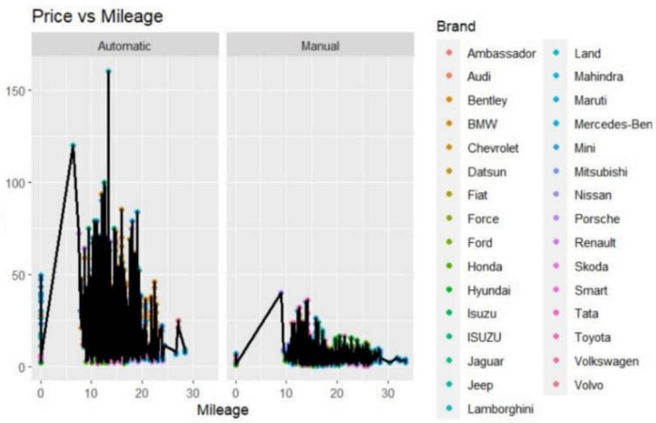

Price vs Age
First
Fourth & Above
Second
Third
Age
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Ben
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

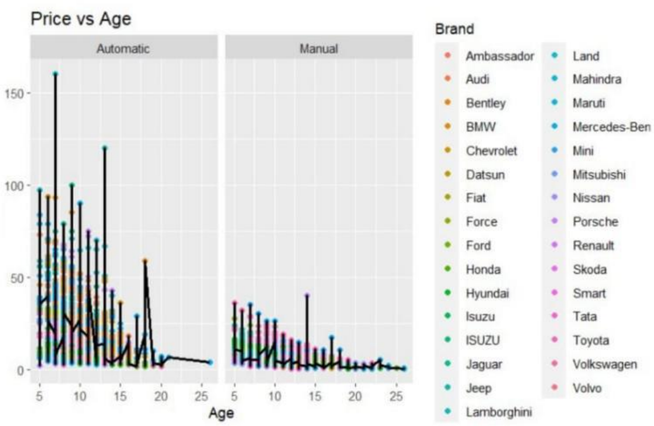

Price vs Power
First
Fourth & Above
Second
Third
Power
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Ben
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

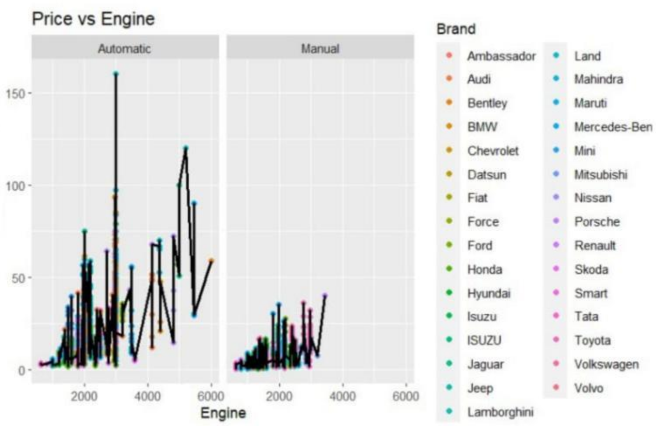

Price vs Engine
First
Fourth & Above
Second
Third
Engine
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Be
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

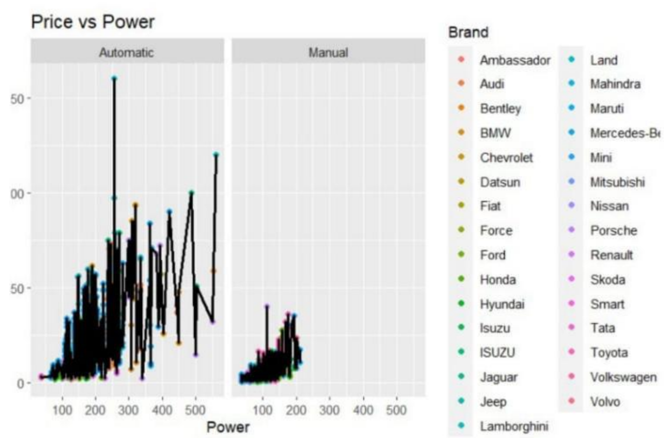
4.


