# Laptop Price Analysis

## Project Report

Submitted to

**UNIFIED MENTOR**

Submitted by

**Mr. Abdul Razzaq**

# Acknowledgement

I would like to express my sincere gratitude to my faculty guide for their continuous guidance, valuable suggestions, and encouragement throughout the completion of this project. I am also thankful to my institution for providing the necessary resources and learning environment to carry out this work. Finally, I would like to acknowledge the open-source R community for the libraries and tools that made data analysis and model development possible.

# Table of Contents

# Chapter 1

## Abstract

With the rapid growth of the laptop market, consumers are presented with a wide range of choices differing in brand, specifications, and price. However, laptop pricing is influenced by multiple technical and design-related factors such as processor type, RAM, storage, screen resolution, and operating system, making it difficult for buyers and businesses to understand which features truly drive price variations.

The problem addressed in this project is to analyse laptop specifications and identify the key factors influencing laptop prices, and to build predictive models that can accurately estimate laptop prices based on their features. This analysis can support data-driven decision-making for consumers, retailers, and manufacturers.

## Objectives

This project applies statistical analysis and machine learning techniques in R to explore laptop price determinants and build reliable predictive models. The primary objective of this project is to analyse laptop price patterns and determine how different hardware and software specifications impact the overall price.

**Specific objectives include:**

- To perform exploratory data analysis (EDA) to understand the distribution and characteristics of laptop prices.
- To analyse the relationship between laptop specifications (RAM, processor, storage, screen type, etc.) and price.
- To preprocess and transform the dataset to make it suitable for machine learning models.
- To build and evaluate multiple regression-based machine learning models for price prediction.
- To compare model performance using appropriate evaluation metrics.
- To identify the most influential features affecting laptop prices.

# Chapter 2

## Data Collection

The dataset used for this project is: Laptop Prices CSV

The dataset was provided in CSV format and imported into **RStudio** for preprocessing and analysis. Before beginning the modelling process, the dataset was examined for missing values, inconsistencies, and data types to ensure that it was clean and suitable for machine learning.

### Dataset Size

- **Total observations:** 1,275 laptops
- **Total features:** 23 variables

### Dataset Description

The independent variables describe the physical, hardware, and software characteristics of laptops and can be grouped as follows:

**1. Manufacturer & Product Information**

- Company – Laptop manufacturer
- Product – Brand and model name
- TypeName – Type of laptop (Notebook, Ultrabook, Gaming, etc.)

**2. Display Features**

- Inches – Screen size in inches
- Screen – Screen resolution category
- ScreenW – Screen width in pixels
- ScreenH – Screen height in pixels
- Touchscreen – Indicates whether the laptop has a touchscreen
- IPSpanel – Indicates presence of IPS panel
- RetinaDisplay – Indicates presence of retina display

**3. Hardware Specifications**

- Ram – Total RAM in GB
- CPU_company – CPU manufacturer
- CPU_model – CPU model
- CPU_freq – CPU frequency in GHz
- GPU_company – GPU manufacturer
- GPU_model – GPU model

**4. Storage Details**

- PrimaryStorage – Primary storage capacity in GB
- PrimaryStorageType – Type of primary storage (SSD, HDD, etc.)
- SecondaryStorage – Secondary storage capacity in GB
- SecondaryStorageType – Type of secondary storage

**5. Software & Physical Attributes**

- OS – Operating system
- Weight – Weight of the laptop in kilograms

## Data Quality

- The dataset contains **no missing values**, ensuring consistency and reliability for analysis.
- Data types include both numerical and categorical variables, requiring appropriate preprocessing before model building.

## Data source and Format

- **File Format:** CSV
- **Number of rows:** 1275
- **Number of columns:** 23

The CSV file was read using the read_csv() function in R and later transformed into Parquet format for faster loading and processing throughout the project.

# Chapter 3

## Tools and Technologies

This project was developed using a combination of data analysis tools, programming libraries, and visualization frameworks. These tools facilitated efficient data preprocessing, exploratory analysis, feature engineering, model building, and evaluation. The selection of tools was based on their robustness, ease of use, and suitability for machine learning workflows.

## R Programming Language

R was used as the primary programming language due to its powerful statistical capabilities and rich ecosystem of machine learning libraries. R provides extensive support for data manipulation, visualization, and model evaluation, making it a suitable choice for predictive analytics projects.

## RStudio

RStudio served as the integrated development environment (IDE) for writing and executing R code. Its user-friendly interface, built-in console, visual debugging features, and project-based workflow made it easier to manage the entire analysis process. RStudio also supports package management, visualization, and real-time code execution, which streamlined the development and testing phases.

### R Libraries Used

Several R packages and libraries were utilized to perform key tasks throughout the project:

- **tidyverse**: Used for efficient data manipulation, transformation, and exploratory analysis, including packages such as dplyr and ggplot2.
- **dplyr**: Enabled streamlined data wrangling operations such as filtering, selecting, and feature creation.
- **ggplot2**: Utilized for creating advanced and visually informative exploratory data analysis (EDA) visualizations.
- **caret**: Served as the core machine learning framework, facilitating model training, cross-validation, hyperparameter tuning, and performance comparison.
- **randomForest**: Implemented Random Forest regression and provided feature importance measures.
- **gbm**: Used to build Gradient Boosting regression models and optimize ensemble-based predictions.

- **Metrics**: Assisted in evaluating model performance using error-based metrics such as RMSE and MAE.
- **lubridate**: Supported any date-related feature handling and transformations.
- **corrplot**: Applied for correlation analysis and visualization among numerical features.

These libraries provided the foundation for building a smooth and efficient machine learning pipeline.

## Microsoft Excel

Excel was used optionally for quick data inspection, verifying data formats, and manually checking the structure of the dataset before importing it into R. It also served as a useful tool for reviewing generated CSV outputs such as feature importance tables and confusion matrices.

## Model Selection

Selecting appropriate machine learning models is a critical step in building a reliable predictive system. The choice of models in this study was guided by three key considerations: interpretability, ability to capture non-linear relationships, and predictive performance. To ensure a comprehensive evaluation, the following models were selected:

1. **Linear Regression**
    - Serves as a baseline model
    - Assumes a linear relationship between predictors and price
    - Easy to interpret
2. **Random Forest Regression**
    - Captures non-linear relationships
    - Robust to outliers and multicollinearity
    - Provides feature importance
3. **Gradient Boost Regression (Optional Advanced Model)**
    - Models non-linear relationships between laptop specifications and price more effectively than linear regression.
    - Models non-linear relationships between laptop specifications and price more effectively than linear regression.

