CONTENTS

CHAPTER 1

INTRODUCTION

Arthur Samuel, a pioneer in the field of artificial intelligence and computer gaming, coined the term “Machine Learning”. He defined machine learning as – a “Field of study that gives computers thecapability to learn without being explicitly programmed”. The process starts with feeding good qualitydata and then training our machines(computers) by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data do we have and whatkind of task we are trying to automate.

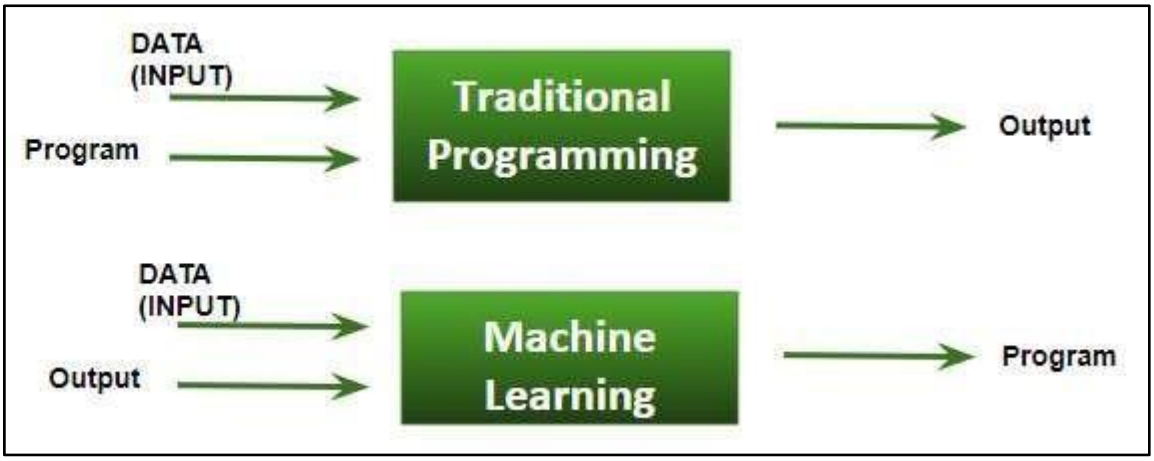


Figure 1.1: Difference between the traditional and machine learning

How does ML work?

- Gathering past data in any form suitable for processing. The better the quality of data, the moresuitable it will be for modeling
- Data Processing – Sometimes, the data collected is in raw form and it needs to be pre-processed. Example: Some tuples may have missing values for certain attributes, and, in this case, it has tobe filled with suitable values in order to perform machine learning or any form of data mining. Missing values for numerical attributes such as the price of the house may be replaced with the mean value of the attribute whereas missing values for categorical attributes

may be replace with the attribute with the highest mode. This invariably depends on the types of filters we use.If data is in the form of text or images then converting it to numerical form will be required, be it a list or array or matrix.

- Divide the input data into training, cross-validation, and test sets. The ratio between therespective sets must be 6:2:2
- Building models with suitable algorithms and techniques on the training set.
- Testing our conceptualized model with data that was not fed to the model at the time oftraining and evaluating its performance using metrics such as F1 score, precision, and recall.
    - Linear Algebra
    - Statistics and Probability
    - Calculus
    - Graph theory
    - Programming Skills – Languages such as Python, R, MATLAB, C++, or Octave.

Types of Machine Learning

Based on the methods and way of learning, machine learning is divided into mainly four types, which are:

1. Supervised Machine Learning
2. Unsupervised Machine Learning
3. Semi-Supervised Machine Learning
4. Reinforcement Learning

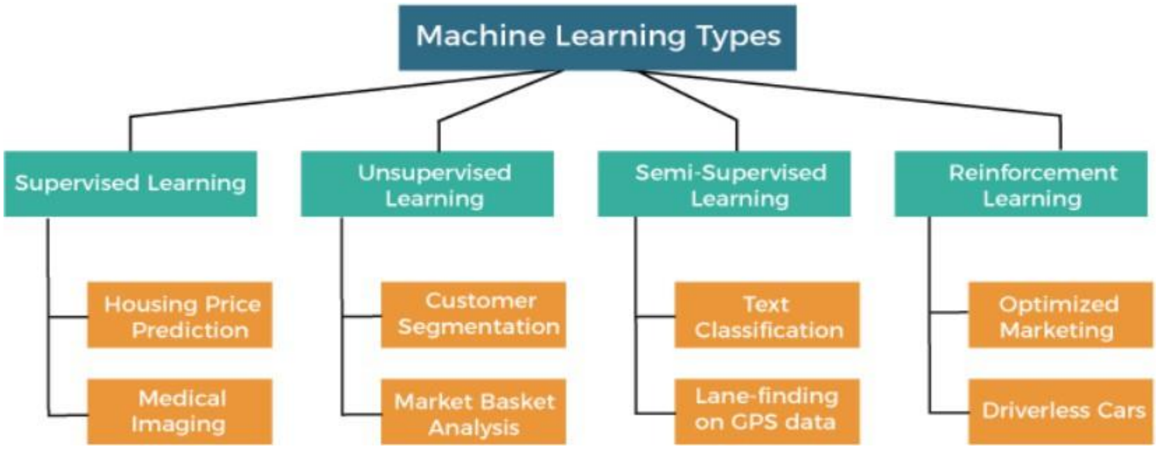


Figure 1.2: Types of Machine Learning

Supervised Machine Learning

As its name suggests, Supervised machine learning is based on supervision. It means in the supervised learning technique, we train the machines using the "labelled" dataset, and based on the training, the machine predicts the output. Here, the labeled data specifies that some of the inputs are already mapped to the output. More preciously, we can say; first, we train the machine with the input and corresponding output, and then we ask the machine to predict the output using the test dataset.

The main goal of the supervised learning technique is to map the input variable(x) with the output variable(y). Some real-world applications of supervised learning are Risk Assessment, Fraud Detection, Spam filtering, **etc.**

Categories of Supervised Machine Learning

Supervised machine learning can be classified into two types of problems, which are given below:

- Classification
- Regression
-

a) Classification

Classification algorithms are used to solve the classification problems in which the output variable is categorical, such as **"Yes" or No,** Male or Female, Red or Blue, etc. The classification algorithms predict the categories present in the dataset. Some real-world examples of classification algorithms are Spam Detection, Email filtering, etc.

Some popular classification algorithms are given below:

- Random Forest Algorithm
- Decision Tree Algorithm
- Logistic Regression Algorithm
- Support Vector Machine Algorithm

b) Regression

Regression algorithms are used to solve regression problems in which there is a linear relationship between input and output variables. These are used to predict continuous output variables, such as market trends, weather prediction, etc.

Some popular Regression algorithms are given below:

- Simple Linear Regression Algorithm
- Multivariate Regression Algorithm
- Decision Tree Algorithm
- Lasso Regression

Advantages and Disadvantages of Supervised Learning

Advantages:

- Since supervised learning work with the labelled dataset so we can have an exact idea about the classes of objects.
- These algorithms are helpful in predicting the output on the basis of prior experience.

Disadvantages:

- These algorithms are not able to solve complex tasks.
- It may predict the wrong output if the test data is different from the training data.
- It requires lots of computational time to train the algorithm.

Unsupervised Machine Learning

Unsupervised learning is different from the Supervised learning technique; as its name suggests, there is no need for supervision. It means, in unsupervised machine learning, the machine is trained using the unlabeled dataset, and the machine predicts the output without any supervision.

In unsupervised learning, the models are trained with the data that is neither classified nor labeled, and the model acts on that data without any supervision.

The main aim of the unsupervised learning algorithm is to group or categories the unsorted dataset according to the similarities, patterns, and differences**.** Machines are instructed to find the hidden patterns from the input dataset.

Categories of Unsupervised Machine Learning

Unsupervised Learning can be further classified into two types, which are given below:

- Clustering
- Association

1) Clustering

The clustering technique is used when we want to find the inherent groups from the data. It is a way to group the objects into a cluster such that the objects with the most similarities remain in one group and have fewer or no similarities with the objects of other groups. An example of the clustering algorithm is grouping the customers by their purchasing behavior.

Some of the popular clustering algorithms are given below:

- K-Means Clustering algorithm
- Mean-shift algorithm
- DBSCAN Algorithm
- Principal Component Analysis
- Independent Component Analysis

2) Association

Association rule learning is an unsupervised learning technique, which finds interesting relations among variables within a large dataset. The main aim of this learning algorithm is to find the dependency of one data item on another data item and map those variables accordingly so that it can generate maximum profit. This algorithm is mainly applied in Market Basket analysis, Web usage mining, continuous production**,** etc.

Some popular algorithms of Association rule learning are Apriori Algorithm, Eclat, FP-growth algorithm.

Advantages and Disadvantages of Unsupervised Learning Algorithm

Advantages:

- These algorithms can be used for complicated tasks compared to the supervised ones because these algorithms work on the unlabeled dataset.
- Unsupervised algorithms are preferable for various tasks as getting the unlabeled dataset is easier as compared to the labelled dataset.

Disadvantages:

- The output of an unsupervised algorithm can be less accurate as the dataset is not labelled, and algorithms are not trained with the exact output in prior.
- Working with Unsupervised learning is more difficult as it works with the unlabelled dataset that does not map with the output.

Semi-Supervised Learning

Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate ground between Supervised (With Labeled training data) and Unsupervised learning (with no labeled training data) algorithms and uses the combination of labeled and unlabeled datasets during the training period.

Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for corporate purposes, they may have few labels. It is completely different from supervised and unsupervised learning as they are based on the presence & absence of labels.

To overcome the drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. The main aim of semi-supervised learning is to effectively use all the available data, rather than only labeled data like in supervised learning. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labelled data. It is because labeled data is a comparatively more expensive acquisition than unlabeled data.

Advantages and disadvantages of Semi-supervised Learning

Advantages:

- It is simple and easy to understand the algorithm.
- It is highly efficient.
- It is used to solve drawbacks of Supervised and Unsupervised Learning algorithms.

Disadvantages:

- Iterations results may not be stable.
- We cannot apply these algorithms to network-level data.
- Accuracy is low.

Reinforcement Learning

Reinforcement learning works on a feedback-based process, in which an AI agent (A software component) automatically explore its surrounding by hitting & trail, taking action, learning from experiences, and improving its performance**.** Agent gets rewarded for each good action and get punished for each bad action; hence the goal of reinforcement learning agent is to maximize the rewards.

In reinforcement learning, there is no labeled data like supervised learning, and agents learn from their experiences only.

The reinforcement learning process is similar to a human being; for example, a child learns various things by experiences in his day-to-day life. An example of reinforcement learning is to play a game, where the Game is the environment, moves of an agent at each step define states, and the goal of the agent is to get a high score. Agent receives feedback in terms of punishment and rewards.

Due to its way of working, reinforcement learning is employed in different fields such as Game theory, Operation Research, Information theory, multi-agent systems**.**

A reinforcement learning problem can be formalized using Markov Decision Process(MDP)**.** In MDP, the agent constantly interacts with the environment and performs actions; at each action, the environment responds and generates a new state.

Categories of Reinforcement Learning

Reinforcement learning is categorized mainly into two types of methods/algorithms:

- **Positive Reinforcement Learning:** Positive reinforcement learning specifies increasing the tendency that the required behaviour would occur again by adding something. It enhances the strength of the behaviour of the agent and positively impacts it.
- **Negative Reinforcement Learning:** Negative reinforcement learning works exactly opposite to the positive RL. It increases the tendency that the specific behaviour would occur again by avoiding the negative condition.

Real-world Use cases of Reinforcement Learning

- **Video Games:** RL algorithms are much popular in gaming applications. It is used to gain super-human performance. Some popular games that use RL algorithms are AlphaGO and AlphaGO Zero.
- **Resource Management:** The "Resource Management with Deep Reinforcement Learning" paper showed that how to use RL in computer to automatically learn and schedule resources to wait for different jobs in order to minimize average job slowdown.
- **Robotics:** RL is widely being used in Robotics applications. Robots are used in the industrial and manufacturing area, and these robots are made more powerful with reinforcement learning. There are different industries that have their vision of building intelligent robots using AI and Machine learning technology.
- **Text Mining:** Text-mining, one of the great applications of NLP, is now being implemented with the help of Reinforcement Learning by Salesforce company.

Advantages and Disadvantages of Reinforcement Learning

Advantages

- It helps in solving complex real-world problems which are difficult to be solved by general techniques.
- The learning model of RL is similar to the learning of human beings; hence most accurate results can be found.
- Helps in achieving long term results.

Disadvantage

- RL algorithms are not preferred for simple problems.
- RL algorithms require huge data and computations.
- Too much reinforcement learning can lead to an overload of states which can weaken the results.

The curse of dimensionality limits reinforcement learning for real physical systems.

Comparison of Machine Learning Algorithms

Comparing machine learning algorithms is important in itself, but there are some not-so-obvious benefits of comparing various experiments effectively.

- **Better performance**

  The primary objective of model comparison and selection is definitely better performance of the machine learning software/solution. The objective is to narrow down on the best algorithms that suit both the data and the business requirements.

- **Longer lifetime**

  High performance can be short-lived if the chosen model is tightly coupled with the training data and fails to interpret unseen data. So, it’s also important to find the model that understands underlying data patterns so that the predictions are long-lasting and the need for re-training is minimal.

- **Easier retraining**

  When models are evaluated and prepared for comparisons, minute details, and metadata get recorded which come in handy during retraining. For example, if a developer can clearly retrace the reasons behind choosing a model, the causes of model failure will immediately pop out and re-training can start with equal speed.

- **Speedy production**

  With the model details available at hand, it’s easy to narrow down on models that can offer high processing speed and use memory resources optimally. Also during production, several parameters are required to configure the machine learning solutions. Having production-level data can be useful for easily aligning with the production engineers. Moreover, knowing the resource demands of different algorithms, it will also be easier to check their compliance and feasibility with respect to the organization’s allocated assets.

Loss Functions and Metrics for Regression:

- **Mean Square Error:** measures the average of the squares of the errors or deviations, that is, the difference between the estimated and true value. It aids in imposing higher weights on outliers, thus reducing the issue of overfitting.

- **Mean Absolute Error:** It’s the absolute difference between the estimated value and true value. It decreases the weight for outlier errors when compared to the mean squared error.
- **Smooth Absolute Error:** It’s the absolute difference between the estimated value and true value for the predictions lying close to the real value, and it’s the square of the difference between the estimated and the true values of the outliers (or points far off from predicted values). Essentially, it’s a combination of MSE and MAE.

Metrics for Classification:

For every classification model prediction, a matrix called the confusion matrix can be constructed which demonstrates the number of test cases correctly and incorrectly classified. It looks something like this (considering 1 – Positive and 0 – Negative are the target classes):

Table 1.1: Confusion Matrix

	Actual 0	**Actual 1**
Predicted 0	**True Negatives (TN)**	**False Negatives (FN)**
Predicted 1	**False Positives (FP)**	**True Positives (TP)**

- TN: Number of negative cases correctly classified
- TP: Number of positive cases correctly classified
- FN: Number of positive cases incorrectly classified as negative
- FP: Number of negative cases correctly classified as positive

Accuracy

Accuracy is the simplest metric and can be defined as the number of test cases correctly classified divided by the total number of test cases.

Accuracy = (TP + TN)/(TP + FP + TN + FN)

It can be applied to most generic problems but is not very useful when it comes to unbalanced datasets. For instance, if we’re detecting fraud in bank data, the ratio of fraud to non-fraud cases can be 1:99. In such cases, if accuracy is used, the model will turn out to be 99% accurate by predicting all test cases as non-fraud.

This is why accuracy is a false indicator of model health, and for such a case, a metric is required that can focus on the fraud data points.

Precision

Precision is the metric used to identify the correctness of classification.

Precision = TP / (TP + FP)

Intuitively, this equation is the ratio of correct positive classifications to the total number of predicted positive classifications. The greater the fraction, the higher the precision, which means better ability of the model to correctly classify the positive class.

Recall

Recall tells us the number of positive cases correctly identified out of the total number of positive cases.

Recall = TP / (TP + FN)

F1 Score

F1 score is the harmonic mean of Recall and Precision, therefore it balances out the strengths of each. It's useful in cases where both recall and precision can be valuable – like in the identification of plane parts that might require repairing. Here, precision will be required to save on the company's cost (because plane parts are extremely expensive) and recall will be required to ensure that the machinery is stable and not a threat to human lives.

F1 Score = 2 * ((Precision * Recall) / (Precision + Recall))

- To predict the price of the laptops.
- An approach to receive higher accuracy.
- To build a machine learning model to classify the given problem statement.

Preprocessing

When we talk about data, we usually think of some large datasets with a huge number of rows and columns. While that is a likely scenario, it is not always the case — data could be in so many different forms: Structured Tables, Images, Audio files, Videos, etc..

Machines don’t understand free text, image, or video data as it is, they understand 1s and 0s. So it probably won't be good enough if we put on a slideshow of all our images and expect our machine learning model to get trained just by that.

A dataset can be viewed as a collection of data objects, which are often also called as a records, points, vectors, patterns, events, cases, samples, observations, or entities.

Data objects are described by a number of features, that capture the basic characteristics of an object, such as the mass of a physical object or the time at which an event occurred, etc.. Features are often called as variables, characteristics, fields, attributes, or dimensions.

For instance, color, mileage and power can be considered as features of a car. There are different types of features that we can come across when we deal with data.