Price vs Year
Automatic
Manual
Year
Brand
Ambassador
Audi
Bentley
BMW
Chevrolet
Datsun
Fiat
Force
Ford
Honda
Hyundai
Isuzu
ISUZU
Jaguar
Jeep
Lamborghini
Land
Mahindra
Maruti
Mercedes-Ben
Mini
Mitsubishi
Nissan
Porsche
Renault
Skoda
Smart
Tata
Toyota
Volkswagen
Volvo

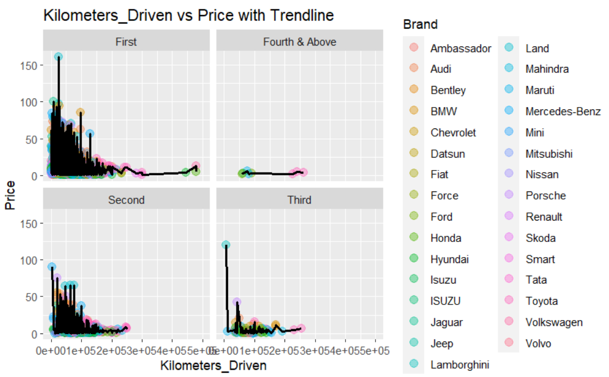

Price vs Mileage
Automatic
Manual
Mileage
Brand

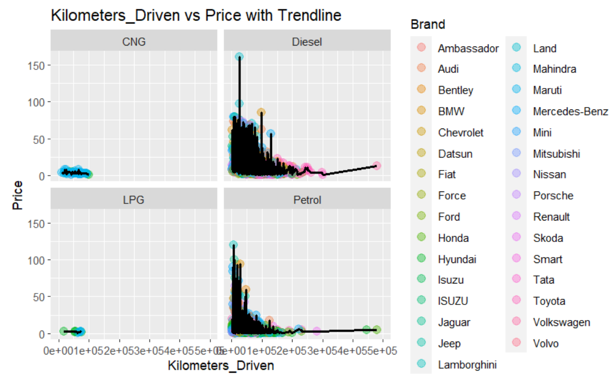

Price vs Age
Automatic
Manual
Age
Brand

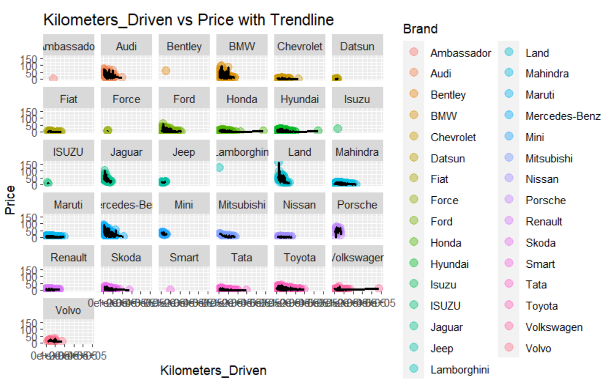

Price vs Engine
Automatic
Manual
Engine
Brand

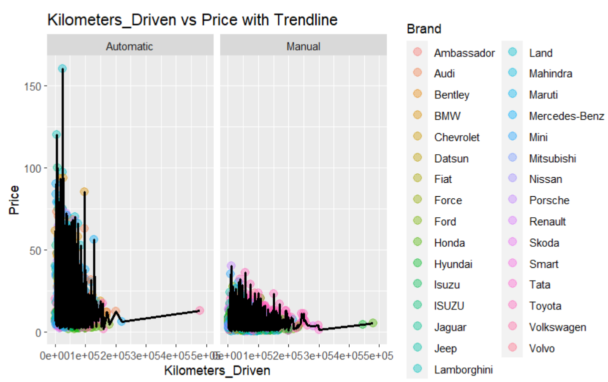

Price vs Power
Automatic
Manual
Power
Brand

5.