# Chapter 4

## Data Preparation

- Install required packages and libraries.
- Load the data:

```
#load the data

laptop_data <- read.csv("C:/Users/abdul/Downloads/laptop_prices.csv")
```

- Data Preview:

```
#Data Preview

head(laptop_data)                #overview of first few rows
colSums(is.na(laptop_data))      #check any missing value
dim(laptop_data)                 #data dimention
str(laptop_data)                 #show each column and its contetnt
summary(laptop_data)             #dataset summary
sum(duplicated(laptop_data))     #check duplicate values
```

```
> head(laptop_data)                  #overview of first few rows
  Company      Product  TypeName Inches Ram         OS Weight Price_euros   Screen
1   Apple MacBook Pro Ultrabook   13.3   8      macOS   1.37     1339.69 Standard
2   Apple Macbook Air Ultrabook   13.3   8      macOS   1.34      898.94 Standard
3      HP       250 G6  Notebook   15.6   8      No OS   1.86      575.00  Full HD
4   Apple MacBook Pro Ultrabook   15.4  16      macOS   1.83     2537.45 Standard
5   Apple MacBook Pro Ultrabook   13.3   8      macOS   1.37     1803.60 Standard
6    Acer     Aspire 3  Notebook   15.6   4 Windows 10   2.10      400.00 Standard
  ScreenW ScreenH Touchscreen IPSpanel RetinaDisplay CPU_company CPU_freq
1    2560    1600          No      Yes           Yes       Intel      2.3
2    1440     900          No       No            No       Intel      1.8
3    1920    1080          No       No            No       Intel      2.5
4    2880    1800          No      Yes           Yes       Intel      2.7
5    2560    1600          No      Yes           Yes       Intel      3.1
6    1366     768          No       No            No         AMD      3.0
       CPU_model PrimaryStorage SecondaryStorage PrimaryStorageType
1        Core i5            128                0                SSD
2        Core i5            128                0      Flash Storage
3  Core i5 7200U            256                0                SSD
4        Core i7            512                0                SSD
5        Core i5            256                0                SSD
6 A9-Series 9420            500                0                HDD
  SecondaryStorageType GPU_company              GPU_model
1                   No       Intel Iris Plus Graphics 640
2                   No       Intel       HD Graphics 6000
3                   No       Intel        HD Graphics 620
4                   No         AMD         Radeon Pro 455
5                   No       Intel Iris Plus Graphics 650
6                   No         AMD              Radeon R5
```

```
> dim(laptop_data)                  #data dimention
[1] 1275   23
```

```
> sum(duplicated(laptop_data))      #check duplicate values
[1] 0
```

```
> summary(laptop_data)           #dataset summary
   Company            Product            TypeName             Inches     
 Length:1275        Length:1275        Length:1275        Min.   :10.10  
 Class :character   Class :character   Class :character   1st Qu.:14.00  
 Mode  :character   Mode  :character   Mode  :character   Median :15.60  
                                                          Mean   :15.02  
                                                          3rd Qu.:15.60  
                                                          Max.   :18.40  
      Ram              OS                Weight        Price_euros  
 Min.   : 2.000   Length:1275        Min.   :0.690   Min.   : 174  
 1st Qu.: 4.000   Class :character   1st Qu.:1.500   1st Qu.: 609  
 Median : 8.000   Mode  :character   Median :2.040   Median : 989  
 Mean   : 8.441                      Mean   :2.041   Mean   :1135  
 3rd Qu.: 8.000                      3rd Qu.:2.310   3rd Qu.:1496  
 Max.   :64.000                      Max.   :4.700   Max.   :6099  
    Screen             ScreenW        ScreenH       Touchscreen       
 Length:1275        Min.   :1366   Min.   : 768   Length:1275        
 Class :character   1st Qu.:1920   1st Qu.:1080   Class :character   
 Mode  :character   Median :1920   Median :1080   Mode  :character   
                    Mean   :1900   Mean   :1074                      
                    3rd Qu.:1920   3rd Qu.:1080                      
                    Max.   :3840   Max.   :2160                      
   IPSpanel         RetinaDisplay      CPU_company           CPU_freq    
 Length:1275        Length:1275        Length:1275        Min.   :0.900  
 Class :character   Class :character   Class :character   1st Qu.:2.000  
 Mode  :character   Mode  :character   Mode  :character   Median :2.500  
                                                          Mean   :2.303  
                                                          3rd Qu.:2.700  
                                                          Max.   :3.600  
  CPU_model         PrimaryStorage   SecondaryStorage PrimaryStorageType
 Length:1275        Min.   :   8.0   Min.   :   0.0   Length:1275       
 Class :character   1st Qu.: 256.0   1st Qu.:   0.0   Class :character  
 Mode  :character   Median : 256.0   Median :   0.0   Mode  :character  
                    Mean   : 444.5   Mean   : 176.1                     
                    3rd Qu.: 512.0   3rd Qu.:   0.0                     
                    Max.   :2048.0   Max.   :2048.0                     
 SecondaryStorageType GPU_company         GPU_model        
 Length:1275          Length:1275        Length:1275       
 Class :character     Class :character   Class :character  
 Mode  :character     Mode  :character   Mode  :character  
```