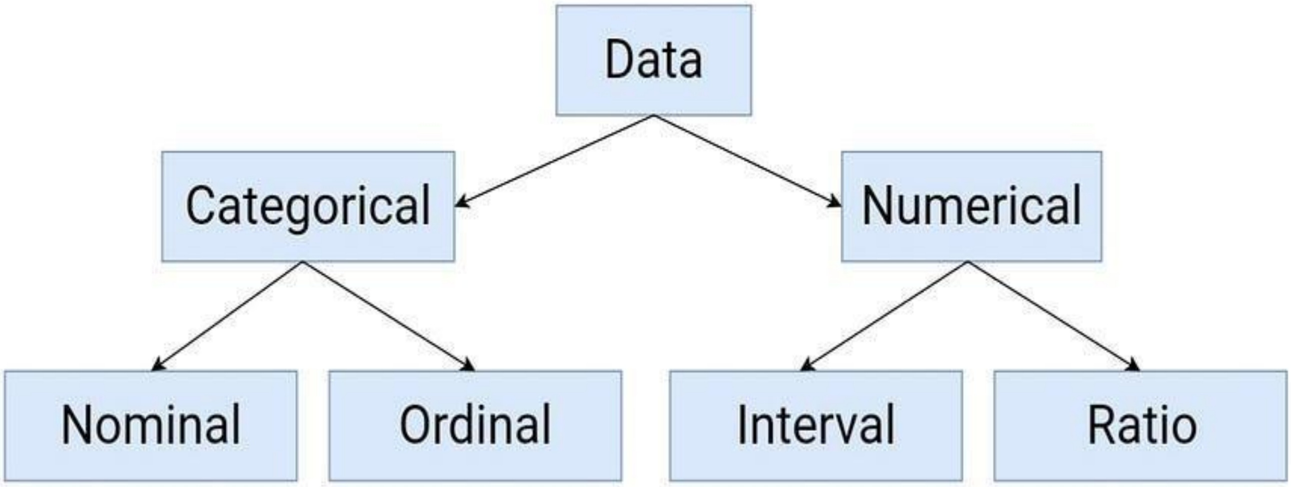
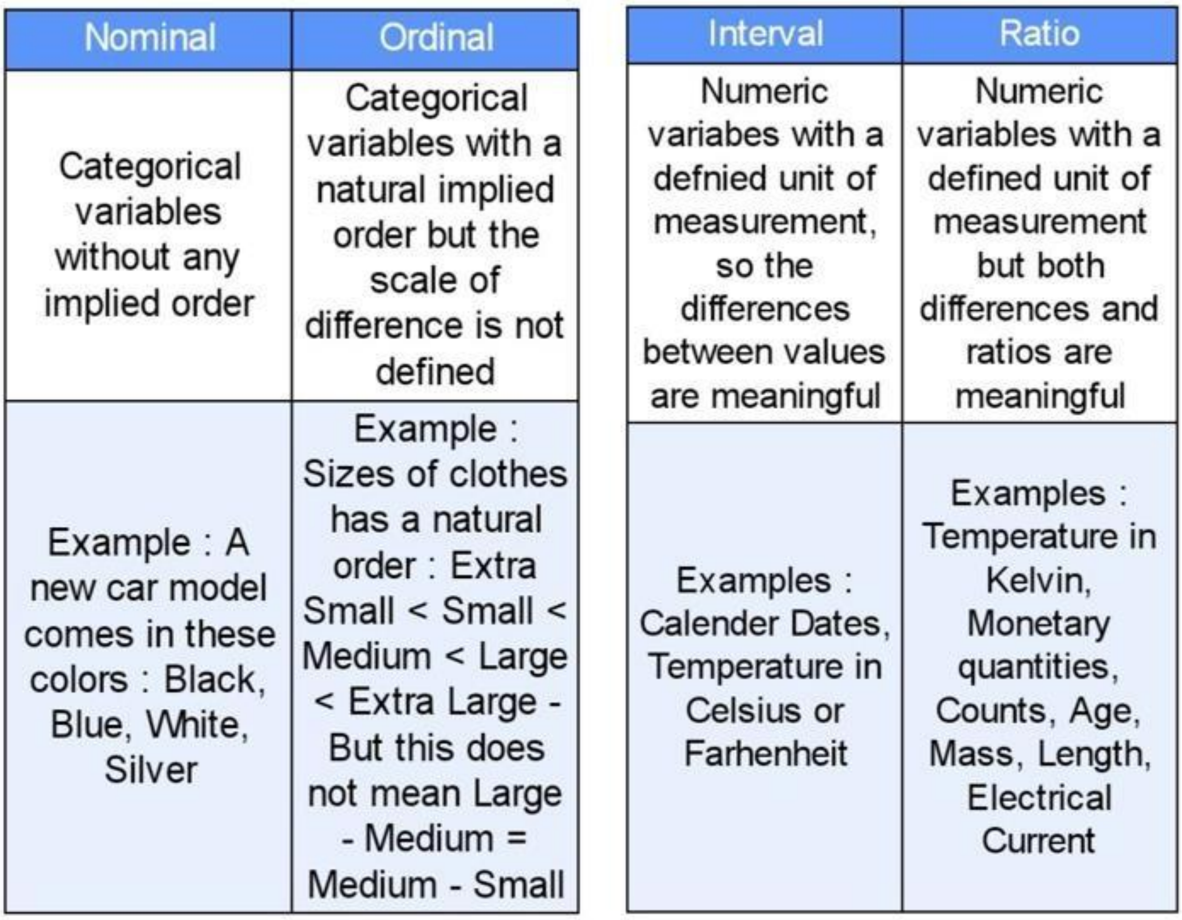


Figure 1.3: Statistical Data Types

Features can be:

- **Categorical:** Features whose values are taken from a defined set of values. For instance, days in a week: {Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday} is a category because its value is always taken from this set. Another example could be the Boolean set : {True, False}
- **Numerical:** Features whose values are continuous or integer-valued. They are represented by numbers and possess most of the properties of numbers. For instance, number of steps you walk in a day, or the speed at which you are driving your car at.



Nominal	Ordinal
Categorical variables without any implied order	Categorical variables with a natural implied order but the scale of difference is not defined
Example : A new car model comes in these colors : Black, Blue, White, Silver	Example : Sizes of clothes has a natural order : Extra Small < Small < Medium < Large < Extra Large - But this does not mean Large - Medium = Medium - Small

Interval	Ratio
Numeric variabes with a defnied unit of measurement, so the differences between values are meaningful	Numeric variables with a defined unit of measurement but both differences and ratios are meaningful
Examples : Calender Dates, Temperature in Celsius or Farhenheit	Examples : Temperature in Kelvin, Monetary quantities, Counts, Age, Mass, Length, Electrical Current

Figure 1.4: Feature Types

The steps of Data Preprocessing: Not all the steps are applicable for each problem, it is highly dependent on the data we are working with, so maybe only a few steps might be required with the dataset. Generally, they are:

- Data Quality Assessment
- Feature Aggregation
- Feature Sampling
- Dimensionality Reduction
- Feature Encoding

Data Quality Assessment

Because data is often taken from multiple sources which are normally not too reliable and that too in different formats, more than half our time is consumed in dealing with data quality issues when working on a machine learning problem. It is simply unrealistic to expect that the data will be perfect. There may be problems due to human error, limitations of measuring devices, or flaws in the data collection process. The methods to deal with the problem :

1. Missing values

It is very much usual to have missing values in your dataset. It may have happened during data collection, or maybe due to some data validation rule, but regardless missing values must be taken into consideration.

- **Eliminate rows with missing data**

  Simple and sometimes effective strategy. Fails if many objects have missing values. If a feature has mostly missing values, then that feature itself can also be eliminated.

- **Estimate missing values**

  If only a reasonable percentage of values are missing, then we can also run simple interpolation methods to fill in those values. However, most common method of dealing with missing values is by filling them in with the mean, median or mode value of the respective feature.

2. Inconsistent values

The data can contain inconsistent values. For instance, the ‘Address’ field contains the ‘Phone number’. It may be due to human error or maybe the information was misread while being scanned from a handwritten form.

It is therefore always advised to perform data assessment like knowing what the data type of the features should be and whether it is the same for all the data objects.

3. Duplicate values

A dataset may include data objects which are duplicates of one another. It may happen when say the same person submits a form more than once. The term deduplication is often used to refer to the process of dealing with duplicates.

In most cases, the duplicates are removed so as to not give that particular data object an advantage or bias, when running machine learning algorithms.

Feature Aggregation

Feature Aggregations are performed so as to take the aggregated values in order to put the data in a better perspective. Think of transactional data, suppose we have day-to-day transactions of a product from recording the daily sales of that product in various store locations over the year. Aggregating the transactions to single store-wide monthly or yearly transactions will help us reducing hundreds or potentially thousands of transactions that occur daily at a specific store, thereby reducing the number of data objects.

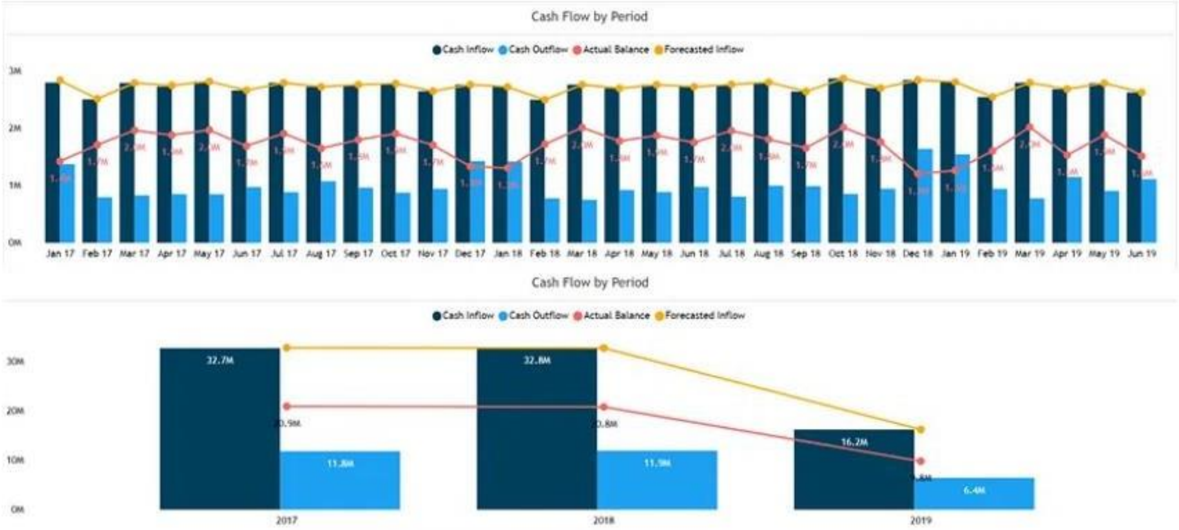


Figure 1.5: Aggregation from monthly to yearly

- This results in reduction of memory consumption and processing time.
- Aggregations provide us with a high-level view of the data as the behavior of groups or aggregates is more stable than individual data objects.

Feature Sampling

Sampling is a very common method for selecting a subset of the dataset that we are analyzing. In most cases, working with the complete dataset can turn out to be too expensive considering the memory and time constraints. Using a sampling algorithm can help us reduce the size of the dataset to a point where we can use a better, but more expensive, machine learning algorithm.

The key principle here is that the sampling should be done in such a manner that the sample generated should have approximately the same properties as the original dataset, meaning that the sample is representative. This involves choosing the correct sample size and sampling strategy.

Simple Random Sampling dictates that there is an equal probability of selecting any particular entity. It has two main variations as well:

- Sampling without Replacement : As each item is selected, it is removed from the set of all the objects that form the total dataset.
- Sampling with Replacement : Items are not removed from the total dataset after getting selected. This means they can get selected more than once.

Dimensionality Reduction

Most real world datasets have a large number of features. For example, consider an image processing problem, we might have to deal with thousands of features, also called as dimensions. As the name suggests, dimensionality reduction aims to reduce the number of features - but not simply by selecting a sample of features from the feature-set, which is something else — Feature Subset Selection or simply Feature Selection.

Conceptually, dimension refers to the number of geometric planes the dataset lies in, which could be high so much so that it cannot be visualized with pen and paper. More the number of such planes, more is the complexity of the dataset.

The Curse of Dimensionality

This refers to the phenomena that generally data analysis tasks become significantly harder as the dimensionality of the data increases. As the dimensionality increases, the number planes occupied by the data increases thus adding more and more sparsity to the data which is difficult to model and visualize.

What dimension reduction essentially does is that it maps the dataset to a lower-dimensional space, which may very well be to a number of planes which can now be visualized, say 2D. The basic objective of techniques which are used for this purpose is to reduce the dimensionality of a dataset by creating new features which are a combination of the old features. In other words, the higher-dimensional feature-space is mapped to a lower-dimensional feature-space. Principal Component Analysis and Singular Value Decomposition are two widely accepted techniques.

A few major benefits of dimensionality reduction are :

- Data Analysis algorithms work better if the dimensionality of the dataset is lower. This is mainly because irrelevant features and noise have now been eliminated.
- The models which are built on top of lower-dimensional data are more understandable and explainable.
- The data may now also get easier to visualize.

Feature Encoding

The whole purpose of data preprocessing is to encode the data in order to bring it to such a state that the machine now understands it.

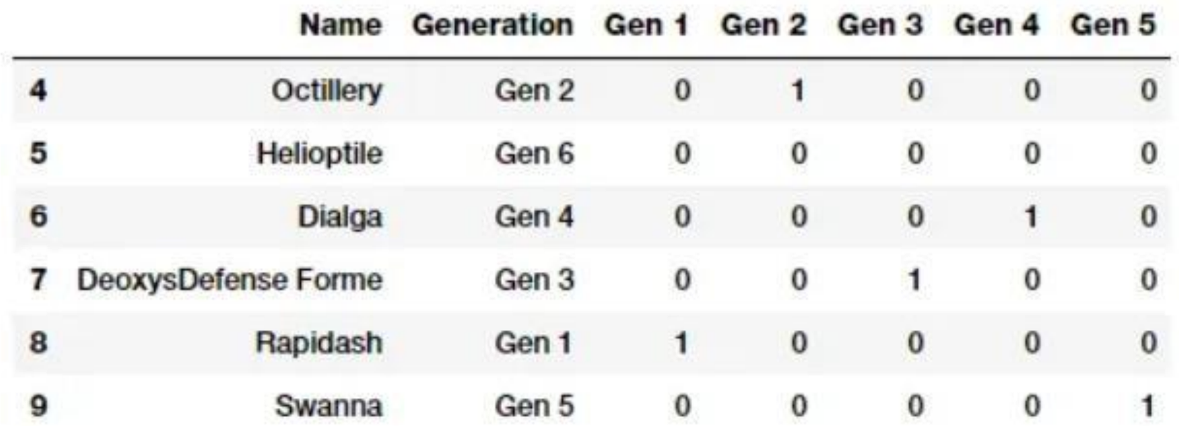
Feature encoding is basically performing transformations on the data such that it can be easily accepted as input for machine learning algorithms while still retaining its original meaning.

There are some general norms or rules which are followed when performing feature encoding.

For Continuous variables

- Nominal: Any one-to-one mapping can be done which retains the meaning. For instance, a permutation of values like in One-Hot Encoding.
- Ordinal: An order-preserving change of values. The notion of small, medium and large can be represented equally well with the help of a new function, that is, <new_value = f(old_value)> - For example, {0, 1, 2} or maybe {1, 2, 3}.

Example of One-hot encoding



	Name	Generation	Gen 1	Gen 2	Gen 3	Gen 4	Gen 5
4	Octillery	Gen 2	0	1	0	0	0
5	Helioptile	Gen 6	0	0	0	0	0
6	Dialga	Gen 4	0	0	0	1	0
7	DeoxysDefense Forme	Gen 3	0	0	1	0	0
8	Rapidash	Gen 1	1	0	0	0	0
9	Swanna	Gen 5	0	0	0	0	1

Figure 1.6: One-hot encoding of the data

For Numeric variables

- Interval: Simple mathematical transformation like using the equation <new_value = a*old_value + b>, a and b being constants. For example, Fahrenheit and Celsius scales, which differ in their Zero values size of a unit, can be encoded in this manner.
- Ratio : These variables can be scaled to any particular measures, of course while still maintaining the meaning and ratio of their values. Simple mathematical transformations work in this case as well, like the transformation <new_value = a*old_value>. For, length can be measured in meters or feet, money can be taken in different currencies.

Train / Validation / Test Split

After feature encoding is done, our dataset is ready for the exciting machine learning algorithms. But before we start deciding the algorithm which should be used, it is always advised to split the dataset into 2 or sometimes 3 parts. Machine Learning algorithms, or any algorithm for that matter, has to be first trained on the data distribution available and then validated and tested, before it can be deployed to deal with real-world data.

- **Training data:** This is the part on which your machine learning algorithms are actually trained to build a model. The model tries to learn the dataset and its various characteristics and intricacies, which also raises the issue of Overfitting v/s Underfitting.

Validation data: This is the part of the dataset which is used to validate our various model fits. In simpler words, we use validation data to choose and improve our model hyperparameters. The model does not learn the validation set but uses it to get to a better state of hyperparameters.

Test data: This part of the dataset is used to test our model hypothesis. It is left untouched and unseen until the model and hyperparameters are decided, and only after that the model is applied on the test data to get an accurate measure of how it would perform when deployed on real-world data.

Training Set
Validation Set
Testing Set

Figure 1.7: Data Split into parts

Split Ratio: Data is split as per a split ratio which is highly dependent on the type of model we are building and the dataset itself. If our dataset and model are such that a lot of training is required, then we use a larger chunk of the data just for training purposes (usually the case) For instance, training on textual data, image data, or video data usually involves thousands of features.

If the model has a lot of hyperparameters that can be tuned, then keeping a higher percentage of data for the validation set is advisable. Models with less number of hyperparameters are easy to tune and update, and so we can keep a smaller validation set.

Like many other things in Machine Learning, the split ratio is highly dependent on the problem we are trying to solve and must be decided after taking into account all the various details about the model and the dataset in hand.