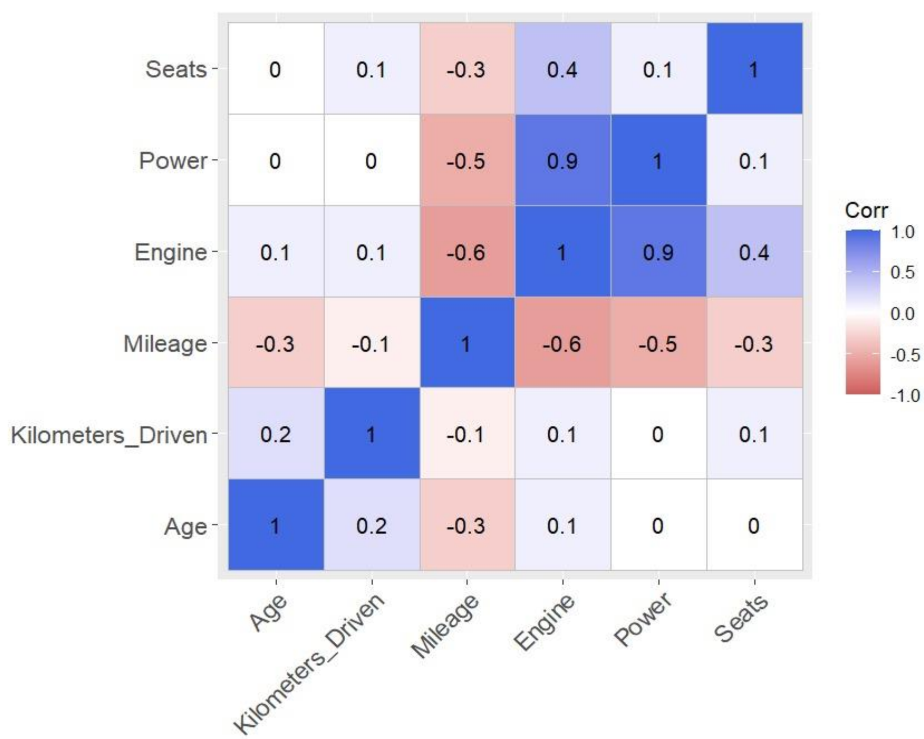
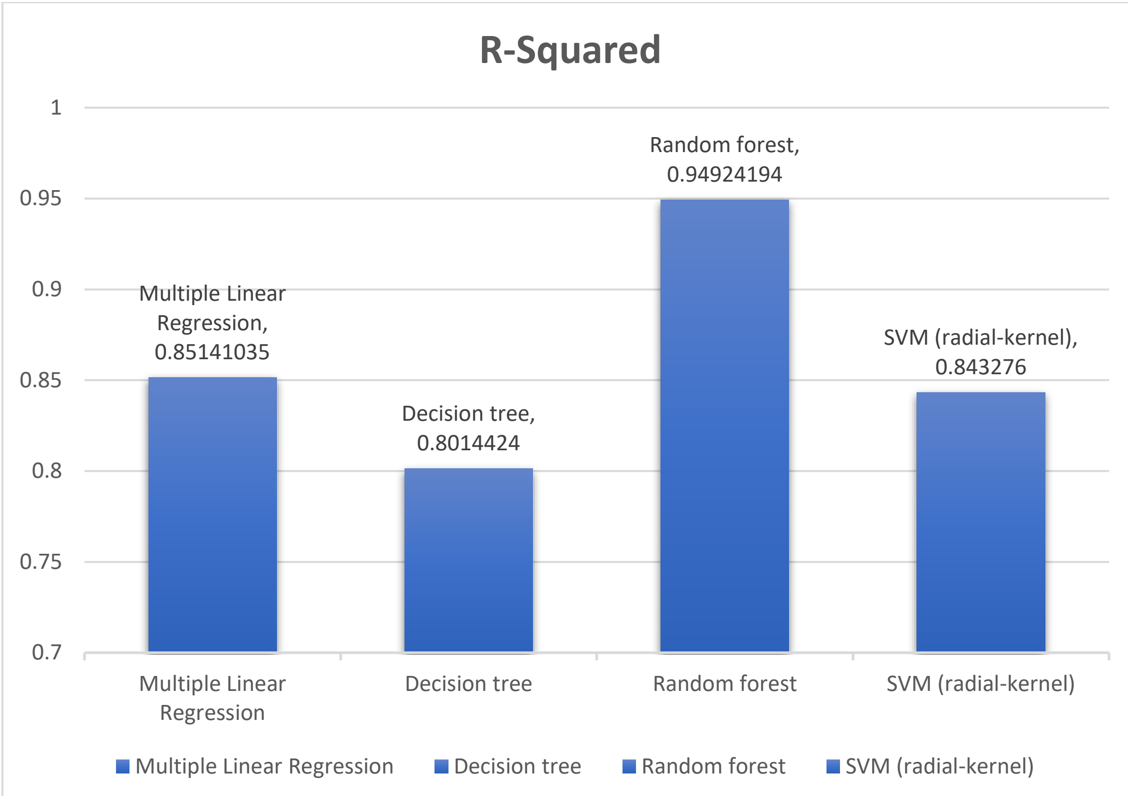