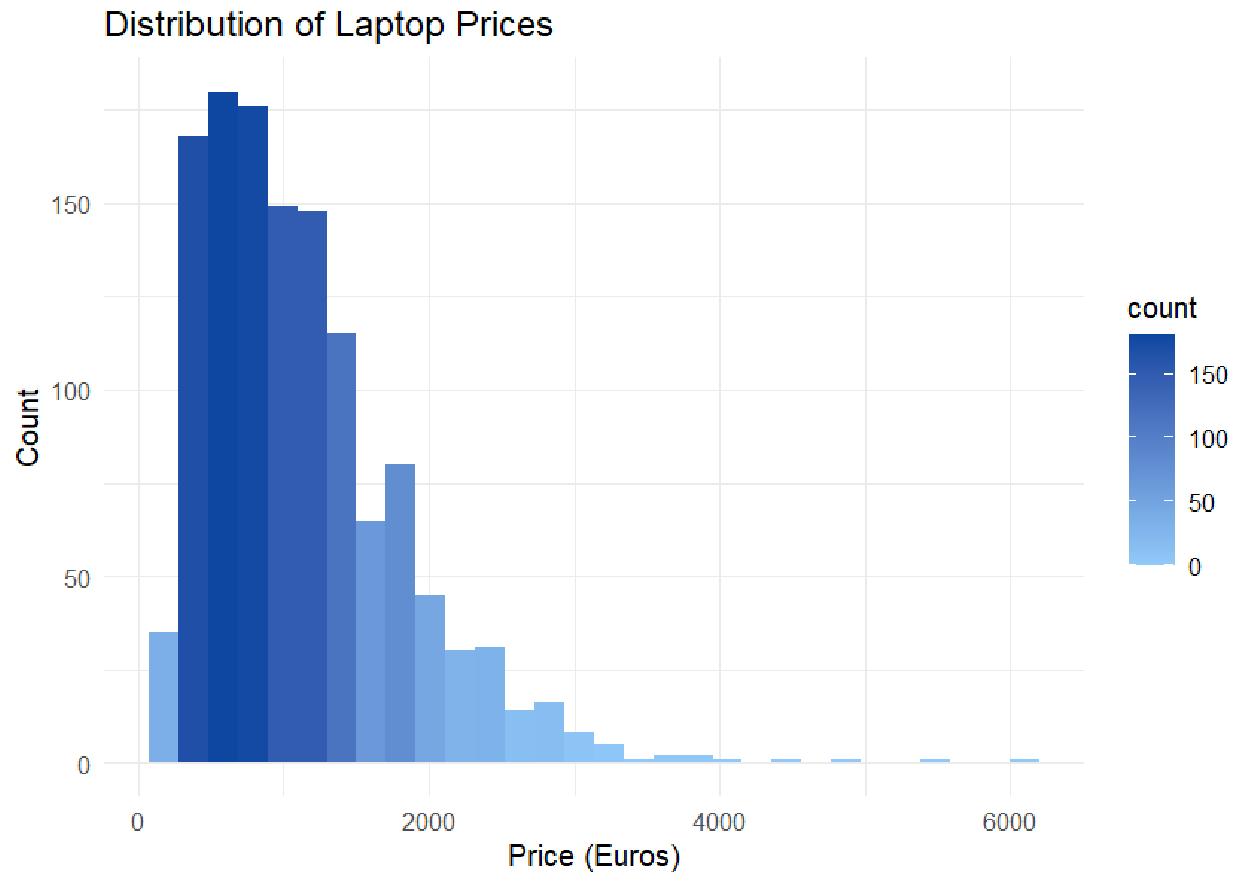
```
> str(laptop_data)                   #show each column and its contetnt
'data.frame':	1275 obs. of  23 variables:
 $ Company             : chr  "Apple" "Apple" "HP" "Apple" ...
 $ Product             : chr  "MacBook Pro" "Macbook Air" "250 G6" "MacBook Pro" ...
 $ TypeName            : chr  "Ultrabook" "Ultrabook" "Notebook" "Ultrabook" ...
 $ Inches              : num  13.3 13.3 15.6 15.4 13.3 15.6 15.4 13.3 14 14 ...
 $ Ram                 : int  8 8 8 16 8 4 16 8 16 8 ...
 $ OS                  : chr  "macOS" "macOS" "No OS" "macOS" ...
 $ Weight              : num  1.37 1.34 1.86 1.83 1.37 2.1 2.04 1.34 1.3 1.6 ...
 $ Price_euros         : num  1340 899 575 2537 1804 ...
 $ Screen              : chr  "Standard" "Standard" "Full HD" "Standard" ...
 $ ScreenW             : int  2560 1440 1920 2880 2560 1366 2880 1440 1920 1920 ...
 $ ScreenH             : int  1600 900 1080 1800 1600 768 1800 900 1080 1080 ...
 $ Touchscreen         : chr  "No" "No" "No" "No" ...
 $ IPSpanel            : chr  "Yes" "No" "No" "Yes" ...
 $ RetinaDisplay       : chr  "Yes" "No" "No" "Yes" ...
 $ CPU_company         : chr  "Intel" "Intel" "Intel" "Intel" ...
 $ CPU_freq            : num  2.3 1.8 2.5 2.7 3.1 3 2.2 1.8 1.8 1.6 ...
 $ CPU_model           : chr  "Core i5" "Core i5" "Core i5 7200U" "Core i7" ...
 $ PrimaryStorage      : int  128 128 256 512 256 500 256 256 512 256 ...
 $ SecondaryStorage    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ PrimaryStorageType  : chr  "SSD" "Flash Storage" "SSD" "SSD" ...
 $ SecondaryStorageType: chr  "No" "No" "No" "No" ...
 $ GPU_company         : chr  "Intel" "Intel" "Intel" "AMD" ...
 $ GPU_model           : chr  "Iris Plus Graphics 640" "HD Graphics 6000" "HD Graphics 620" "Radeon Pro 455" ...
```

```
> colSums(is.na(laptop_data))       #check any missing value
             Company              Product             TypeName 
                   0                    0                    0 
              Inches                  Ram                   OS 
                   0                    0                    0 
              Weight          Price_euros               Screen 
                   0                    0                    0 
             ScreenW              ScreenH          Touchscreen 
                   0                    0                    0 
            IPSpanel        RetinaDisplay          CPU_company 
                   0                    0                    0 
            CPU_freq            CPU_model       PrimaryStorage 
                   0                    0                    0 
    SecondaryStorage   PrimaryStorageType SecondaryStorageType 
                   0                    0                    0 
         GPU_company            GPU_model 
                   0                    0 
```

## Exploratory Data Analysis

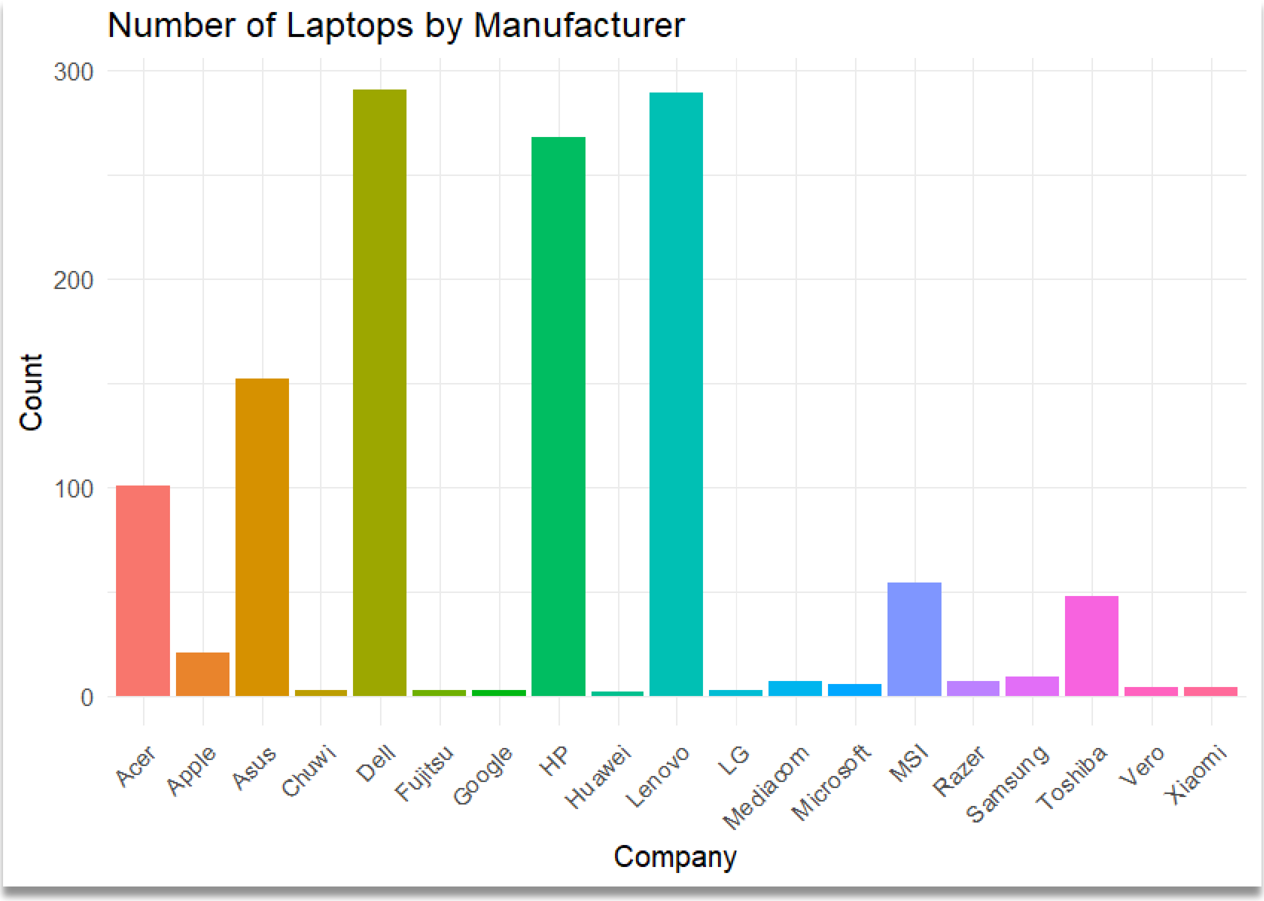
- Laptop Price Distribution:

```
#EDA

#Laptop price distribution
ggplot(laptop_data, aes(x = Price_euros)) +
  geom_histogram(aes(fill = ..count..), bins = 30) +
  scale_fill_gradient(low = "#90CAF9", high = "#0D47A1") +
  labs(
    title = "Distribution of Laptop Prices",
    x = "Price (Euros)",
    y = "Count"
  ) +
  theme_minimal()
```