Exploratory Data Analysis

Exploratory Data Analysis is a process of examining or understanding the data and extracting insights or main characteristics of the data. EDA is generally classified into two methods, i.e. graphical analysis and non-graphical analysis.

EDA is very essential because it is a good practice to first understand the problem statement and the various relationships between the data features before getting your hands dirty.

Technically, The primary motive of EDA is to

- Examine the data distribution
- Handling missing values of the dataset(a most common issue with every dataset)
- Handling the outliers
- Removing duplicate data
- Encoding the categorical variables
- Normalizing and Scaling

1.1 Problem Statement

In today's digital age, laptops have become an essential tool for work, education, and entertainment. With so many options available on the market, it can be challenging to choose the right laptop for your needs. The models predict the price of the laptop based on the requirement and the best model is selected based on the root mean square error and r2score.

1.2 Objectives

- To predict the price of the laptop.
- An approach to receive higher accuracy.
- To build a machine learning model to classify the given problem statement.

1.3 Future Scope

I realized that feature scaling is an essential aspect of ML models during this project. The basic concept is to make sure that all of the functionalities are on the same scale. We can expand the existing system with additional analysis methods such as analysis and implementation with other algorithms and enhanced coding.

CHAPTER 2

REQUIREMENTS SPECIFICATION

2.1 SOFTWARE REQUIREMENTS

- Operating system – Windows 7/8/10/11
- Google Collab Environment
- Libraries – NumPy, Pandas, Matplotlib, and seaborn
- Language used is Python

2.2 HARDWARE REQUIREMENTS

- Processor – i3 Processor
- Processor Speed – 1 GHz
- Memory – 2 GB RAM
- 1TB Hard Disk Drive
- Mouse or any other pointing device
- Keyboard
- Display device: Color Monitor

CHAPTER 3

SYSTEM DEFINITION

PROJECT DESCRIPTION

Supervised Machine Learning algorithm can be broadly classified into Regression and Classification Algorithms. Regression algorithms are used to predict the output for continuous values whereas the classification algorithms are used to predict discrete values. In this project regression algorithms are used to predict the price of the laptops.

Regression analysis is often used in finance, investing, and others, and finds out the relationship between a single dependent variable(target variable) dependent on several independent ones. For example, predicting house price, stock market or salary of an employee, etc are the most common regression problems. Here, the target variable is the price of the Laptops.

Regression Algorithms

1. Linear Regression
2. Decision Tree
3. Support Vector Regression
4. Random Forest

Linear regression

Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable(target) based on the given independent variable(s). So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables. Hence, the name of this algorithm is Linear Regression.

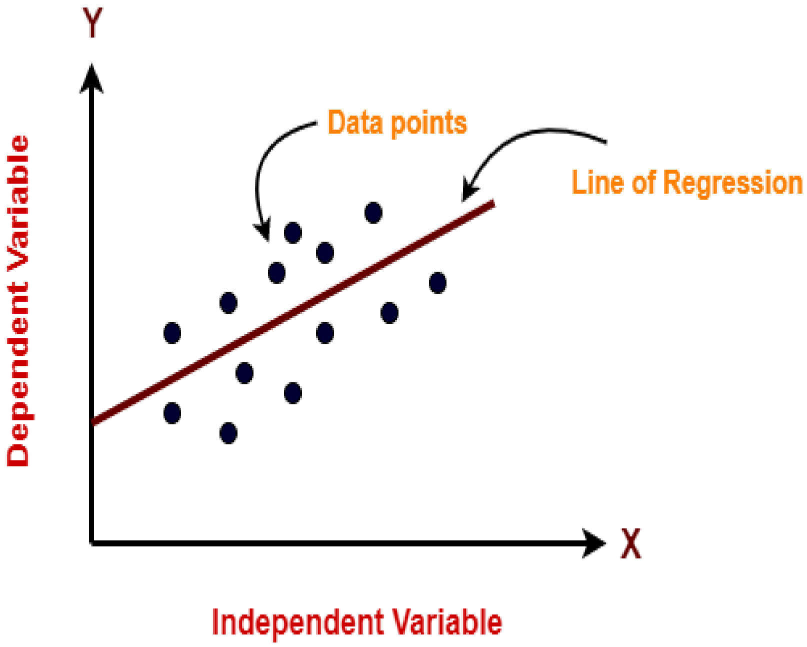


Figure 3.1: Linear Regression Algorithm

In the figure above, on X-axis is the independent variable and on Y-axis is the output. The regression line is the best fit line for a model. And our main objective in this algorithm is to find this best fit line.

Pros:

- Linear Regression is simple to implement.
- Less complexity compared to other algorithms.
- Linear Regression may lead to over-fitting but it can be avoided using some dimensionality reduction techniques, regularization techniques, and cross-validation.

Cons:

- Outliers affect this algorithm badly.
- It over-simplifies real-world problems by assuming a linear relationship among the variables, hence not recommended for practical use-cases.

Decision Tree

The decision tree models can be applied to all those data which contains numerical features and categorical features. Decision trees are good at capturing non-linear interaction between the features and the target variable. Decision trees somewhat match human-level thinking so it's very intuitive to understand the data.

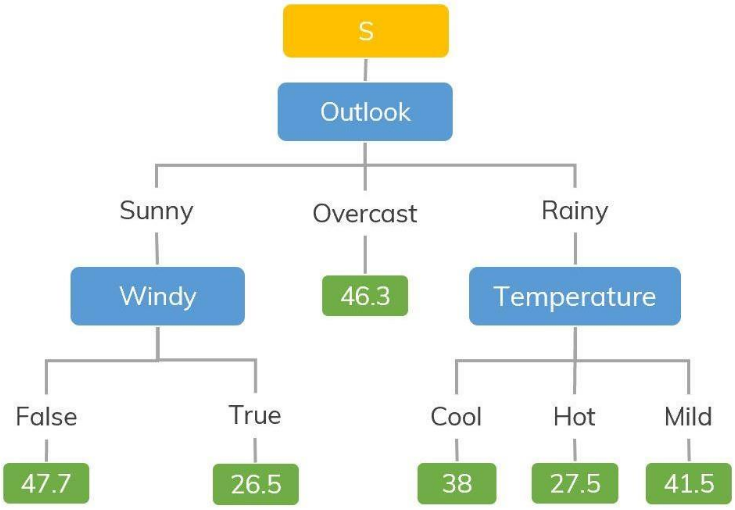


Figure 3.2: Illustration of Decision Tree

For example, if we are classifying how many hours a kid plays in particular weather then the decision tree looks like somewhat this above in the image.

So, in short, a decision tree is a tree where each node represents a feature, each branch represents a decision, and each leaf represents an outcome(numerical value for regression).

How does the Decision Tree algorithm Work?

Step 1. Begin the tree with the root node, says S, which contains the complete dataset.

Step 2. Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step 3. Divide the S into subsets that contains possible values for the best attributes.

Step 4. Generate the decision tree node, which contains the best attribute.

Step 5. Recursively make new decision trees using the subsets of the datasets of the dataset created in step 3.

Continue this process until a stage is reached where you cannot further classify the nodes and calledthe final node as a leaf node.

Pros:

- Easy to understand and interpret, visually intuitive.
- It can work with numerical and categorical features.
- Requires little data preprocessing: no need for one-hot encoding, dummy variables, etc.

Cons:

- It tends to overfit.
- A small change in the data tends to cause a big difference in the tree structure, which causes instability.

Support Vector Regression

SVR uses the same idea of SVM but here it tries to predict the real values. This algorithm uses hyperplanes to segregate the data. In case this separation is not possible then it uses kernel trick where the dimension is increased and then the data points become separable by a hyperplane.

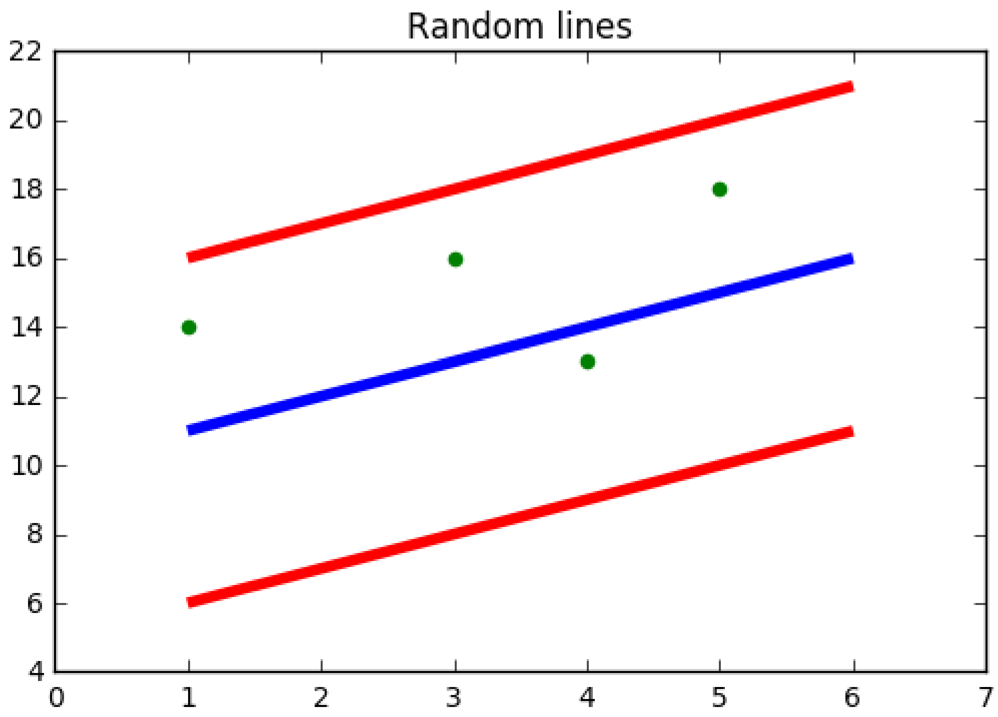


Figure 3.3: Support Vector Regression

In the figure above, the Blue line is the Hyper Plane; Red Line is the Boundary Line

All the data points are within the boundary line(Red Line). The main objective of SVR is to basically consider the points that are within the boundary line.

Pros:

- Robust to outliers.
- Excellent generalization capability
- High prediction accuracy.

Cons:

- Not suitable for large datasets.
- They do not perform very well when the data set has more noise.

Random Forest Regression

Random Forests are an ensemble(combination) of decision trees. It is a Supervised Learning algorithm used for classification and regression. The input data is passed through multiple decision trees. It executes by constructing a different number of decision trees at training time and outputting the class that is the mode of the classes (for classification) or mean prediction (for regression) of the individual trees.

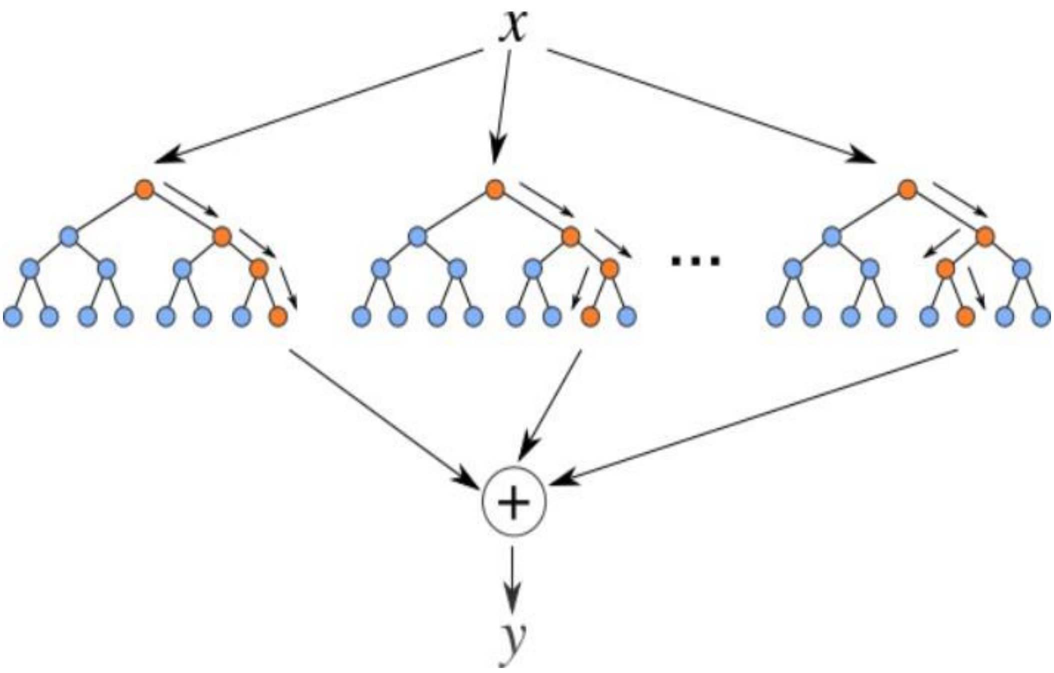


Figure 3.4: Random Forest Regression

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps:

Step 1. Select random K data points from the training set.

Step 2. Build the decision trees associated with the selected data points.

Step 3. Choose the number N for decision trees that you want to build.

Step 4. Repeat Step 1 & 2.

Step 5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Pros:

- Good at learning complex and non-linear relationships
- Very easy to interpret and understand

Cons:

- They are prone to overfitting
- Using larger random forest ensembles to achieve higher performance slows down their speed and then they also need more memory.

WORKING DESCRIPTION

Laptop Selection is a beginner’s example of regression task which involves predicting the price of the laptop based on the specifications of the laptop. The data is extracted from Kaggle.

LAPTOP SELECTION

As in today's digital age, laptops have become an essential tool for work, education, and entertainment. With so many options available on the market, it can be challenging to choose the right laptop for people’s needs. To solve this challenge designing a model which predicts the price of the laptop based on their specification and brands of the laptops.

Primary goals of the analysis are:

- Do an exploratory analysis of the Laptop Selection dataset.
- Do a visualization analysis of the Laptop Selection dataset.
- To predict the price of the laptops.
- Check the accuracy of each regression model.

Context of Dataset

The dataset provides detailed information on 1000 laptops available on Flipkart, including technical specifications, customer reviews and ratings, and prices. Researchers, analysts, and consumers can use this dataset to gain insights into the Indian laptop market, compare different models, and make informed purchase decisions. With this comprehensive dataset, anyone can find the perfect laptop for their needs, whether for work, gaming, or personal use.

Data Preprocessing

Data preprocessing is an important step before using it. It refers to the cleaning, transforming, and integrating of data in order to make it ready for analysis. The goal of data preprocessing is to improve the quality of the data and to make it more suitable for the specific model to train.

In the laptop selection dataset, there are both numerical and categorical features. The categorical features need to be converted to numerical as the models takes only the numerical values.

The categorical features present are image link, name, processor, ram, os, storage
And numerical features are price(in Rs.), display(in inch), rating, number of ratings, number of reviews.

Also, there are many null values in the dataset, they are filled before using the dataset. The null values of numerical values is filled with mean value of the feature whereas the null values of the categorical features is filled with mode value.

Training and Testing Split

Before splitting the data for training and testing, we have to assign the response variable and predictor variable to Y and X respectively. Now we have to split the data in an 80:20 ratio. 80% of thedata will be used for training the models and 20% of the data will be used for testing.

Performing Regression

Prepare the model using the X_train and y_train (training data) using the following algorithms:

- Decision Tree
- Random Forest
- Support Vector Regression

With the prepared model ,test that with the 20% (X_test) testing data and assign that to the y_pred variable Now test the performance of the model using root squared mean error and r2 score.

Libraries Used:

- **NumPy**

  Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

- **Pandas**

  Pandas is an open-source library that is built on top of NumPy library. It is a Python package that offers various data structures and operations for manipulating numerical data and time series. It is mainly popular for importing and analyzing data much easier. Pandas is fast and it has high-performance & productivity for users.

- **Matplotlib**

  Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

- **Seaborn**

  Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and color palettes to make statistical plots more attractive.

TECHNOLOGY USED

Machine Learning

Machine learning (ML) is the study of computer algorithms that improve automatically through experience and using data. It is seen as a part of artificial intelligence. Machine learning algorithms build a model based on sample data, known as "training data”, in order to make predictions or decisions without being explicitly programmed to do so.

Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks.