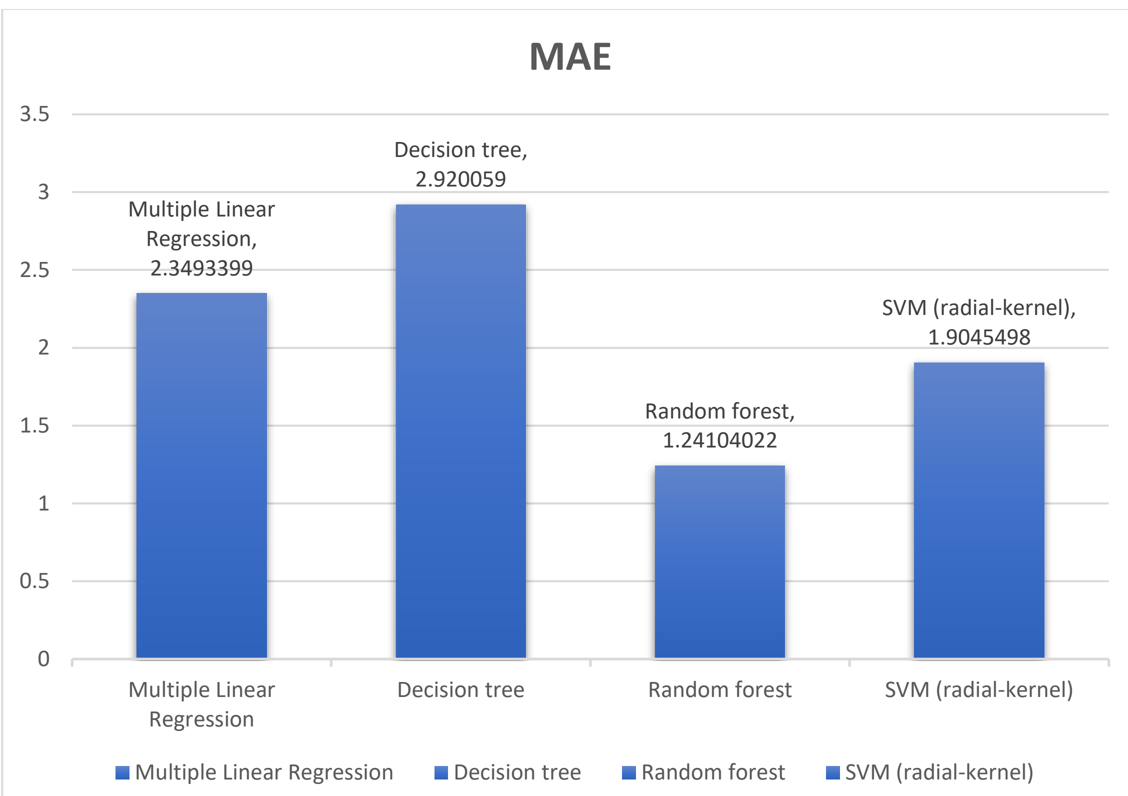
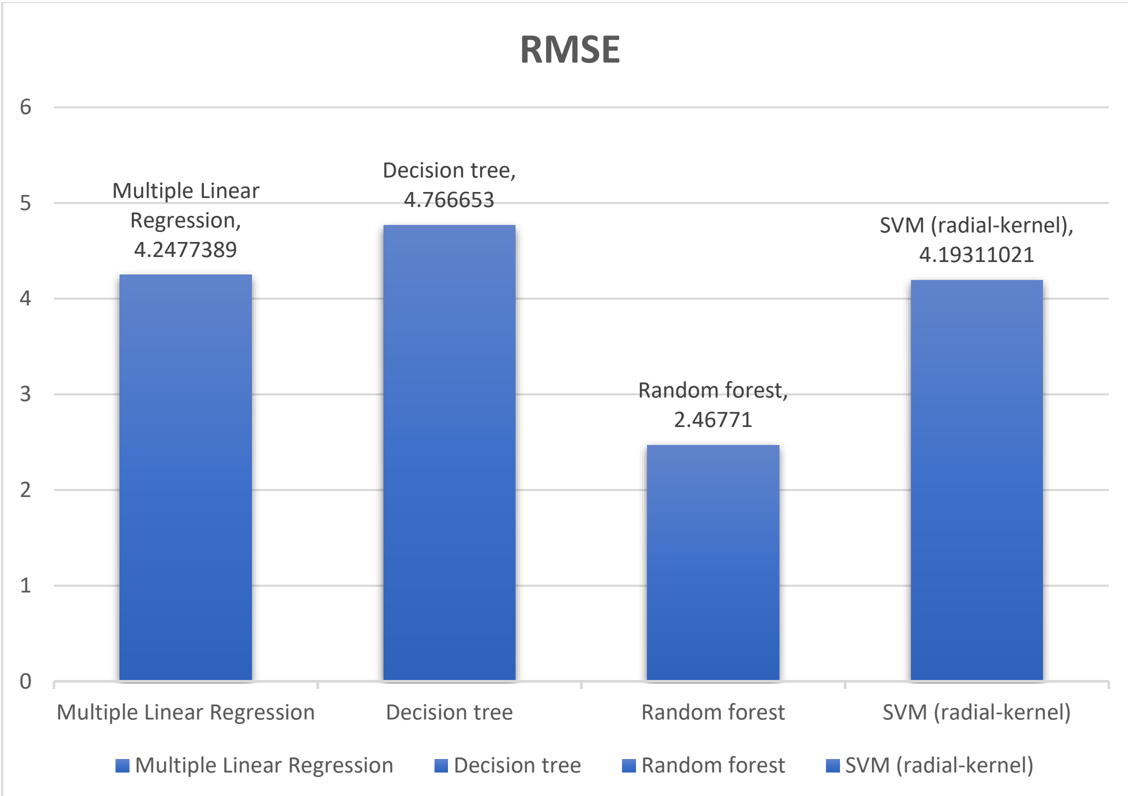