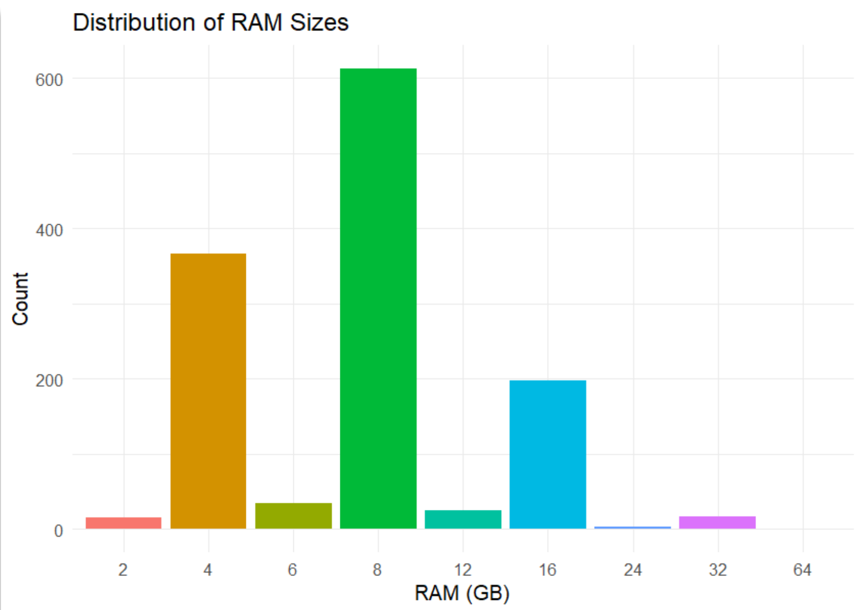

- Laptop Company distribution:

```
#Laptops by company
ggplot(laptop_data, aes(x = Company, fill = Company)) +
  geom_bar(show.legend = FALSE) +
  labs(
    title = "Number of Laptops by Manufacturer",
    x = "Company",
    y = "Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



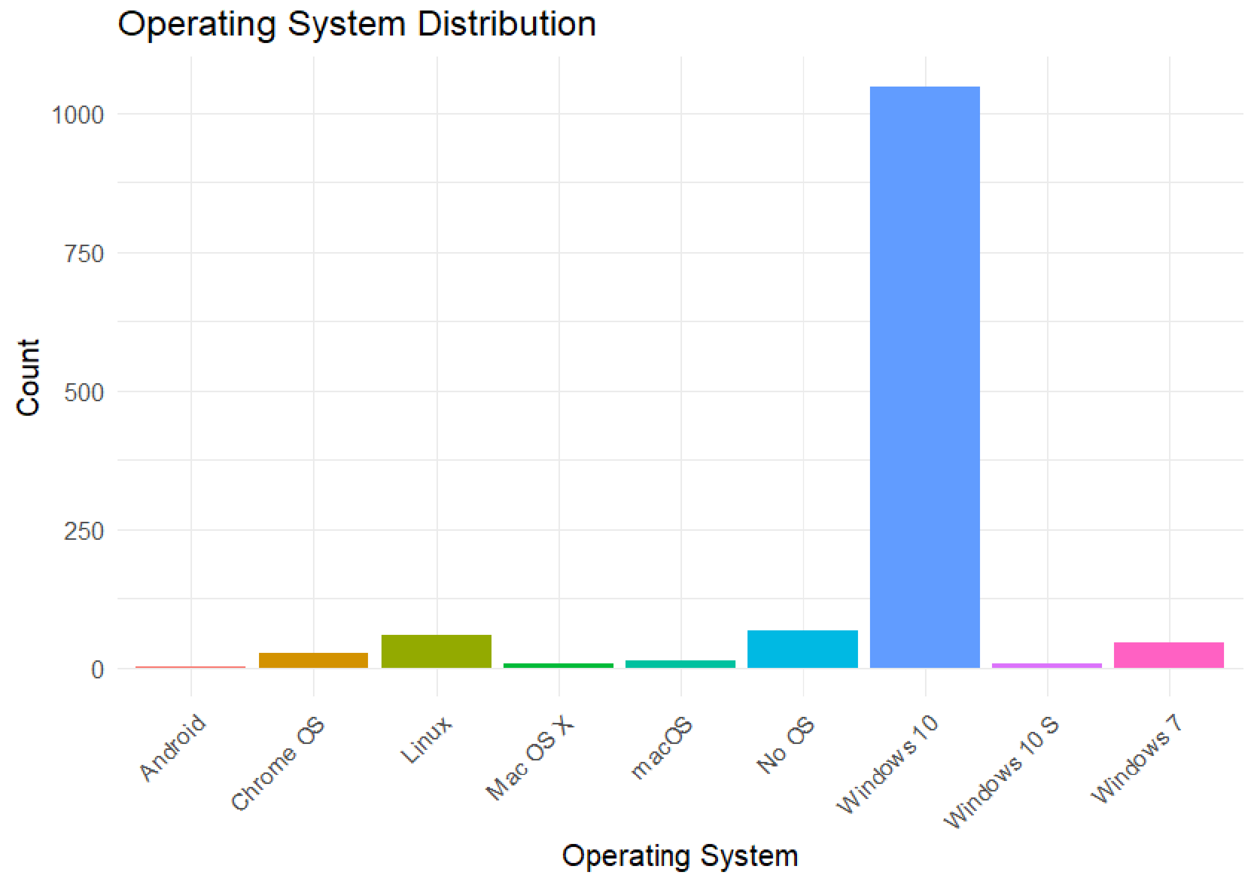

➢ RAM distribution in laptops:

```
#Laptop RAM distribution
ggplot(laptop_data, aes(x = factor(Ram), fill = factor(Ram))) +
  geom_bar(show.legend = FALSE) +
  labs(
    title = "Distribution of RAM Sizes",
    x = "RAM (GB)",
    y = "Count"
  ) +
  theme_minimal()
```



➢ Operating System Distribution:

```
#OS Distribution
ggplot(laptop_data, aes(x = OS, fill = OS)) +
  geom_bar(show.legend = FALSE) +
  labs(
    title = "Operating System Distribution",
    x = "Operating System",
    y = "Count"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