LANGUAGE USED

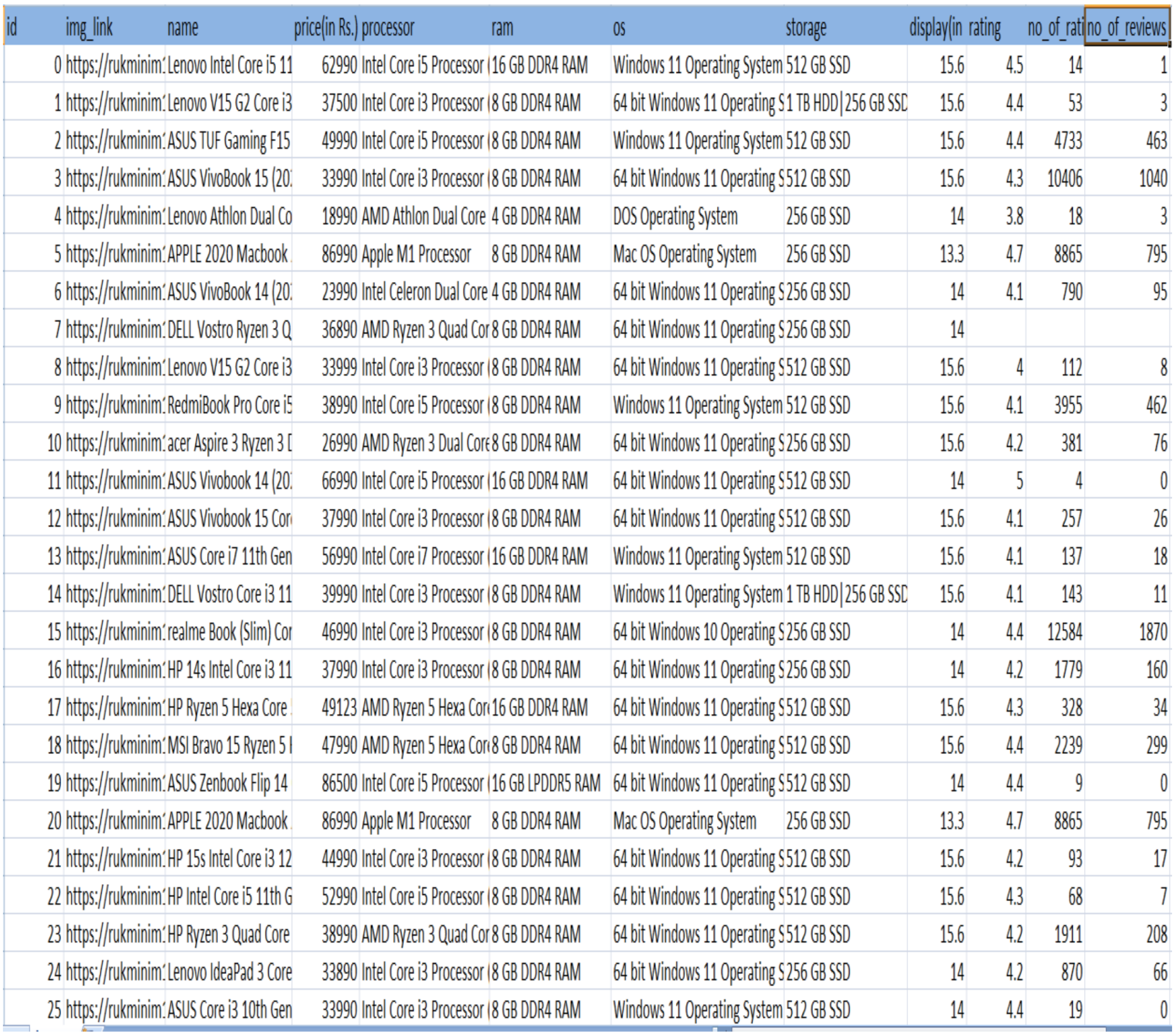
Python

Python is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Industry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like

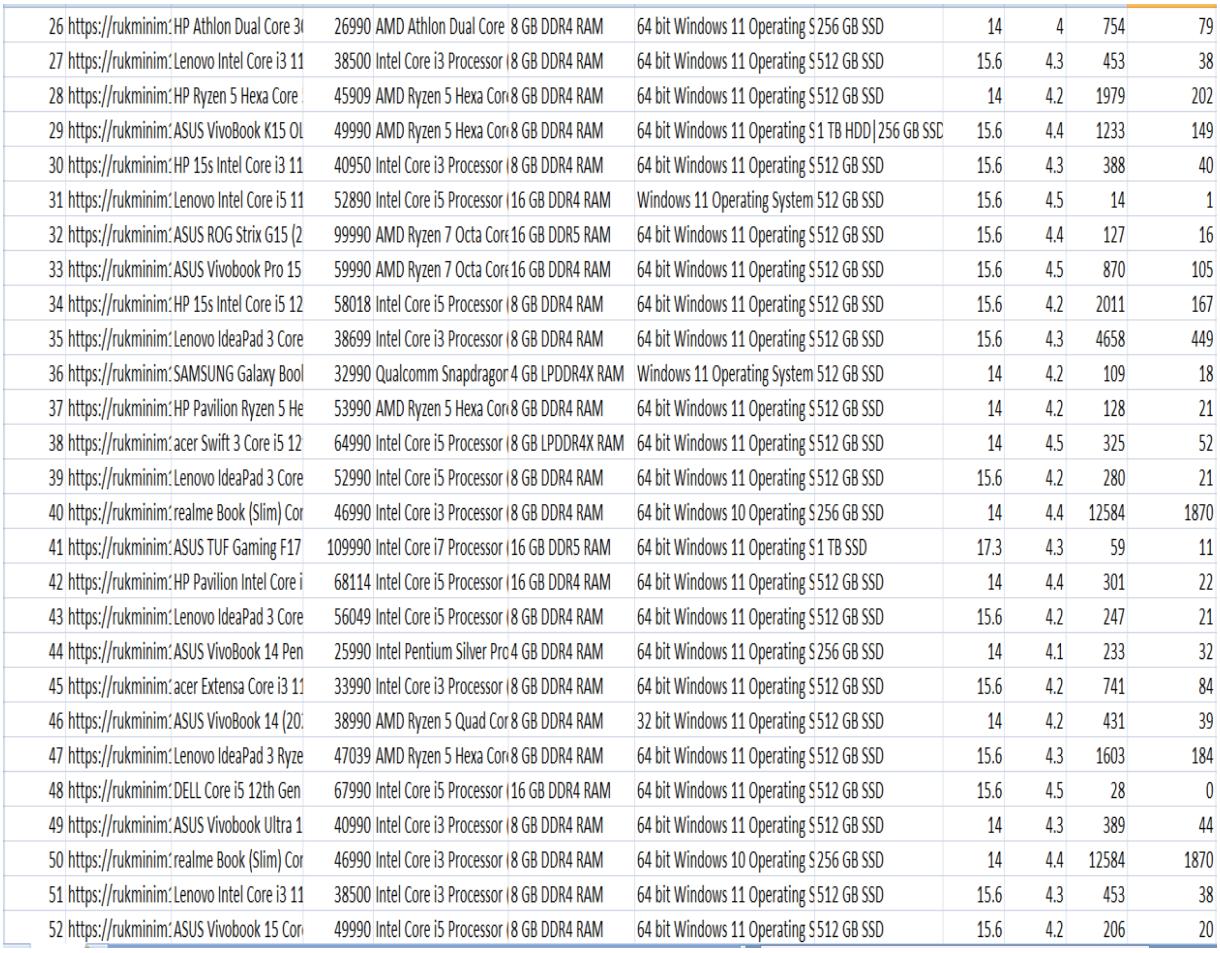
C++ and Java.

DATASET

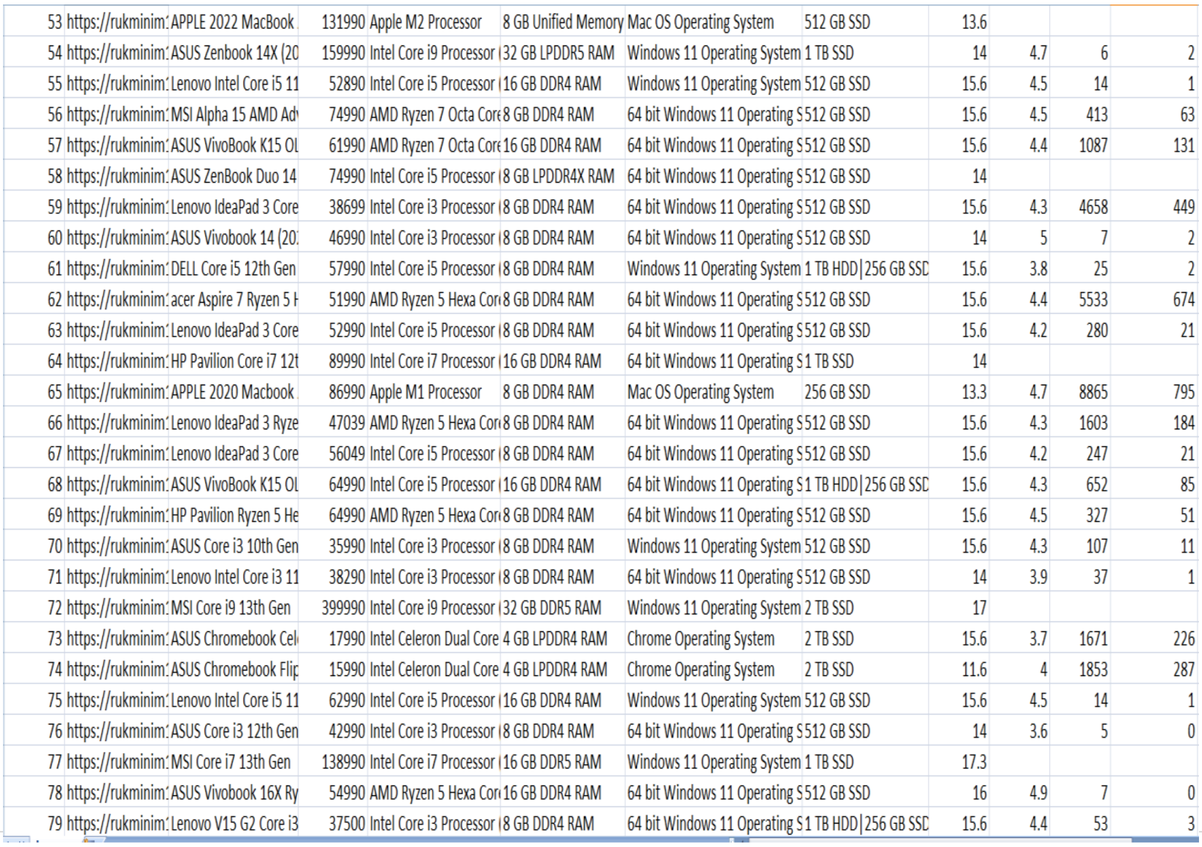
For this project, I have used the dataset extracted from kaggle. The dataset given by the source is fairly accurate and it taken from flipkart about 1000 laptops. The dataset has 984 rows and 12 columns. Snapshot of part of the dataset is given below



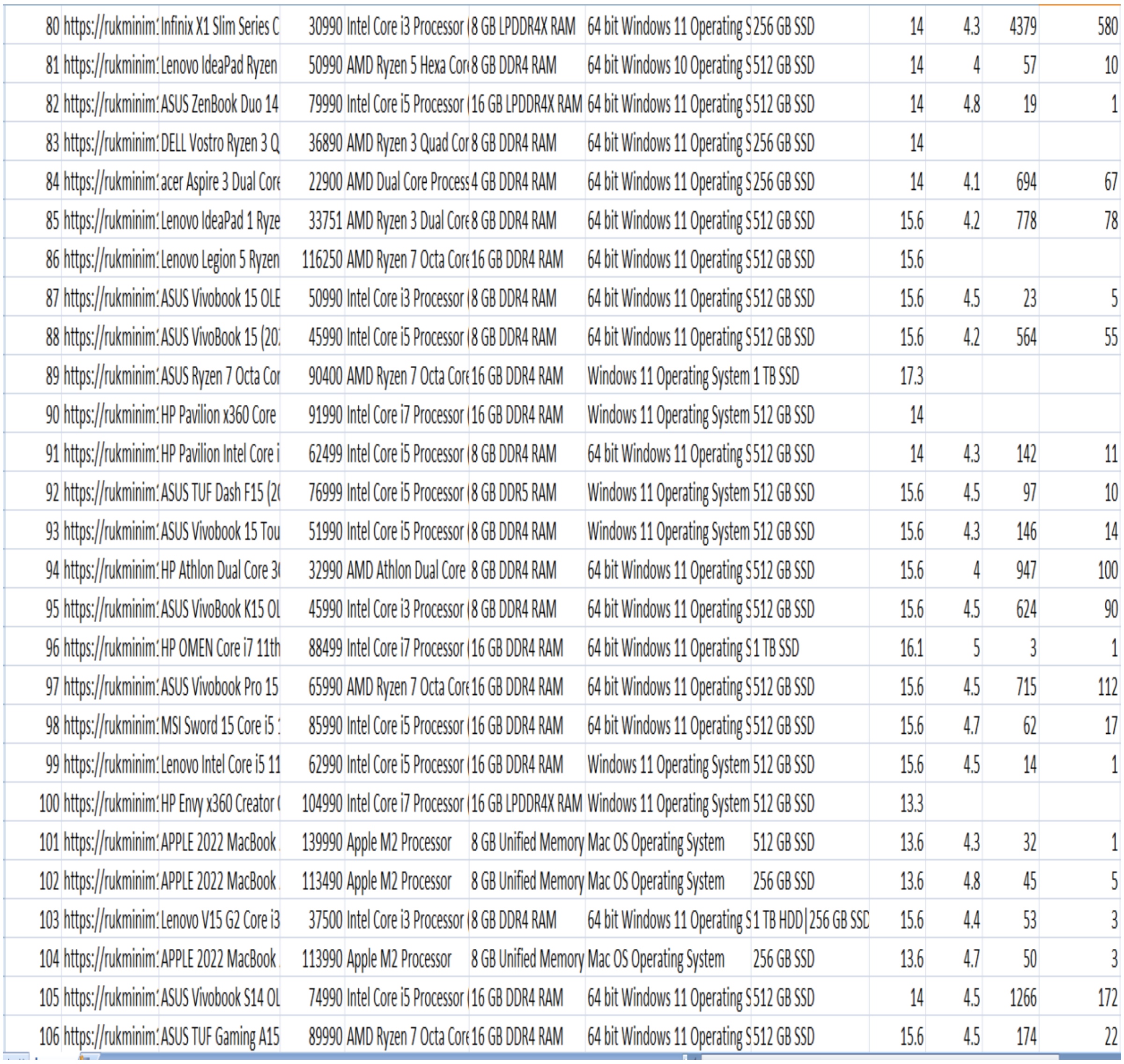
id	img_link	name	price(in Rs.)	processor	ram	os	storage	display(in	rating	no_of_rati	no_of_reviews
0	https://rukminim1	Lenovo Intel Core i5 11	62990	Intel Core i5 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	14	1
1	https://rukminim1	Lenovo V15 G2 Core i3	37500	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB HDD|256 GB SSD	15.6	4.4	53	3
2	https://rukminim1	ASUS TUF Gaming F15	49990	Intel Core i5 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.4	4733	463
3	https://rukminim1	ASUS VivoBook 15 (20	33990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	10406	1040
4	https://rukminim1	Lenovo Athlon Dual Co	18990	AMD Athlon Dual Core	4 GB DDR4 RAM	DOS Operating System	256 GB SSD	14	3.8	18	3
5	https://rukminim1	APPLE 2020 Macbook	86990	Apple M1 Processor	8 GB DDR4 RAM	Mac OS Operating System	256 GB SSD	13.3	4.7	8865	795
6	https://rukminim1	ASUS VivoBook 14 (20	23990	Intel Celeron Dual Core	4 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4.1	790	95
7	https://rukminim1	DELL Vostro Ryzen 3 Q	36890	AMD Ryzen 3 Quad Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14			
8	https://rukminim1	Lenovo V15 G2 Core i3	33999	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4	112	8
9	https://rukminim1	RedmiBook Pro Core i5	38990	Intel Core i5 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.1	3955	462
10	https://rukminim1	acer Aspire 3 Ryzen 3 D	26990	AMD Ryzen 3 Dual Core	8 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	15.6	4.2	381	76
11	https://rukminim1	ASUS Vivobook 14 (20	66990	Intel Core i5 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	5	4	0
12	https://rukminim1	ASUS Vivobook 15 Cor	37990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.1	257	26
13	https://rukminim1	ASUS Core i7 11th Gen	56990	Intel Core i7 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.1	137	18
14	https://rukminim1	DELL Vostro Core i3 11	39990	Intel Core i3 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	1 TB HDD|256 GB SSD	15.6	4.1	143	11
15	https://rukminim1	realme Book (Slim) Cor	46990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 10 Operating S	256 GB SSD	14	4.4	12584	1870
16	https://rukminim1	HP 14s Intel Core i3 11	37990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4.2	1779	160
17	https://rukminim1	HP Ryzen 5 Hexa Core	49123	AMD Ryzen 5 Hexa Cor	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	328	34
18	https://rukminim1	MSI Bravo 15 Ryzen 5	47990	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.4	2239	299
19	https://rukminim1	ASUS Zenbook Flip 14	86500	Intel Core i5 Processor (	16 GB LPDDR5 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.4	9	0
20	https://rukminim1	APPLE 2020 Macbook	86990	Apple M1 Processor	8 GB DDR4 RAM	Mac OS Operating System	256 GB SSD	13.3	4.7	8865	795
21	https://rukminim1	HP 15s Intel Core i3 12	44990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	93	17
22	https://rukminim1	HP Intel Core i5 11th G	52990	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	68	7
23	https://rukminim1	HP Ryzen 3 Quad Core	38990	AMD Ryzen 3 Quad Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	1911	208
24	https://rukminim1	Lenovo IdeaPad 3 Core	33890	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4.2	870	66
25	https://rukminim1	ASUS Core i3 10th Gen	33990	Intel Core i3 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	14	4.4	19	0