Fig : Feature Correlation Diagram


R-Squared
1
0.95
0.9
0.85
0.8
0.75
0.7
Multiple Linear Regression, 0.85141035
Decision tree, 0.8014424
Random forest, 0.94924194
SVM (radial-kernel), 0.843276
Multiple Linear Regression
Decision tree
Random forest
SVM (radial-kernel)
Multiple Linear Regression
Decision tree
Random forest
SVM (radial-kernel)


MAE
3.5
3
2.5
2
1.5
1
0.5
0
Multiple Linear Regression, 2.3493399
Decision tree, 2.920059
Random forest, 1.24104022
SVM (radial-kernel), 1.9045498
Multiple Linear Regression
Decision tree
Random forest
SVM (radial-kernel)
Multiple Linear Regression
Decision tree
Random forest
SVM (radial-kernel)



15 .Conclusions:

After conducting a comprehensive evaluation of various machine learning algorithms, it is evident that Random Forest emerges as the optimal choice for the predictive modeling task at hand. Random Forest surpasses Linear Regression, Decision Tree, and SVM (with radial) in accuracy. Its superiority is confirmed by lower MAE and RMSE, along with a higher R-squared value.

The values of metrics are :

Random Forest: MAE: 1.24104022 RMSE: 2.46771 R-Squared: 0.94924194 Based on this, the Random Forest algorithm has the lowest Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), and it also achieves the highest R-squared value. Therefore, it appears to be the most accurate algorithm among the ones tested.

References:

[1] Kriswantara, B., & Sadikin, R. (2022). Used Car Price Prediction with Random Forest Regressor Model. *Journal of Information Systems, Informatics and Computing Issue Period*, *6*(1), 40–49.

[2] MUTİ, S., & YILDIZ, K. (2023). Using Linear Regression For Used Car Price Prediction. *International Journal of Computational and Experimental Science and Engineering*, *9*(1), 11–16.

[3] Gupta, V., M.L, S., & K.C, T. (2021). USED CAR PRICE PREDICTION. *International Journal of Multidisciplinary Advanced Scientific Research and Innovation*, *1*(10), 256–262.