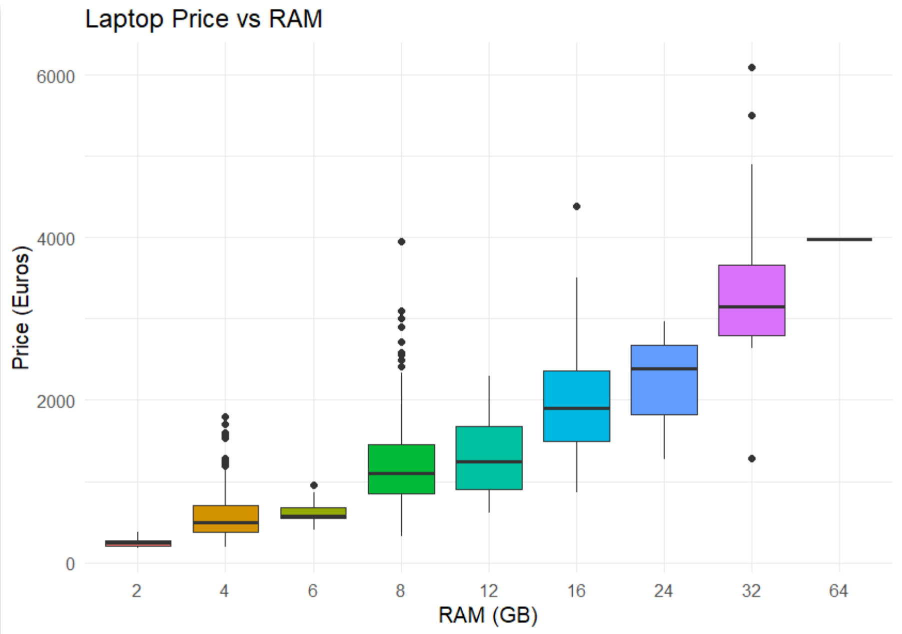


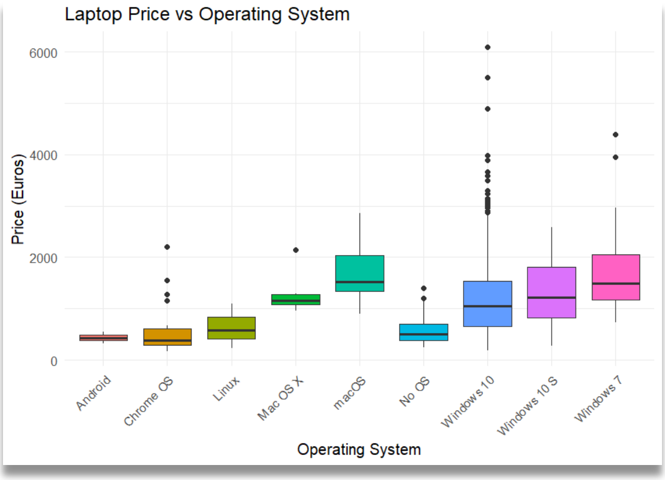

➢ Laptop prices against RAM, OS, Touchscreen, Primary Storage type and CPU manufacturer:

```
#Price vs Ram
ggplot(laptop_data, aes(x = factor(Ram), y = Price_euros, fill = factor(Ram))) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Laptop Price vs RAM",
    x = "RAM (GB)",
    y = "Price (Euros)"
  ) +
  theme_minimal()
```

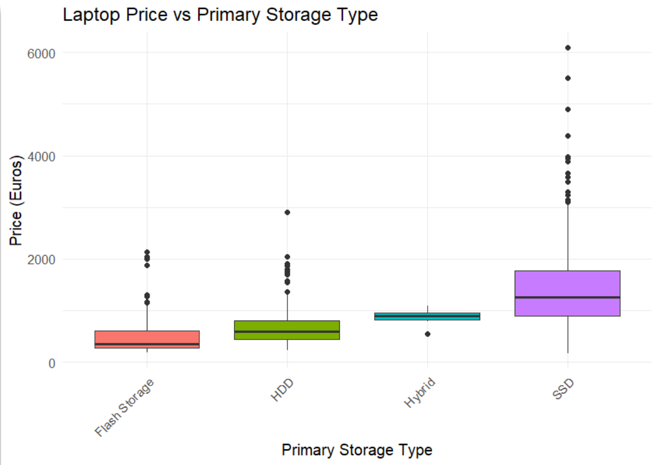
➢ L

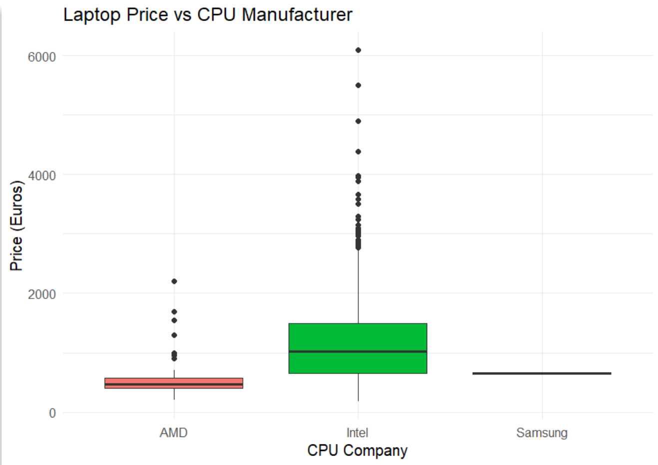
```
#Price vs Ram
ggplot(laptop_data, aes(x = factor(Ram), y = Price_euros, fill = factor(Ram))) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Laptop Price vs RAM",
    x = "RAM (GB)",
    y = "Price (Euros)"
  ) +
  theme_minimal()

#Price vs OS
ggplot(laptop_data, aes(x = OS, y = Price_euros, fill = OS)) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Laptop Price vs Operating System",
    x = "Operating System",
    y = "Price (Euros)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Price vs Touchscreen
ggplot(laptop_data, aes(x = factor(Touchscreen), y = Price_euros, fill = factor(Touchscreen))) +
  geom_boxplot(show.legend = FALSE) +
  scale_fill_manual(values = c("#FFCC80", "#FF7043")) +
  labs(
    title = "Laptop Price vs Touchscreen Availability",
    x = "Touchscreen (0 = No, 1 = Yes)",
    y = "Price (Euros)"
  ) +
  theme_minimal()

#Price vs Primary storage type
ggplot(laptop_data, aes(x = PrimaryStorageType, y = Price_euros, fill = PrimaryStorageType)) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Laptop Price vs Primary Storage Type",
    x = "Primary Storage Type",
    y = "Price (Euros)"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Price vs CPU company
ggplot(laptop_data, aes(x = CPU_company, y = Price_euros, fill = CPU_company)) +
  geom_boxplot(show.legend = FALSE) +
  labs(
    title = "Laptop Price vs CPU Manufacturer",
    x = "CPU Company",
    y = "Price (Euros)"
  ) +
  theme_minimal()
```