26	https://rukminim1	HP Athlon Dual Core 3	26990	AMD Athlon Dual Core	8 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4	754	79
27	https://rukminim1	Lenovo Intel Core i3 11	38500	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	453	38
28	https://rukminim1	HP Ryzen 5 Hexa Core	45909	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.2	1979	202
29	https://rukminim1	ASUS VivoBook K15 OL	49990	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB HDD|256 GB SSD	15.6	4.4	1233	149
30	https://rukminim1	HP 15s Intel Core i3 11	40950	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	388	40
31	https://rukminim1	Lenovo Intel Core i5 11	52890	Intel Core i5 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	14	1
32	https://rukminim1	ASUS ROG Strix G15 (2	99990	AMD Ryzen 7 Octa Core	16 GB DDR5 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.4	127	16
33	https://rukminim1	ASUS Vivobook Pro 15	59990	AMD Ryzen 7 Octa Core	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	870	105
34	https://rukminim1	HP 15s Intel Core i5 12	58018	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	2011	167
35	https://rukminim1	Lenovo IdeaPad 3 Core	38699	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	4658	449
36	https://rukminim1	SAMSUNG Galaxy Bool	32990	Qualcomm Snapdragon	4 GB LPDDR4X RAM	Windows 11 Operating System	512 GB SSD	14	4.2	109	18
37	https://rukminim1	HP Pavilion Ryzen 5 He	53990	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.2	128	21
38	https://rukminim1	acer Swift 3 Core i5 12	64990	Intel Core i5 Processor (	8 GB LPDDR4X RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.5	325	52
39	https://rukminim1	Lenovo IdeaPad 3 Core	52990	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	280	21
40	https://rukminim1	realme Book (Slim) Cor	46990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 10 Operating S	256 GB SSD	14	4.4	12584	1870
41	https://rukminim1	ASUS TUF Gaming F17	109990	Intel Core i7 Processor (	16 GB DDR5 RAM	64 bit Windows 11 Operating S	1 TB SSD	17.3	4.3	59	11
42	https://rukminim1	HP Pavilion Intel Core i	68114	Intel Core i5 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.4	301	22
43	https://rukminim1	Lenovo IdeaPad 3 Core	56049	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	247	21
44	https://rukminim1	ASUS VivoBook 14 Pen	25990	Intel Pentium Silver Pro	4 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4.1	233	32
45	https://rukminim1	acer Extensa Core i3 11	33990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	741	84
46	https://rukminim1	ASUS VivoBook 14 (20	38990	AMD Ryzen 5 Quad Cor	8 GB DDR4 RAM	32 bit Windows 11 Operating S	512 GB SSD	14	4.2	431	39
47	https://rukminim1	Lenovo IdeaPad 3 Ryze	47039	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	1603	184
48	https://rukminim1	DELL Core i5 12th Gen	67990	Intel Core i5 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	28	0
49	https://rukminim1	ASUS Vivobook Ultra 1	40990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.3	389	44
50	https://rukminim1	realme Book (Slim) Cor	46990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 10 Operating S	256 GB SSD	14	4.4	12584	1870
51	https://rukminim1	Lenovo Intel Core i3 11	38500	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	453	38
52	https://rukminim1	ASUS Vivobook 15 Cor	49990	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	206	20



53	https://rukminim1	APPLE 2022 MacBook	131990	Apple M2 Processor	8 GB Unified Memory	Mac OS Operating System	512 GB SSD	13.6			
54	https://rukminim1	ASUS Zenbook 14X (20	159990	Intel Core i9 Processor (	32 GB LPDDR5 RAM	Windows 11 Operating System	1 TB SSD	14	4.7	6	2
55	https://rukminim1	Lenovo Intel Core i5 11	52890	Intel Core i5 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	14	1
56	https://rukminim1	MSI Alpha 15 AMD Ad	74990	AMD Ryzen 7 Octa Core	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	413	63
57	https://rukminim1	ASUS VivoBook K15 OL	61990	AMD Ryzen 7 Octa Core	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.4	1087	131
58	https://rukminim1	ASUS ZenBook Duo 14	74990	Intel Core i5 Processor (	8 GB LPDDR4X RAM	64 bit Windows 11 Operating S	512 GB SSD	14			
59	https://rukminim1	Lenovo IdeaPad 3 Core	38699	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	4658	449
60	https://rukminim1	ASUS Vivobook 14 (20	46990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	5	7	2
61	https://rukminim1	DELL Core i5 12th Gen	57990	Intel Core i5 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	1 TB HDD|256 GB SSD	15.6	3.8	25	2
62	https://rukminim1	acer Aspire 7 Ryzen 5 H	51990	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.4	5533	674
63	https://rukminim1	Lenovo IdeaPad 3 Core	52990	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	280	21
64	https://rukminim1	HP Pavilion Core i7 12t	89990	Intel Core i7 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB SSD	14			
65	https://rukminim1	APPLE 2020 Macbook	86990	Apple M1 Processor	8 GB DDR4 RAM	Mac OS Operating System	256 GB SSD	13.3	4.7	8865	795
66	https://rukminim1	Lenovo IdeaPad 3 Ryze	47039	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.3	1603	184
67	https://rukminim1	Lenovo IdeaPad 3 Core	56049	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	247	21
68	https://rukminim1	ASUS VivoBook K15 OL	64990	Intel Core i5 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB HDD|256 GB SSD	15.6	4.3	652	85
69	https://rukminim1	HP Pavilion Ryzen 5 He	64990	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	327	51
70	https://rukminim1	ASUS Core i3 10th Gen	35990	Intel Core i3 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.3	107	11
71	https://rukminim1	Lenovo Intel Core i3 11	38290	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	3.9	37	1
72	https://rukminim1	MSI Core i9 13th Gen	399990	Intel Core i9 Processor (	32 GB DDR5 RAM	Windows 11 Operating System	2 TB SSD	17			
73	https://rukminim1	ASUS Chromebook Cel	17990	Intel Celeron Dual Core	4 GB LPDDR4 RAM	Chrome Operating System	2 TB SSD	15.6	3.7	1671	226
74	https://rukminim1	ASUS Chromebook Flip	15990	Intel Celeron Dual Core	4 GB LPDDR4 RAM	Chrome Operating System	2 TB SSD	11.6	4	1853	287
75	https://rukminim1	Lenovo Intel Core i5 11	62990	Intel Core i5 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	14	1
76	https://rukminim1	ASUS Core i3 12th Gen	42990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	3.6	5	0
77	https://rukminim1	MSI Core i7 13th Gen	138990	Intel Core i7 Processor (	16 GB DDR5 RAM	Windows 11 Operating System	1 TB SSD	17.3			
78	https://rukminim1	ASUS Vivobook 16X Ry	54990	AMD Ryzen 5 Hexa Cor	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	16	4.9	7	0
79	https://rukminim1	Lenovo V15 G2 Core i3	37500	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB HDD|256 GB SSD	15.6	4.4	53	3



80	https://rukminim1	Infinix X1 Slim Series C	30990	Intel Core i3 Processor (	8 GB LPDDR4X RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4.3	4379	580
81	https://rukminim1	Lenovo IdeaPad Ryzen	50990	AMD Ryzen 5 Hexa Cor	8 GB DDR4 RAM	64 bit Windows 10 Operating S	512 GB SSD	14	4	57	10
82	https://rukminim1	ASUS ZenBook Duo 14	79990	Intel Core i5 Processor (	16 GB LPDDR4X RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.8	19	1
83	https://rukminim1	DELL Vostro Ryzen 3 Q	36890	AMD Ryzen 3 Quad Cor	8 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14			
84	https://rukminim1	acer Aspire 3 Dual Core	22900	AMD Dual Core Process	4 GB DDR4 RAM	64 bit Windows 11 Operating S	256 GB SSD	14	4.1	694	67
85	https://rukminim1	Lenovo IdeaPad 1 Ryze	33751	AMD Ryzen 3 Dual Core	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	778	78
86	https://rukminim1	Lenovo Legion 5 Ryzen	116250	AMD Ryzen 7 Octa Core	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6			
87	https://rukminim1	ASUS Vivobook 15 OLE	50990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	23	5
88	https://rukminim1	ASUS VivoBook 15 (20	45990	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.2	564	55
89	https://rukminim1	ASUS Ryzen 7 Octa Cor	90400	AMD Ryzen 7 Octa Core	16 GB DDR4 RAM	Windows 11 Operating System	1 TB SSD	17.3			
90	https://rukminim1	HP Pavilion x360 Core	91990	Intel Core i7 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	14			
91	https://rukminim1	HP Pavilion Intel Core i	62499	Intel Core i5 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.3	142	11
92	https://rukminim1	ASUS TUF Dash F15 (2(	76999	Intel Core i5 Processor (	8 GB DDR5 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	97	10
93	https://rukminim1	ASUS Vivobook 15 Tou	51990	Intel Core i5 Processor (	8 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.3	146	14
94	https://rukminim1	HP Athlon Dual Core 3(	32990	AMD Athlon Dual Core	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4	947	100
95	https://rukminim1	ASUS VivoBook K15 OL	45990	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	624	90
96	https://rukminim1	HP OMEN Core i7 11th	88499	Intel Core i7 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB SSD	16.1	5	3	1
97	https://rukminim1	ASUS Vivobook Pro 15	65990	AMD Ryzen 7 Octa Core	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	715	112
98	https://rukminim1	MSI Sword 15 Core i5 1	85990	Intel Core i5 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.7	62	17
99	https://rukminim1	Lenovo Intel Core i5 11	62990	Intel Core i5 Processor (	16 GB DDR4 RAM	Windows 11 Operating System	512 GB SSD	15.6	4.5	14	1
100	https://rukminim1	HP Envy x360 Creator (	104990	Intel Core i7 Processor (	16 GB LPDDR4X RAM	Windows 11 Operating System	512 GB SSD	13.3			
101	https://rukminim1	APPLE 2022 MacBook	139990	Apple M2 Processor	8 GB Unified Memory	Mac OS Operating System	512 GB SSD	13.6	4.3	32	1
102	https://rukminim1	APPLE 2022 MacBook	113490	Apple M2 Processor	8 GB Unified Memory	Mac OS Operating System	256 GB SSD	13.6	4.8	45	5
103	https://rukminim1	Lenovo V15 G2 Core i3	37500	Intel Core i3 Processor (	8 GB DDR4 RAM	64 bit Windows 11 Operating S	1 TB HDD|256 GB SSD	15.6	4.4	53	3
104	https://rukminim1	APPLE 2022 MacBook	113990	Apple M2 Processor	8 GB Unified Memory	Mac OS Operating System	256 GB SSD	13.6	4.7	50	3
105	https://rukminim1	ASUS Vivobook S14 OL	74990	Intel Core i5 Processor (	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	14	4.5	1266	172
106	https://rukminim1	ASUS TUF Gaming A15	89990	AMD Ryzen 7 Octa Core	16 GB DDR4 RAM	64 bit Windows 11 Operating S	512 GB SSD	15.6	4.5	174	22

CHAPTER 4

IMPLEMENTATION

The implementation is done using python language and executed in jupyter Notebook.

```
# importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_squared_error,r2_score

# understanding the data
data = pd.read_csv('laptops.csv')
data.head()
data.shape
data.info()
data.describe()

# EDA for analyzing the data
# dropping unnecessary columns
data.drop(['Unnamed: 0','img_link'],axis=1,inplace=True)
data.fillna(0, inplace=True)
data.info()
```

```
#price range
plt.figure(figsize=(12,6))
sns.histplot(data = data, x = 'price(in Rs.)', kde = True,bins=20)
plt.title('Price Range of the Laptops')
plt.show()

# Most Popular brands
brand = []
for i in data['name']:
    brand.append(i.split(' ')[0].capitalize())
data['brand'] = brand

plt.figure(figsize=(12,6))
data.groupby('brand').size().sort_values(ascending=False).head(5).plot(kind = 'bar',color =
sns.color_palette('Paired'))
plt.xlabel('Laptop Brand')
plt.ylabel('Number of Laptops')
plt.title('Top 5 most popular laptops brand')
plt.show()

# Ram
ram = []
for i in data['ram']:
    ram.append(int(i.split(' ')[0]))

data['ramSize'] = ram

plt.figure(figsize=(12,6))
data.groupby('ramSize').size().sort_values(ascending=False).plot(kind = 'bar',color =
sns.color_palette('Blues'))
plt.xlabel('Ram Size in GB')
plt.ylabel('Number of Laptops')
plt.show()
```

```
#processor
plt.figure(figsize=(12,6))
df.groupby('processor').size().sort_values(ascending = False).head(6).plot(kind = 'bar', color = sns.color_palette("coolwarm"))
plt.xlabel("Processor")
plt.ylabel("Frequency")
plt.title("Top 6 Most Frequent Processor")
plt.show()

# OS
os = []
for i in data['os']:
    os.append(i.split('bit')[-1].strip())

data['operating_system'] = os
data.drop(['name','os'],axis=1,inplace=True)

plt.figure(figsize=(12,6))
data.groupby('operating_system').size().sort_values(ascending = False).plot(kind = 'bar', color = sns.color_palette("hsv_r"))
plt.xlabel("Operating System")
plt.ylabel("Frequency")
plt.title("Most Frequent OS")
plt.show()

# Relation between price and ratings
new_df = data[data['no_of_ratings'] != 0]

plt.figure(figsize=(12,6))
sns.scatterplot(data = new_df,x = 'price(in Rs.)', y = 'no_of_ratings')
plt.xlabel('Price in INR')
plt.ylabel("Number of Ratings")
plt.title("Price VS Number of Ratings")
plt.show()
```

```
# categorical and numerical features
categorical_features = data.columns[(data.dtypes == 'object') == True].to_list()

numerical_df = data.drop(categorical_features, axis=1)
categorical_df = data[categorical_features]



# Price Dependency
# Processor
plt.figure(figsize=(15,5))
sns.scatterplot(x=categorical_df['processor'], y=numerical_df['price(in Rs.)'])
plt.xticks(rotation=70, horizontalalignment='right',
           fontsize=9)

print()

# ram
plt.figure(figsize=(10,5))
sns.scatterplot(x=categorical_df['ram'], y=numerical_df['price(in Rs.)'])
plt.xticks(rotation=40, horizontalalignment='right',
           fontsize=9)

print()

# Sorted list of laptop brands based on the ratings
laptops_sorted_by_rating = data.sort_values(['rating'])[::-1]

for name, rating in zip(laptops_sorted_by_rating['brand'][:20],
laptops_sorted_by_rating['rating'][:20]):
    print(f'{name} - {rating}')

plt.figure(figsize=(10,4))
```

```
sns.barplot(x=laptops_sorted_by_rating['brand'][:20],
y=laptops_sorted_by_rating['no_of_ratings'][:20])
plt.xticks(rotation=40, horizontalalignment='right',
          fontsize=10)

print()



plt.figure(figsize=(10,4))
sns.barplot(x=laptops_sorted_by_rating['brand'][:20],
y=laptops_sorted_by_rating['no_of_reviews'][:20])
plt.xticks(rotation=40, horizontalalignment='right',
          fontsize=10)

# Numerical features
sns.pairplot(numerical_df)

#Preprocessing

#ram
data['ramType'] = data.ram.str.split().apply(lambda x : ' '.join(x[2:-1]))
data.drop('ram',axis=1,inplace=True)

#storage
data = data[data["storage"].str.contains("PCI-e SSD (NVMe) ready,Silver-Lining Print
Keyboard,Matrix Display (Extend),Cooler Boost 5,Hi-Res Audio,Nahimic 3,144Hz
Panel,Thin Bezel,RGB Gaming Keyboard,Speaker Tuning Engine,MSI Center",
regex=False) == False]
data = data[data["storage"].str.contains("PCI-e Gen4 SSD?SHIFT?Matrix Display
(Extend)?Cooler Boost 3?Thunderbolt 4?Finger Print Security?True Color 2.0?Hi-Res
Audio?Nahimic 3? 4-Sided Thin bezel?MSI Center?Silky Smooth Touchpad?Military-Grade
Durability", regex=False) == False]
```

```
data["storage"] = data["storage"].str.replace(' GB', '')
data["storage"] = data["storage"].str.replace(' TB', '000')

new = data["storage"].str.split("|", n = 1, expand = True)

data["first"]= new[0]
data["first"]=data["first"].str.strip()
data["second"]= new[1]

data["Layer1HDD"] = data["first"].apply(lambda x: 1 if "HDD" in x else 0)
data["Layer1SSD"] = data["first"].apply(lambda x: 1 if "SSD" in x else 0)

data['first'] = data['first'].str.replace(r'\D', '',regex=True)

data["second"].fillna("0", inplace = True)

data["Layer2HDD"] = data["second"].apply(lambda x: 1 if "HDD" in x else 0)
data["Layer2SSD"] = data["second"].apply(lambda x: 1 if "SSD" in x else 0)

data['second'] = data['second'].str.replace(r'\D', '',regex=True)

data["first"] = data["first"].astype(int)
data["second"] = data["second"].astype(int)

data["HDD"]=(data["first"]*data["Layer1HDD"]+data["second"]*data["Layer2HDD"])
data["SSD"]=(data["first"]*data["Layer1SSD"]+data["second"]*data["Layer2SSD"])
data.drop(columns=['first', 'second', 'Layer1HDD', 'Layer1SSD','Layer2HDD',
'Layer2SSD'],inplace=True)

data.drop('storage',axis=1,inplace=True)

# encoding the categorical features using onehot encoding
data.operating_system.value_counts()
data.ramType.value_counts()
data.brand.value_counts()
dummies = pd.get_dummies(data.operating_system)
dummies
data = pd.concat([data,dummies],axis=1)
data.drop('operating_system',axis=1,inplace=True)

dummies = pd.get_dummies(data.brand)
data = pd.concat([data,dummies],axis=1)
data.drop('brand',axis=1,inplace=True)
data.head()

dummies = pd.get_dummies(data.ramType)
data = pd.concat([data,dummies],axis=1)
data.drop('ramType',axis=1,inplace=True)
data.head()

data.info()

data.processor.value_counts()
data['cpu'] = data['processor'].apply(lambda x:" ".join(x.split()[0:3]))
data.cpu.value_counts()

dummies = pd.get_dummies(data.cpu)
data = pd.concat([data,dummies],axis=1)
data.drop('processor',axis=1,inplace=True)
data.head()

data.drop('cpu',axis=1,inplace=True)

data.info()


#splitting dataset into train and test datasets
```