[4] -, D. A. G., -, P. S. S., -, Y. R. M., -, A. U. P., & -, S. G. T. (2023). Used Car Price Prediction Using Random Forest Algorithm. *International Journal For Multidisciplinary Research*, *5*(3).

[5] Cui, B., Ye, Z., Zhao, H., Renqing, Z., Meng, L., & Yang, Y. (2022). Used Car Price Prediction Based on the Iterative Framework of XGBoost+LightGBM. *Electronics (Switzerland)*, *11*(18).

[6] S, R., R, B. T., T, B. G., Hegde, R. P., & Ramesh, S. (2023). Used Car Price Prediction Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, *11*(5), 1176–1180.

[7] Dahiya, H., Aggarwal, C., Goyal, S., & Agarwal, M. (2021). USED CAR PRICE PREDICTION USING MACHINE LEARNING. *International Journal of Multidisciplinary Advanced Scientific Research and Innovation*, *1*(10), 246–251.

[8] Varshitha, J., Jahnavi, K., & Lakshmi, C. (2022). Prediction Of Used Car Prices Using Artificial Neural Networks And Machine Learning. In *2022 International Conference on Computer Communication and Informatics, ICCCI 2022*. Institute of Electrical and Electronics Engineers Inc.

[9] Alhakamy, A., Alhowaity, A., Alatawi, A. A., & Alsaadi, H. (2023). Are Used Cars More Sustainable? Price Prediction Based on Linear Regression. *Sustainability*, *15*(2), 911.

[10] Huang, J., Saw, S. N., Feng, W., Jiang, Y., Yang, R., Qin, Y., & Seng, L. S. (2023). A Latent Factor-Based Bayesian Neural Networks Model in Cloud Platform for Used Car Price Prediction. IEEE Transactions on Engineering Management.

[11] PREDICTION PRICE OF USED CARS. (2023). International Research Journal of Modernization in Engineering Technology and Science. https://doi.org/10.56726/irjmets33331

[12] Shaprapawad, S., Borugadda, P., & Koshika, N. (2023). Car Price Prediction:An Application of Machine Learning. 6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings. https://doi.org/10.1109/ICICT57646.2023.10134142

[13] Zhu, Y. (2023). Prediction of the price of used cars based on machine learning algorithms. Applied and Computational Engineering, 6(1). https://doi.org/10.54254/2755-2721/6/20230917

[14] Venkatasubbu P. , Ganesh M. Used Cars Price Prediction using Supervised Learning Techniques . International Journal of Engineering and Advanced Technology (2019)

[15] 1Amit Kewat , Nitesh Kanojiya . Price Prediction of Used Cars Using Linear Regression. Journal of Online Engineering Education

[16] Gegic, E., Isakovic, B., Keco, D., Masetic, Z., & Kevric, J. (2019). Car price prediction using machine learning techniques. TEM Journal, 8(1). https://doi.org/10.18421/TEM81-16

[17] -, D. A. G., -, P. S. S., -, Y. R. M., -, A. U. P., & -, S. G. T. (2023). Used Car Price Prediction Using Random Forest Algorithm. International Journal For Multidisciplinary Research, 5(3). https://doi.org/10.36948/ijfmr.2023.v05i03.3308

[18] Mrs Shyamali Das1 , Mr Ananta Laha2 , Mr Alok Jena3 , Ms Priyadarshini Samal4. Used Car Price Prediction Using Machine Learning Techniques. International Journal of Research Publication and Reviews

[19] **Cui, B., Ye, Z., Zhao, H., Renqing, Z., Meng, L., & Yang, Y. (2022). Used Car Price Prediction Based on the Iterative Framework of XGBoost+LightGBM. Electronics (Switzerland), 11(18). https://doi.org/10.3390/electronics11182932**

[20] **Monburinon, N., Chertchom, P., Kaewkiriya, T., Rungpheung, S., Buya, S., & Boonpou, P. (2018). Prediction of prices for used car by using regression models. Proceedings of 2018 5th International Conference on Business and Industrial Research: Smart Technology for Next Generation of Information, Engineering, Business and Social Science, ICBIR 2018. https://doi.org/10.1109/ICBIR.2018.8391177**