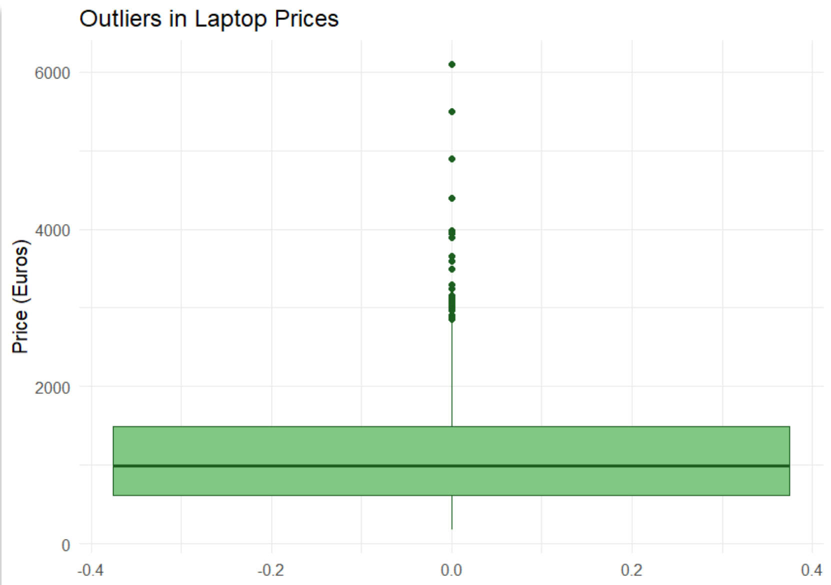





- **Outlier Visualization:**

```r
#Outlier Visualization
ggplot(laptop_data, aes(y = Price_euros)) +
  geom_boxplot(fill = "#81C784", color = "#1B5E20") +
  labs(
    title = "Outliers in Laptop Prices",
    y = "Price (Euros)"
  ) +
  theme_minimal()
```



## Feature Engineering

- Required conversions and mutations:

```
#Feature Engineering
#do required conversions and mutations
laptop_data <- laptop_data %>%
  mutate(ScreenResolution = ScreenW * ScreenH)
laptop_data <- laptop_data %>%
  mutate(TotalStorage = PrimaryStorage + SecondaryStorage)
laptop_data <- laptop_data %>%
  mutate(
    Touchscreen   = ifelse(Touchscreen == 1, 1, 0),
    IPSpanel      = ifelse(IPSpanel == 1, 1, 0),
    RetinaDisplay = ifelse(RetinaDisplay == 1, 1, 0)
  )
factor_cols <- c(
  "Company", "TypeName", "OS", "Screen",
  "CPU_company", "CPU_model",
  "PrimaryStorageType", "SecondaryStorageType",
  "GPU_company", "GPU_model"
)

laptop_data[factor_cols] <- lapply(
  laptop_data[factor_cols],
  as.factor
)
```

- Remove redundant colomns and create test-train split:

```
#remove redundant columns
laptop_data <- laptop_data %>%
  select(-ScreenW, -ScreenH)

#create test-train split
set.seed(123)

train_index <- createDataPartition(
  laptop_data$Price_euros,
  p = 0.8,
  list = FALSE
)

train_data <- laptop_data[train_index, ]
test_data  <- laptop_data[-train_index, ]
```

- Separate features and target variables:

```
#separate features and target variables
x_train <- train_data %>% select(-Price_euros)
y_train <- train_data$Price_euros

x_test  <- test_data %>% select(-Price_euros)
y_test  <- test_data$Price_euros
```

➢ Scale Variables:

```
#scaling numerical variables
numeric_cols <- c(
  "Inches", "Ram", "Weight", "CPU_freq",
  "PrimaryStorage", "SecondaryStorage",
  "ScreenResolution", "TotalStorage"
)

valid_numeric_cols <- numeric_cols[
  numeric_cols %in% colnames(x_train) &
    sapply(x_train[numeric_cols], is.numeric)
]
train_scaled <- scale(x_train[valid_numeric_cols])

train_center <- attr(train_scaled, "scaled:center")
train_scale  <- attr(train_scaled, "scaled:scale")
```

➢ Apply TRAIN parameters to train and test data:

```
# Apply to train
x_train[valid_numeric_cols] <- train_scaled

# Apply to test using TRAIN parameters
x_test[valid_numeric_cols] <- scale(
  x_test[valid_numeric_cols],
  center = train_center,
  scale  = train_scale
)
preproc <- preProcess(
  x_train[valid_numeric_cols],
  method = c("center", "scale")
)

x_train[valid_numeric_cols] <- predict(preproc, x_train[valid_numeric_cols])
x_test[valid_numeric_cols]  <- predict(preproc, x_test[valid_numeric_cols])
```

➢ Remoeve columns with high cardinality:

```
#remoeve columns with high cardinality
high_cardinality <- c("Product", "CPU_model", "GPU_model")

x_train <- x_train %>% select(-all_of(high_cardinality))
x_test  <- x_test %>% select(-all_of(high_cardinality))
```

## Model Building and Evaluation

- Setup train control:

```
#Model building and evaluation

#train-control setup
set.seed(123)

train_control <- trainControl(
  method = "cv",
  number = 5
)
```

- Linear Regression Model (Baseline):

```
#LRM Baseline
set.seed(123)

lm_model <- train(
  x = x_train,
  y = y_train,
  method = "lm",
  trControl = train_control
)

lm_model
#predictions and metrics
lm_pred <- predict(lm_model, x_test)

lm_rmse <- rmse(y_test, lm_pred)
lm_mae  <- mae(y_test, lm_pred)
lm_r2   <- R2(lm_pred, y_test)
```

```
> lm_model
Linear Regression

1021 samples
  19 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 817, 818, 817, 816, 816
Resampling results:

  RMSE      Rsquared   MAE
  336.0698  0.7817376  238.6685

Tuning parameter 'intercept' was held constant at a value of TRUE
>
```

- Random Forest Model:

```r
#RF Model
set.seed(123)

rf_model <- train(
  x = x_train,
  y = y_train,
  method = "rf",
  trControl = train_control,
  tuneLength = 5,
  importance = TRUE
)

rf_model
#predictions and metrics
rf_pred <- predict(rf_model, x_test)

rf_rmse <- rmse(y_test, rf_pred)
rf_mae  <- mae(y_test, rf_pred)
rf_r2   <- R2(rf_pred, y_test)
```

```
> rf_model
Random Forest

1021 samples
  19 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 817, 818, 817, 816, 816
Resampling results across tuning parameters:

  mtry  RMSE      Rsquared   MAE
   2    335.9931  0.8172777  229.3034
   6    292.8555  0.8386572  191.0449
  10    291.2948  0.8380317  191.3895
  14    291.7209  0.8357403  191.6010
  19    298.9617  0.8268759  193.8213

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 10.
```