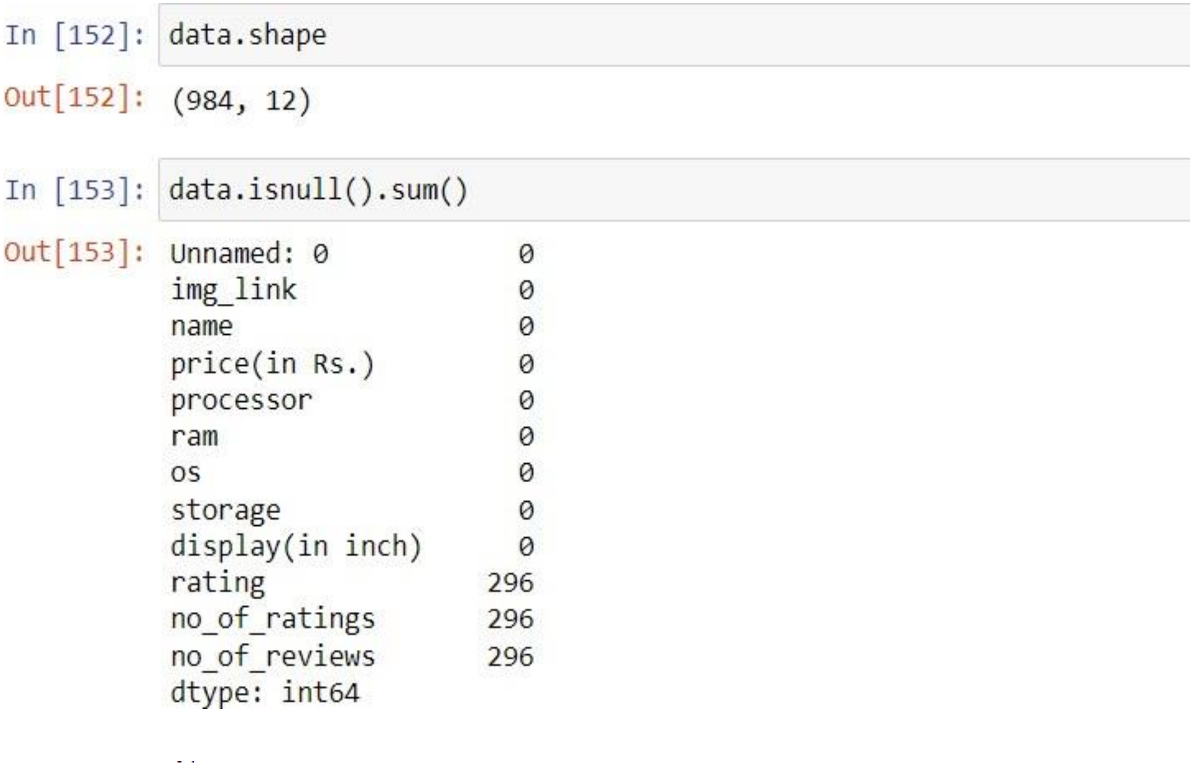
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20)
```

```
len(x_train),len(y_train),len(x_test),len(y_test)

#standardizing data
x_sc = StandardScaler()
y_sc = StandardScaler()

y__train = y_train.reshape(len(y_train),1)
y__test = y_test.reshape(len(y_test), 1)

x__train = x_sc.fit_transform(x_train)
y__train = y_sc.fit_transform(y__train)
x__test = x_sc.transform(x_test)
y_test = y_sc.transform(y__test)

#models

#decision tree
dreg = DecisionTreeRegressor()
dreg.fit(x__train,y__train.reshape(len(y__train)))
dtpred = dreg.predict(x__test)

#random forest
rreg = RandomForestRegressor(n_estimators=10)
rreg.fit(x__train,y__train.reshape(len(y__train)))
rfpred = rreg.predict(x__test)

#svr
sreg = SVR(kernel='linear')
sreg.fit(x__train,y__train.reshape(len(y__train)))
srpred = sreg.predict(x__test)
```

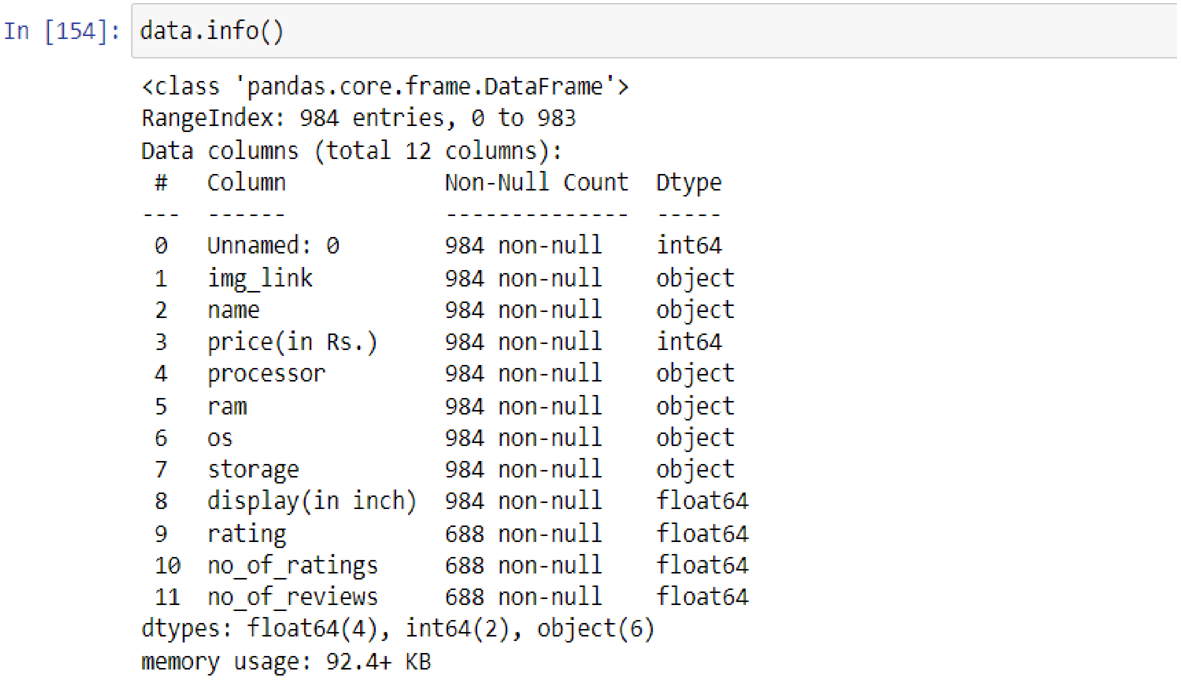
```
#rsme and r2score

#decision tree
rsme_dt = np.sqrt(mean_squared_error(y_test,dtpred))
rsdt = r2_score(y_test,dtpred)
rsme_dt,rsdt

#random forest
rsme_rf = np.sqrt(mean_squared_error(y_test,rfpred))
rsrf = r2_score(y_test,rfpred)
rsme_rf,rsrf

#svr
rsme_sr = np.sqrt(mean_squared_error(y_test,srpred))
rssr = r2_score(y_test,srpred)
rsme_sr,rssr
```

SNAPSHOTS

Before Preprocessing

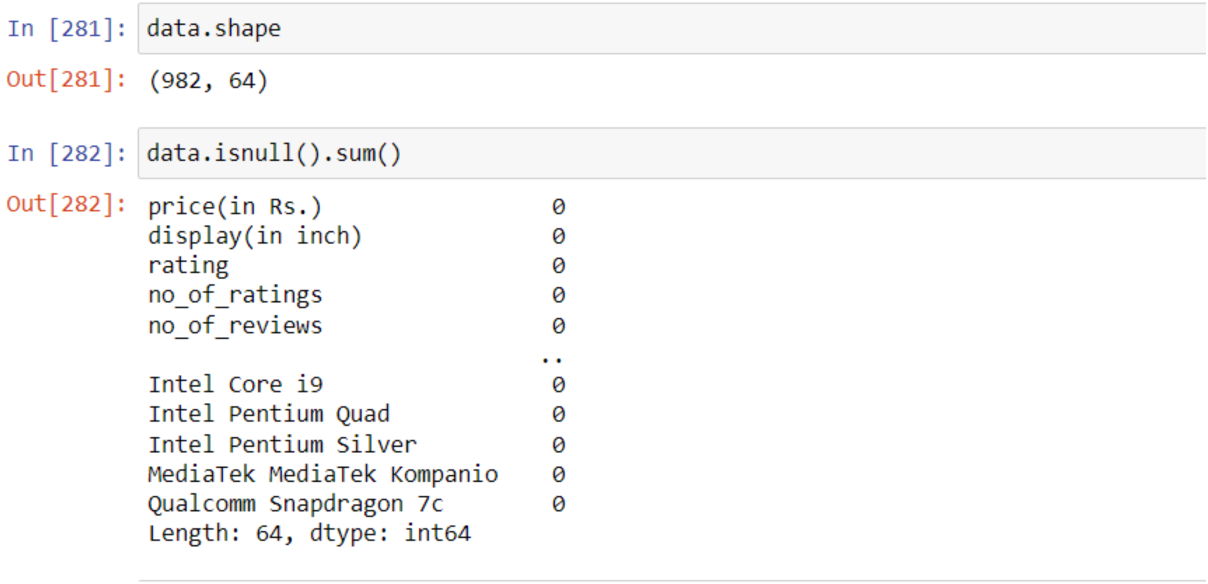


```
In [152]: data.shape

Out[152]: (984, 12)

In [153]: data.isnull().sum()

Out[153]: Unnamed: 0             0
          img_link               0
          name                   0
          price(in Rs.)          0
          processor              0
          ram                    0
          os                     0
          storage                0
          display(in inch)       0
          rating               296
          no_of_ratings        296
          no_of_reviews        296
          dtype: int64
```

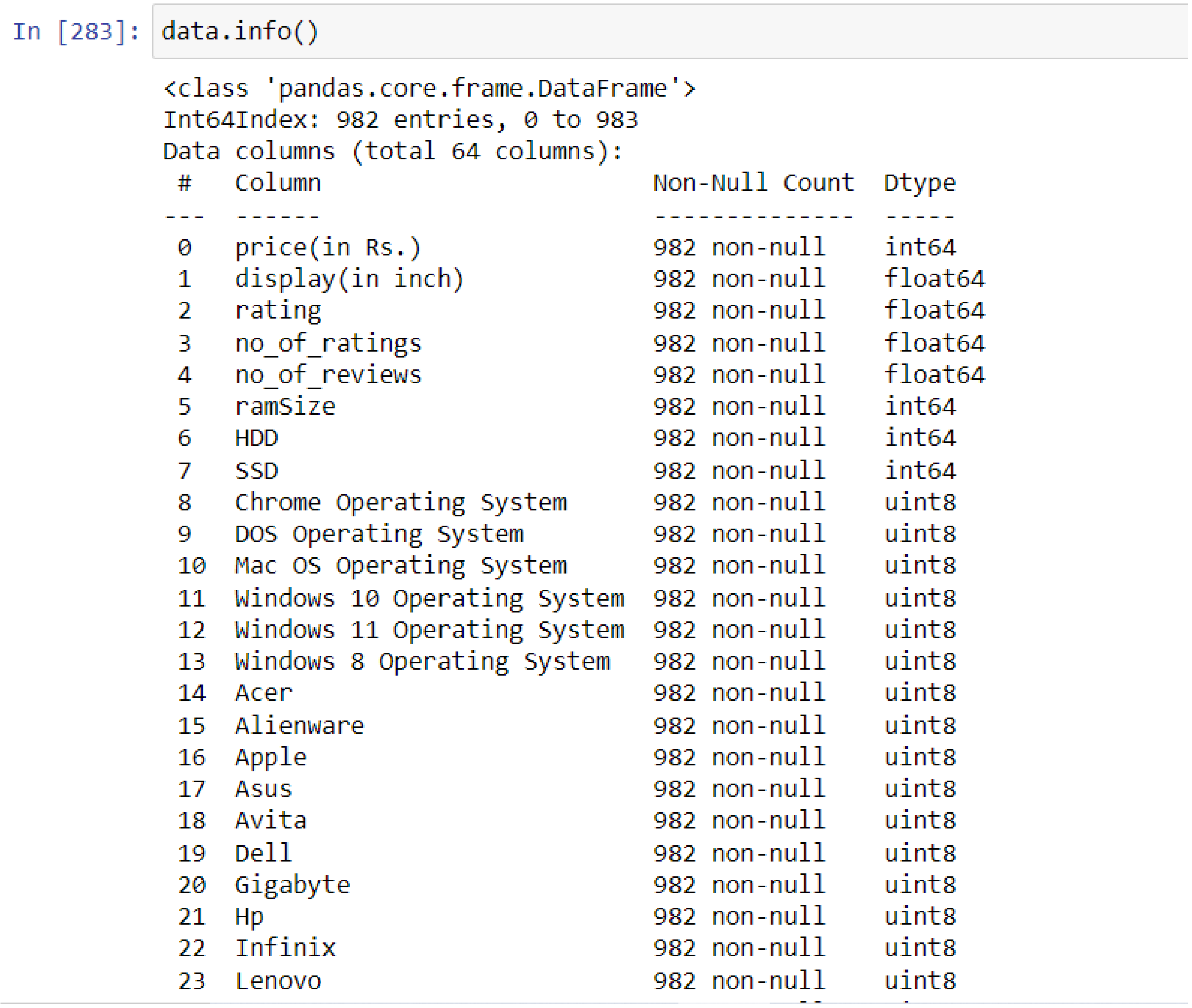


```
In [154]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 984 entries, 0 to 983
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   Unnamed: 0        984 non-null    int64  
 1   img_link          984 non-null    object 
 2   name              984 non-null    object 
 3   price(in Rs.)     984 non-null    int64  
 4   processor         984 non-null    object 
 5   ram               984 non-null    object 
 6   os                984 non-null    object 
 7   storage           984 non-null    object 
 8   display(in inch)  984 non-null    float64
 9   rating            688 non-null    float64
 10  no_of_ratings     688 non-null    float64
 11  no_of_reviews     688 non-null    float64
dtypes: float64(4), int64(2), object(6)
memory usage: 92.4+ KB
```

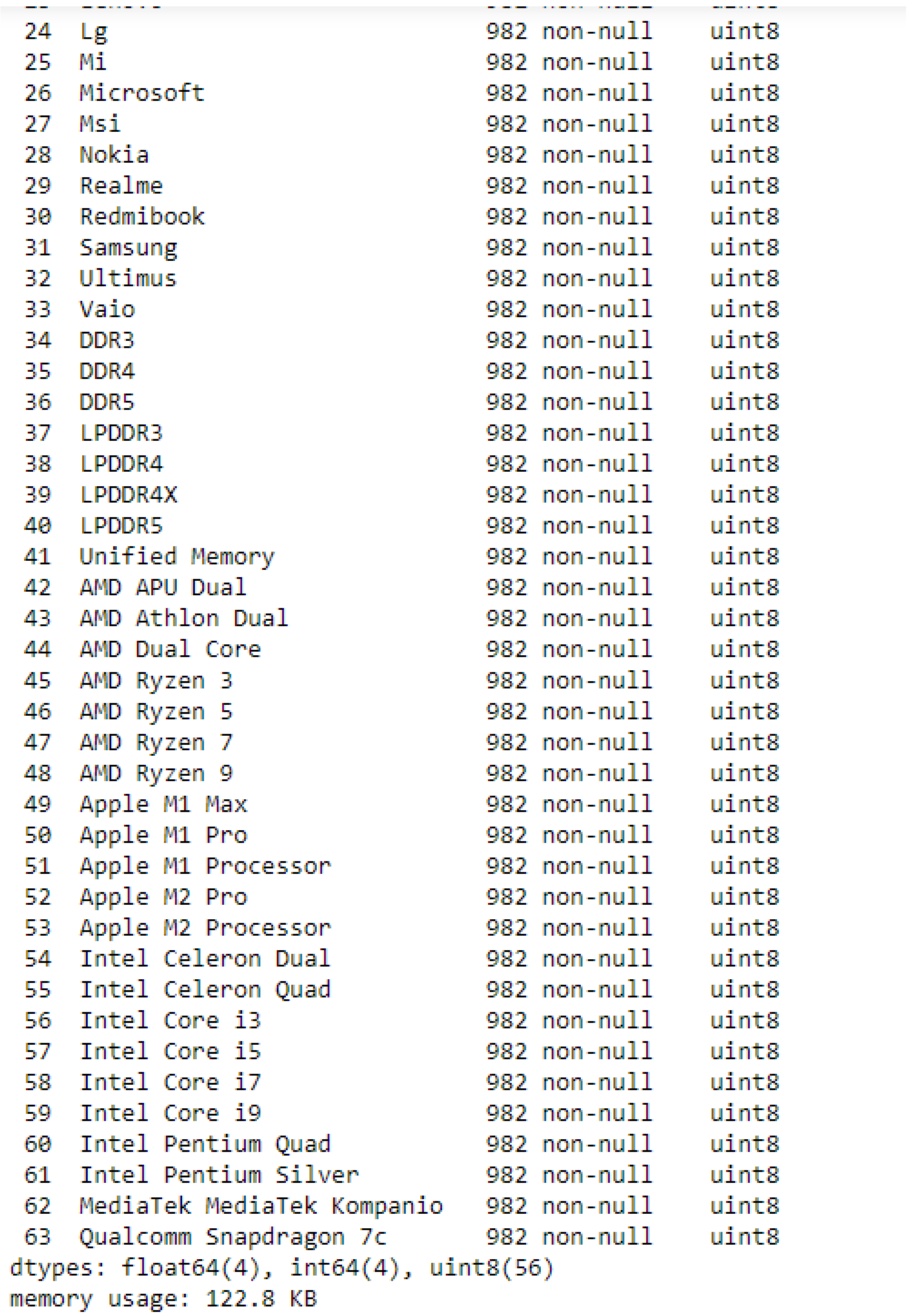
After Preprocessing



```
In [281]: data.shape

Out[281]: (982, 64)

In [282]: data.isnull().sum()

Out[282]: price(in Rs.)                0
          display(in inch)             0
          rating                       0
          no_of_ratings                0
          no_of_reviews                0
                                      ..
          Intel Core i9                0
          Intel Pentium Quad           0
          Intel Pentium Silver         0
          MediaTek MediaTek Kompanio   0
          Qualcomm Snapdragon 7c       0
          Length: 64, dtype: int64
```



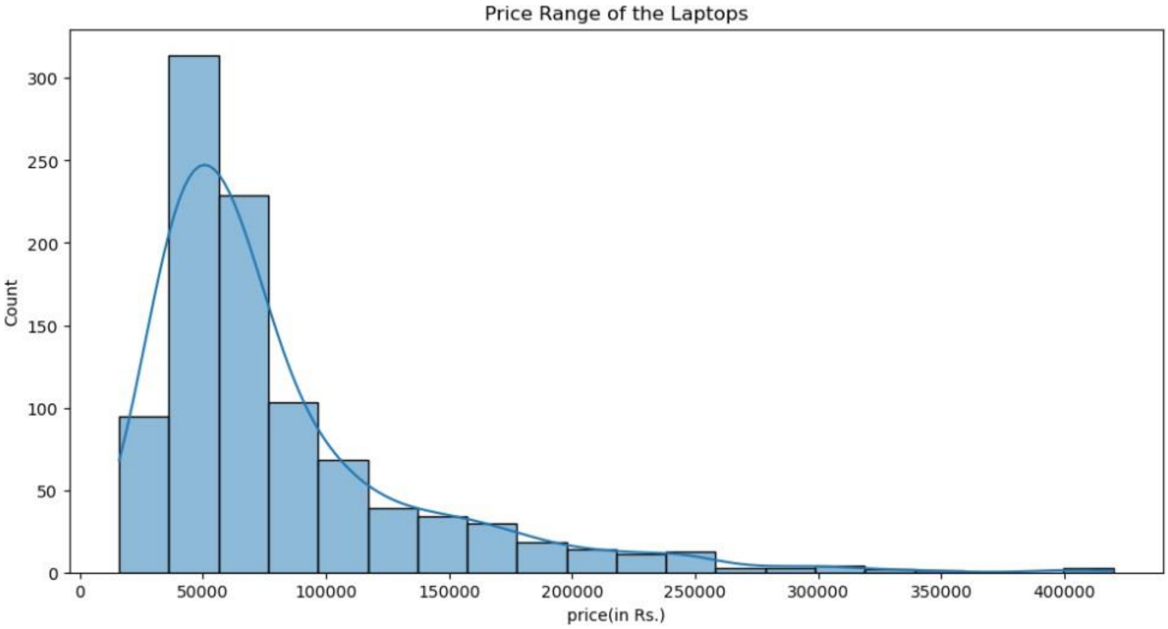
```
In [283]: data.info()

          <class 'pandas.core.frame.DataFrame'>
          Int64Index: 982 entries, 0 to 983
          Data columns (total 64 columns):
           #   Column                       Non-Null Count  Dtype
          ---  ------                       --------------  -----
           0   price(in Rs.)                982 non-null    int64
           1   display(in inch)             982 non-null    float64
           2   rating                       982 non-null    float64
           3   no_of_ratings                982 non-null    float64
           4   no_of_reviews                982 non-null    float64
           5   ramSize                      982 non-null    int64
           6   HDD                          982 non-null    int64
           7   SSD                          982 non-null    int64
           8   Chrome Operating System      982 non-null    uint8
           9   DOS Operating System         982 non-null    uint8
           10  Mac OS Operating System      982 non-null    uint8
           11  Windows 10 Operating System  982 non-null    uint8
           12  Windows 11 Operating System  982 non-null    uint8
           13  Windows 8 Operating System   982 non-null    uint8
           14  Acer                         982 non-null    uint8
           15  Alienware                    982 non-null    uint8
           16  Apple                        982 non-null    uint8
           17  Asus                         982 non-null    uint8
           18  Avita                        982 non-null    uint8
           19  Dell                         982 non-null    uint8
           20  Gigabyte                     982 non-null    uint8
           21  Hp                           982 non-null    uint8
           22  Infinix                      982 non-null    uint8
           23  Lenovo                       982 non-null    uint8
```