- Gradient Boost Model:

```r
#Gradient boost model
set.seed(123)

gbm_model <- train(
  x = x_train,
  y = y_train,
  method = "gbm",
  trControl = train_control,
  verbose = FALSE
)
gbm_model
#predictions and metrics
gbm_pred <- predict(gbm_model, x_test)

gbm_rmse <- rmse(y_test, gbm_pred)
gbm_mae  <- mae(y_test, gbm_pred)
gbm_r2   <- R2(gbm_pred, y_test)
```

```
> gbm_model
Stochastic Gradient Boosting

1021 samples
  19 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 817, 818, 817, 816, 816
Resampling results across tuning parameters:

  interaction.depth  n.trees  RMSE      Rsquared   MAE
  1                   50      384.2666  0.7334878  272.7607
  1                  100      341.9173  0.7732958  234.5283
  1                  150      329.0706  0.7879291  225.5486
  2                   50      343.9715  0.7767222  237.6542
  2                  100      317.5159  0.8026206  218.9629
  2                  150      309.5170  0.8116724  214.4874
  3                   50      338.1260  0.7790422  229.6965
  3                  100      311.3464  0.8104644  215.7285
  3                  150      304.0508  0.8185939  210.8059

Tuning parameter 'shrinkage' was held constant at a value of 0.1
Tuning parameter 'n.minobsinnode' was
 held constant at a value of 10
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were n.trees = 150, interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode
 = 10.
```

## Model Comparison and Variable Importance

- Model results:

```
#Model Performance Comparision
model_results <- data.frame(
  Model = c("Linear Regression", "Random Forest", "Gradient Boosting"),
  RMSE  = c(lm_rmse, rf_rmse, gbm_rmse),
  MAE   = c(lm_mae, rf_mae, gbm_mae),
  R2    = c(lm_r2, rf_r2, gbm_r2)
)

model_results
```

```
> model_results
              Model     RMSE      MAE        R2
1 Linear Regression 316.6925 239.7882 0.7707278
2     Random Forest 251.1217 166.5138 0.8612272
3 Gradient Boosting 276.4293 191.2987 0.8255166
> 
```

- rmse comparision plot:

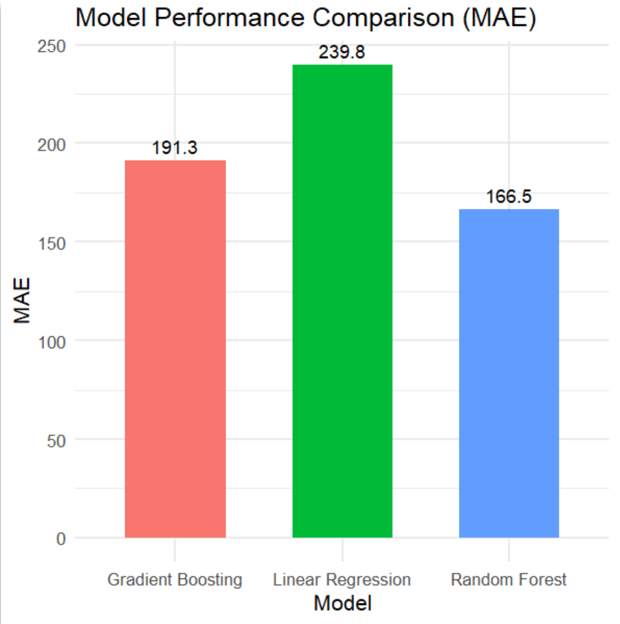
```
#rmse comparison plot
ggplot(model_results, aes(x = Model, y = RMSE, fill = Model)) +
  geom_bar(stat = "identity", width = 0.6) +
  geom_text(aes(label = round(RMSE, 1)), vjust = -0.5, size = 4) +
  labs(
    title = "Model Performance Comparison (RMSE)",
    x = "Model",
    y = "RMSE"
  ) +
  theme_minimal(base_size = 13) +
  theme(legend.position = "none")
```

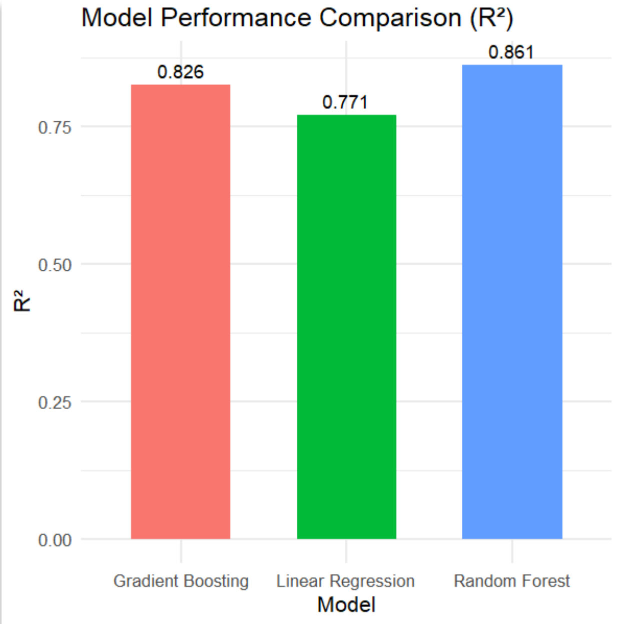
- MAE comparision plot:

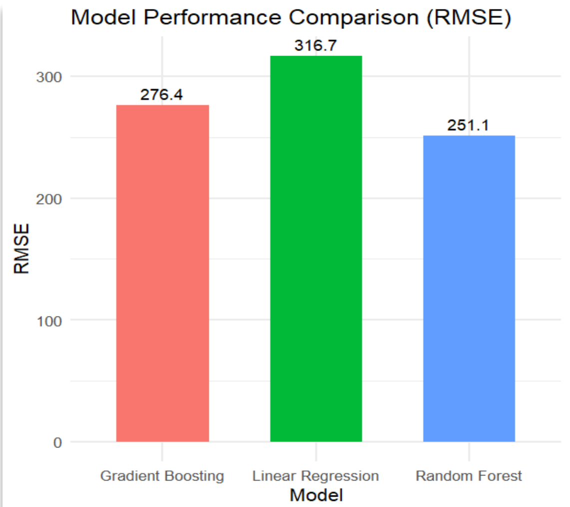
```
#MAE comparison PLot
ggplot(model_results, aes(x = Model, y = MAE, fill = Model)) +
  geom_bar(stat = "identity", width = 0.6) +
  geom_text(aes(label = round(MAE, 1)), vjust = -0.5, size = 4) +
  labs(
    title = "Model Performance Comparison (MAE)",
    x = "Model",
    y = "MAE"
  ) +
  theme_minimal(base_size = 13) +
  theme(legend.position = "none")
```

- R square comparision plot:

```
#R square comparioson plot
ggplot(model_results, aes(x = Model, y = R2, fill = Model)) +
  geom_bar(stat = "identity", width = 0.6) +
  geom_text(aes(label = round(R2, 3)), vjust = -0.5, size = 4) +
  labs(
    title = "Model Performance Comparison (R²)",
    x = "Model",
    y = "R²"
  ) +
  theme_minimal(base_size = 13) +
  theme(legend.position = "none")
```



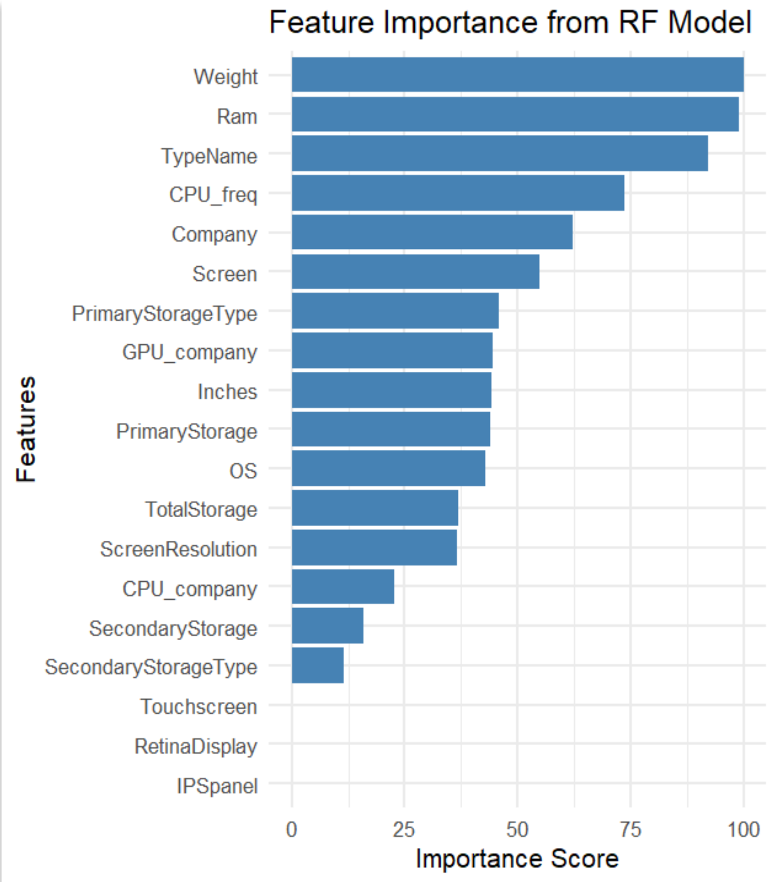



- Feature (Variable) Importance:

```r
#Feature Impoortance

rf_importance <- varImp(rf_model, scale = TRUE)

rf_importance

rf_imp_df <- rf_importance$importance
rf_imp_df$Feature <- rownames(rf_imp_df)

ggplot(rf_imp_df, aes(x = reorder(Feature, Overall), y = Overall)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Feature Importance from Random Forest Model",
    x = "Features",
    y = "Importance Score"
  ) +
  theme_minimal(base_size = 13)
```

```
> rf_importance
rf variable importance

                     Overall
Weight                100.00
Ram                    99.03
TypeName               92.24
CPU_freq               73.56
Company                62.23
Screen                 54.73
PrimaryStorageType     45.81
GPU_company            44.36
Inches                 44.25
PrimaryStorage         43.91
OS                     42.98
TotalStorage           36.91
ScreenResolution       36.72
CPU_company            22.58
SecondaryStorage       15.94
SecondaryStorageType   11.55
RetinaDisplay           0.00
Touchscreen             0.00
IPSpanel                0.00
```



# Chapter 5

## Results

This study evaluated multiple machine learning models to predict laptop prices using hardware specifications and system features. A structured modelling pipeline was implemented, and models were assessed using 5-fold cross-validation to ensure reliable performance estimation.

### Model Performance and Comparison

Linear Regression served as the baseline model, achieving an RMSE of 336.07, MAE of 238.67, and an $R^2$ value of 0.78. While the model explained a substantial portion of price variability, its performance was constrained by the assumption of linear relationships among predictors.

The Gradient Boosting model demonstrated a notable improvement over Linear Regression. With optimized hyperparameters, it achieved an RMSE of 304.05, MAE of 210.81, and an $R^2$ of 0.82. These results indicate Gradient Boosting's ability to capture non-linear patterns and interactions among laptop specifications more effectively than linear methods.

The Random Forest model produced the strongest performance among all evaluated models. It achieved the lowest RMSE (291.29), lowest MAE (191.39), and highest $R^2$ (0.84), reflecting superior predictive accuracy and robustness across validation folds.

### Final Model Selection

Based on comparative performance and cross-validation results, Random Forest Regression was selected as the final model. Its ensemble structure effectively captured interactions among hardware features while maintaining robustness to noise and multicollinearity.

### Feature Importance Results

Feature importance analysis of the Random Forest model revealed that laptop prices are primarily driven by performance-oriented specifications. RAM capacity and CPU frequency emerged as the most influential predictors, followed by total storage capacity, storage type, and screen resolution. GPU manufacturer and physical attributes such as weight also contributed meaningfully, while brand and model-level identifiers had limited generalizable impact.

### Additional Key Findings

- Laptop pricing exhibits strong non-linear dependencies, justifying the use of ensemble models.
- SSD-based storage and higher-resolution displays consistently correspond to higher price ranges.
- Manufacturer-level features provide predictive value without overfitting, unlike model-specific identifiers.

- Model validation showed minimal bias and strong alignment between actual and predicted prices, confirming good generalization

## Project Summary

The objective of this project was to build a predictive model capable of accurately estimating laptop prices based on technical specifications. After comprehensive exploratory data analysis and feature engineering, several regression models were trained and compared. Linear Regression served as a baseline model, while ensemble techniques such as Gradient Boosting and Random Forest were employed to capture non-linear relationships. Random Forest emerged as the most effective model, achieving an $R^2$ score of approximately 0.84 and significantly lower error metrics compared to the baseline. The findings provide actionable insights for manufacturers, retailers, and consumers by identifying key features that drive laptop pricing. The project demonstrates a structured end-to-end machine learning workflow applicable to real-world business problems.

## Conclusion

This project successfully developed a machine learning–based framework to analyse and predict laptop prices using a wide range of hardware specifications and system features. Through systematic data preprocessing, exploratory data analysis, feature engineering, and model development, meaningful patterns influencing laptop pricing were identified.

Multiple regression models were implemented and evaluated using cross-validation, including Linear Regression, Gradient Boosting Regression, and Random Forest Regression. Among these, the Random Forest model demonstrated superior performance, achieving the lowest prediction error and the highest explained variance. This confirms that laptop pricing is influenced by complex, non-linear interactions among features, which are effectively captured by ensemble-based models.

Overall, the study demonstrates that machine learning techniques can provide accurate and interpretable solutions for price prediction in the consumer electronics domain.

## Recommendations

Based on the analysis, the following recommendations can be made:

**For Manufacturers**

- Focus on optimizing RAM, processor speed, and SSD storage to position products in higher price segments.
- Lightweight designs and high-resolution displays can enhance perceived product value.

**For Retailers**

- Use machine learning-based pricing insights to segment products into budget, mid-range, and premium categories.
- Highlight key hardware specifications that drive pricing in marketing strategies.

**For Consumers**

- Prioritize investments in RAM, CPU performance, and SSD storage for better long-term value.
- Brand and model names should be evaluated alongside core technical specifications rather than in isolation.

## Future Scope

The scope of this project can be extended in several ways:

- Incorporating time-series data to study price trends and depreciation.
- Including consumer ratings and reviews to capture sentiment-based pricing effects.
- Applying advanced models such as XGBoost or deep learning architectures for further performance improvement.
- Deploying the model as a web-based pricing recommendation system for real-time use.
- Expanding the dataset to include newer laptop models and emerging hardware technologies.