```
 24  Lg                         982 non-null    uint8
 25  Mi                         982 non-null    uint8
 26  Microsoft                  982 non-null    uint8
 27  Msi                        982 non-null    uint8
 28  Nokia                      982 non-null    uint8
 29  Realme                     982 non-null    uint8
 30  Redmibook                  982 non-null    uint8
 31  Samsung                    982 non-null    uint8
 32  Ultimus                    982 non-null    uint8
 33  Vaio                       982 non-null    uint8
 34  DDR3                       982 non-null    uint8
 35  DDR4                       982 non-null    uint8
 36  DDR5                       982 non-null    uint8
 37  LPDDR3                     982 non-null    uint8
 38  LPDDR4                     982 non-null    uint8
 39  LPDDR4X                    982 non-null    uint8
 40  LPDDR5                     982 non-null    uint8
 41  Unified Memory             982 non-null    uint8
 42  AMD APU Dual               982 non-null    uint8
 43  AMD Athlon Dual            982 non-null    uint8
 44  AMD Dual Core              982 non-null    uint8
 45  AMD Ryzen 3                982 non-null    uint8
 46  AMD Ryzen 5                982 non-null    uint8
 47  AMD Ryzen 7                982 non-null    uint8
 48  AMD Ryzen 9                982 non-null    uint8
 49  Apple M1 Max               982 non-null    uint8
 50  Apple M1 Pro               982 non-null    uint8
 51  Apple M1 Processor         982 non-null    uint8
 52  Apple M2 Pro               982 non-null    uint8
 53  Apple M2 Processor         982 non-null    uint8
 54  Intel Celeron Dual         982 non-null    uint8
 55  Intel Celeron Quad         982 non-null    uint8
 56  Intel Core i3              982 non-null    uint8
 57  Intel Core i5              982 non-null    uint8
 58  Intel Core i7              982 non-null    uint8
 59  Intel Core i9              982 non-null    uint8
 60  Intel Pentium Quad         982 non-null    uint8
 61  Intel Pentium Silver       982 non-null    uint8
 62  MediaTek MediaTek Kompanio 982 non-null    uint8
 63  Qualcomm Snapdragon 7c     982 non-null    uint8
dtypes: float64(4), int64(4), uint8(56)
memory usage: 122.8 KB
```

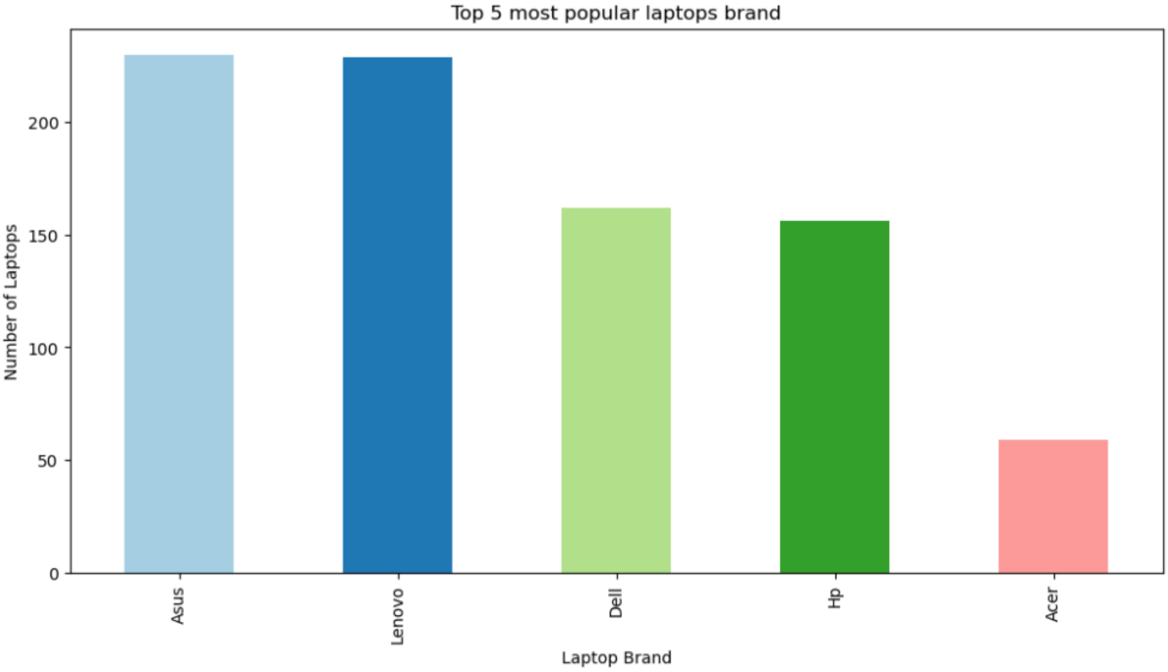
EDA

Price Range



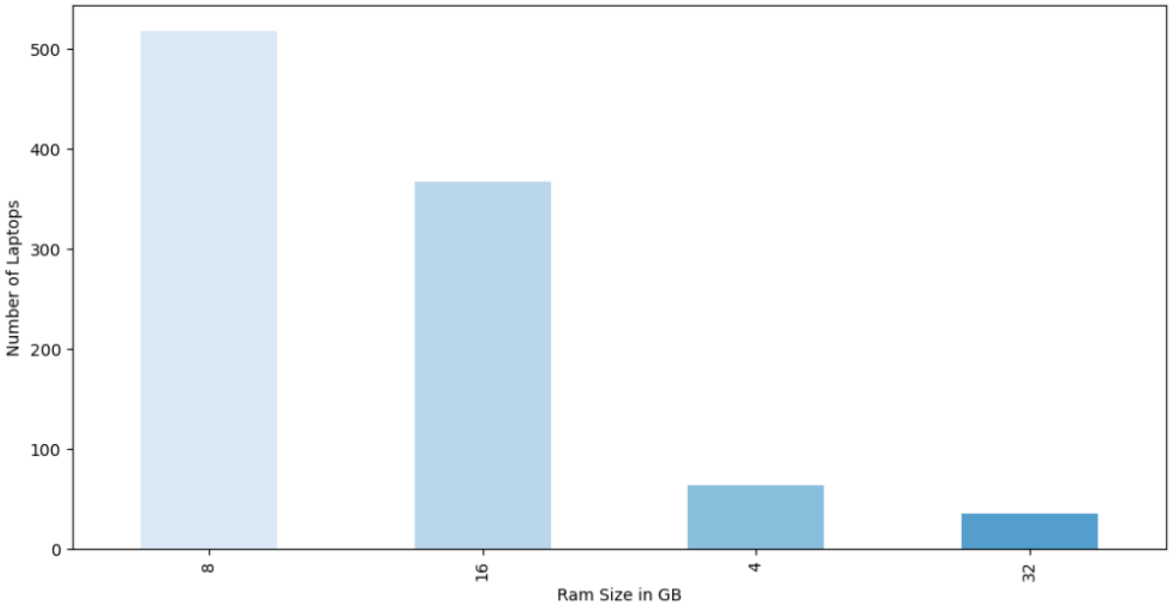
The price range of the laptops ranges from 15,990 to 4,19,990 with the moximum frequency of laptops lies between 20,000 to 75,000.

Popular Brands



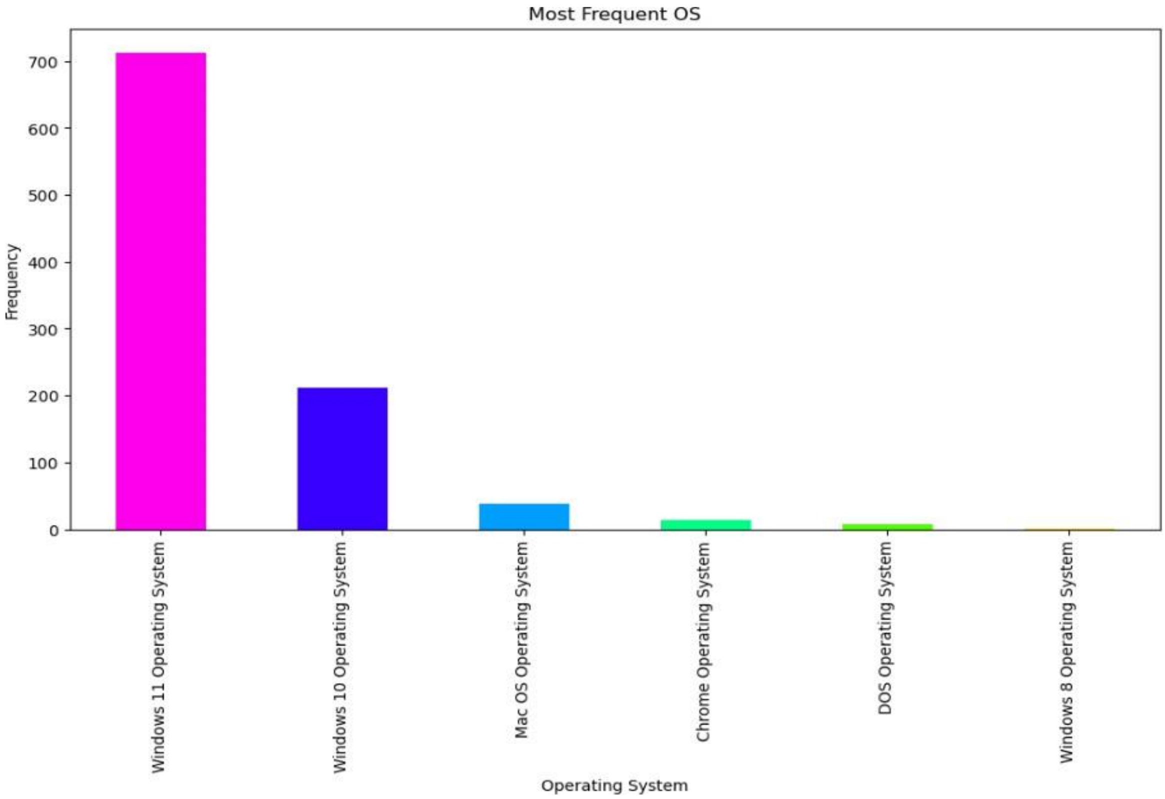
Asus, Lenovo, Dell, HP, Acer are the top five Popular Brand

RAM



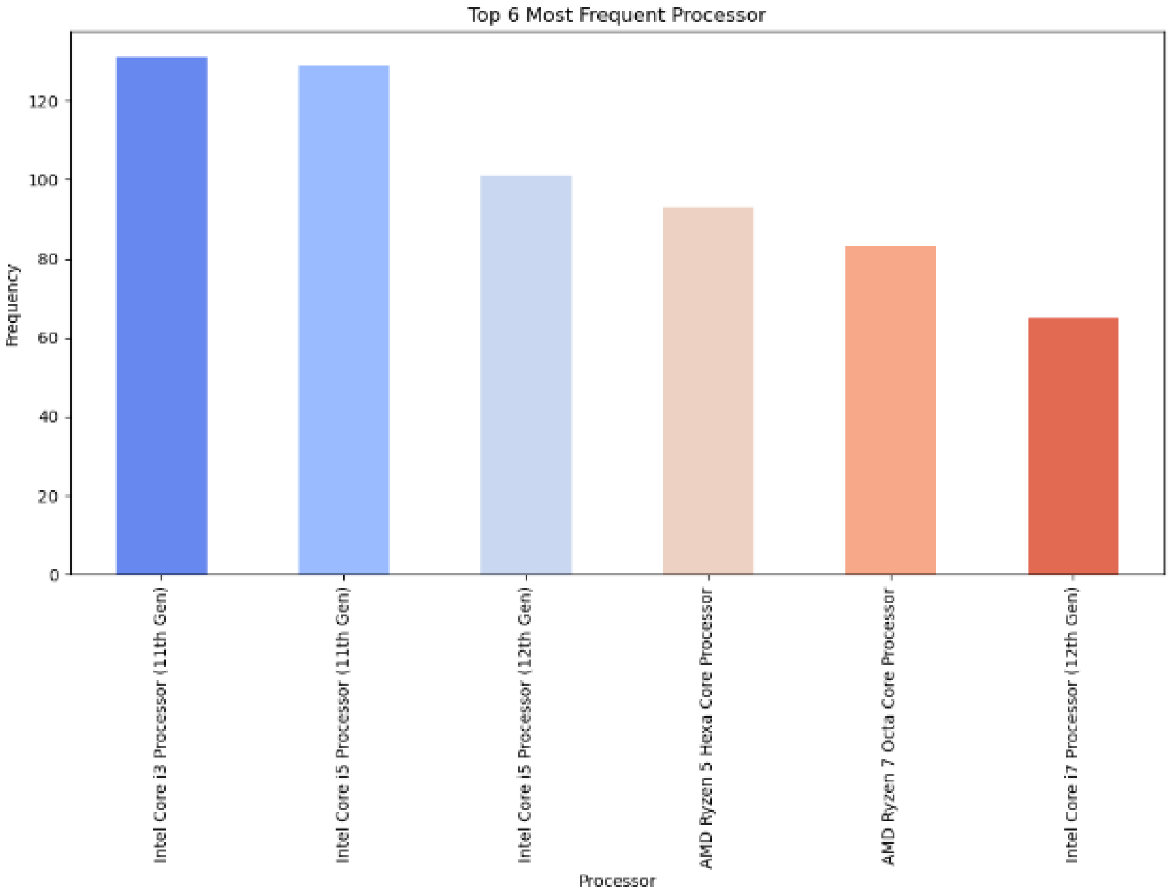
Ram size ranges from 4 to 32 GB. As the cost of most of the laptops lies between Rs 45,000 to Rs 75,000 the ram would be either 8GB or 16GB.

Operating System

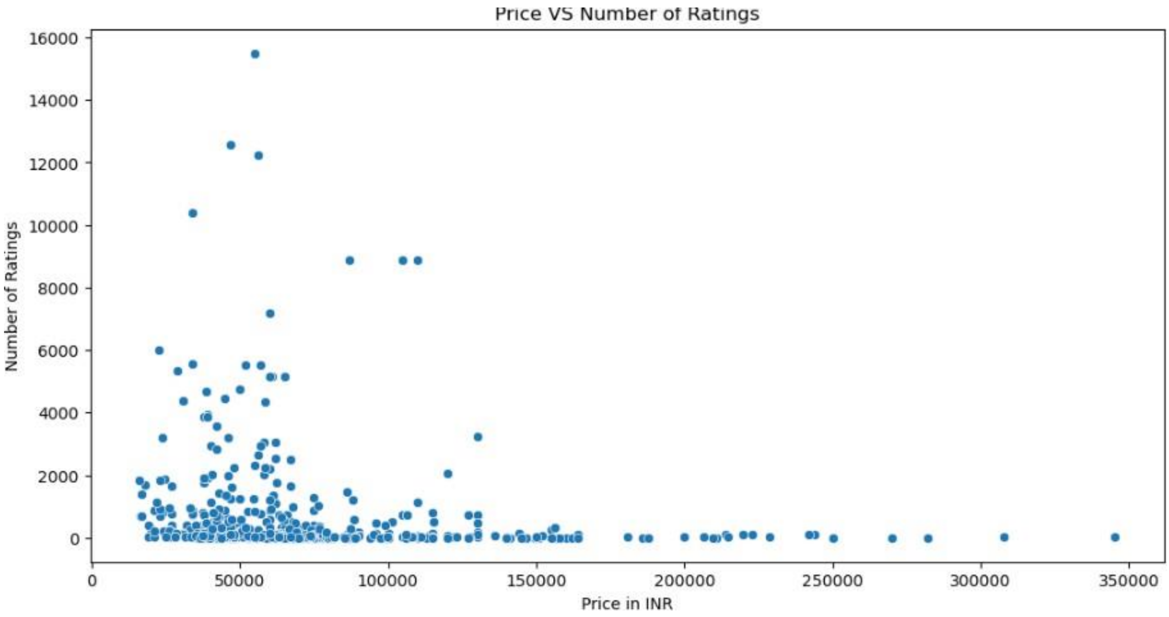


Windows 11 operating system is most frequently used operating system compared to others.

Processor



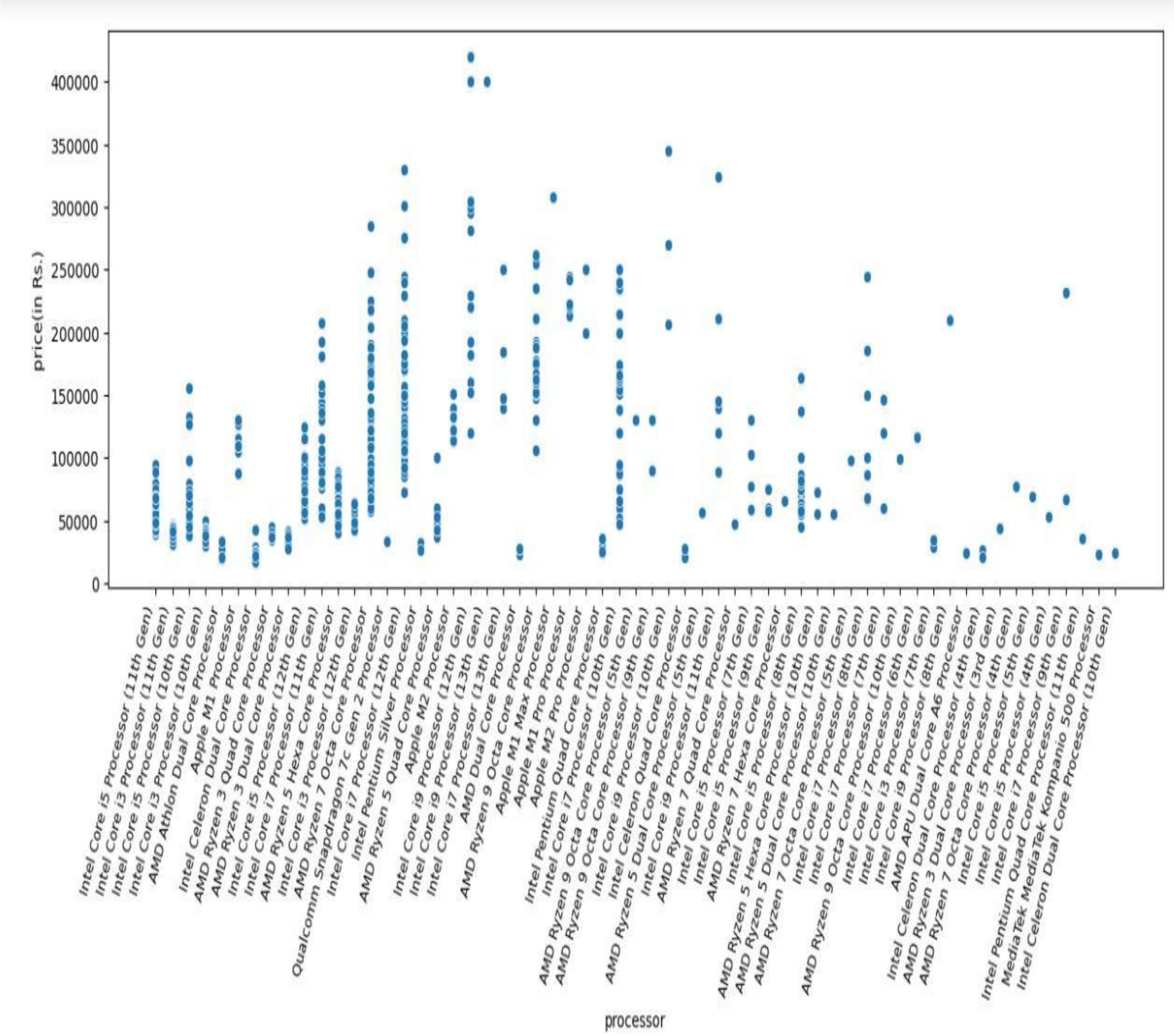
Relation between rating and pricing



When the price is between 25,000 to 1,00,000 there are more number of rating which means that this is the price range in which most people are interested upon

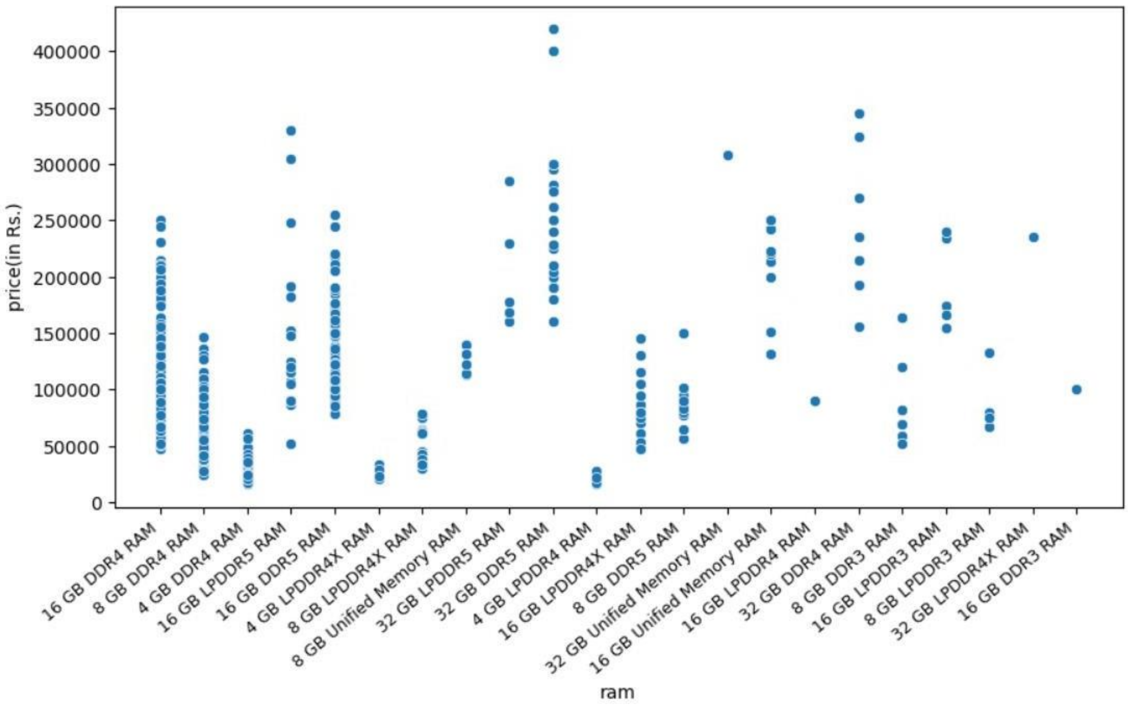
Price dependency

Processor



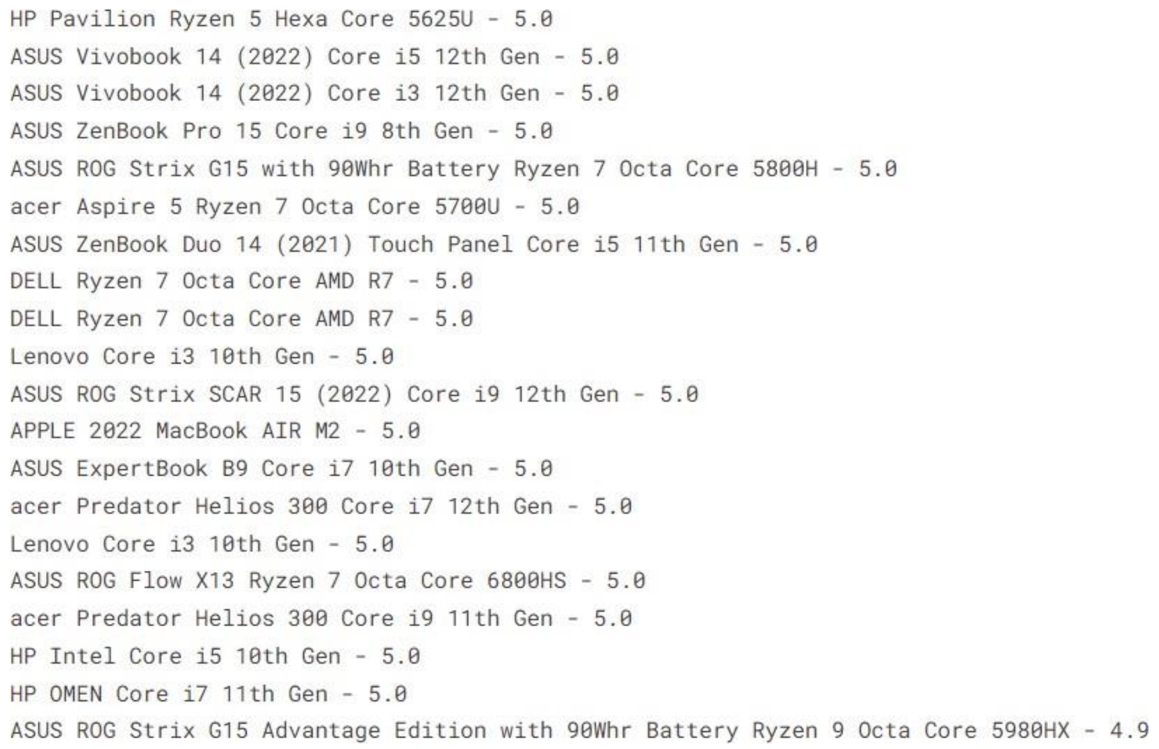
I think the price is most influenced by the processor that is installed in the laptop. Based on the above graph, we can conclude that the price is indeed to some degree based on the type of processor, in addition, there are some processors that are not often found in laptops.

RAM



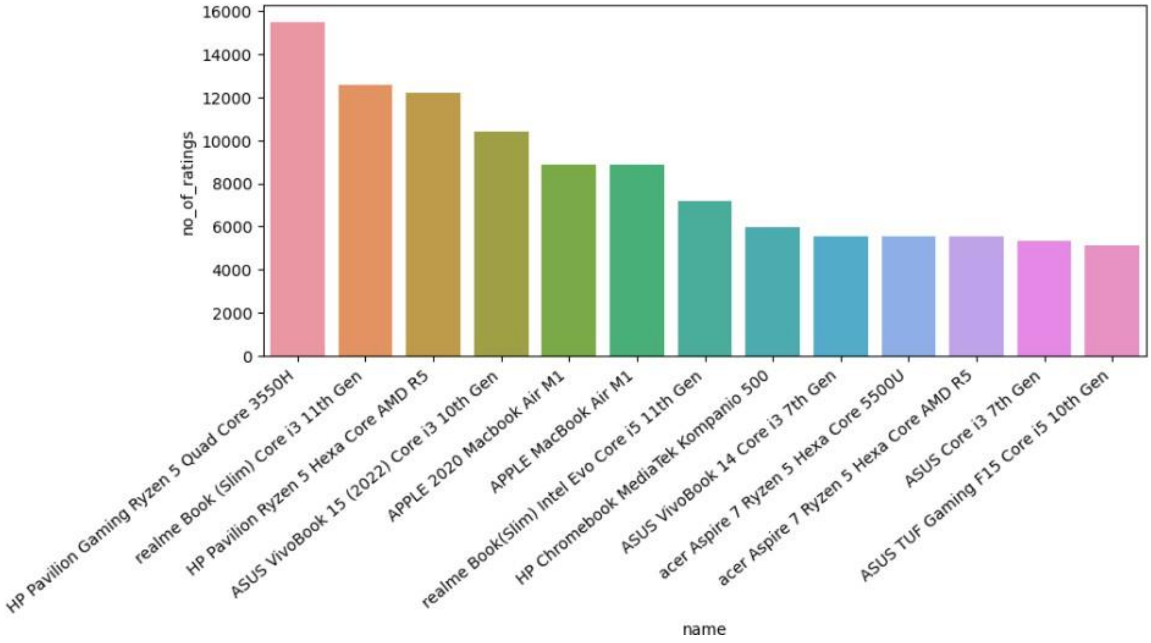
The price depends on the amount of memory, more memory - more expensive laptop.

Best Rating

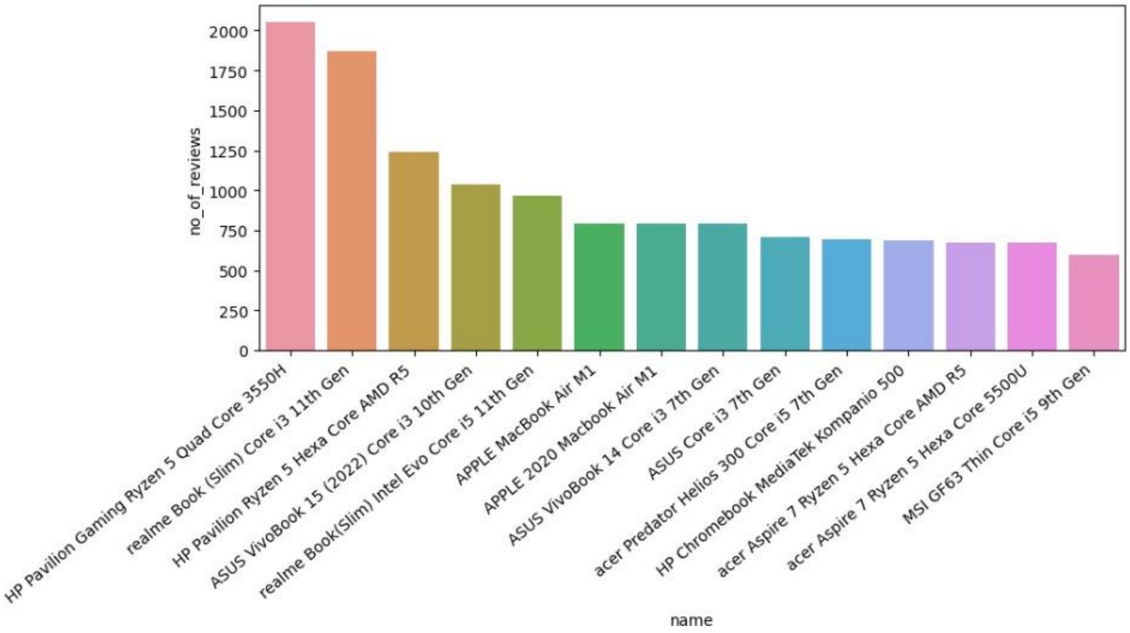


```
HP Pavilion Ryzen 5 Hexa Core 5625U - 5.0
ASUS Vivobook 14 (2022) Core i5 12th Gen - 5.0
ASUS Vivobook 14 (2022) Core i3 12th Gen - 5.0
ASUS ZenBook Pro 15 Core i9 8th Gen - 5.0
ASUS ROG Strix G15 with 90Whr Battery Ryzen 7 Octa Core 5800H - 5.0
acer Aspire 5 Ryzen 7 Octa Core 5700U - 5.0
ASUS ZenBook Duo 14 (2021) Touch Panel Core i5 11th Gen - 5.0
DELL Ryzen 7 Octa Core AMD R7 - 5.0
DELL Ryzen 7 Octa Core AMD R7 - 5.0
Lenovo Core i3 10th Gen - 5.0
ASUS ROG Strix SCAR 15 (2022) Core i9 12th Gen - 5.0
APPLE 2022 MacBook AIR M2 - 5.0
ASUS ExpertBook B9 Core i7 10th Gen - 5.0
acer Predator Helios 300 Core i7 12th Gen - 5.0
Lenovo Core i3 10th Gen - 5.0
ASUS ROG Flow X13 Ryzen 7 Octa Core 6800HS - 5.0
acer Predator Helios 300 Core i9 11th Gen - 5.0
HP Intel Core i5 10th Gen - 5.0
HP OMEN Core i7 11th Gen - 5.0
ASUS ROG Strix G15 Advantage Edition with 90Whr Battery Ryzen 9 Octa Core 5980HX - 4.9
```

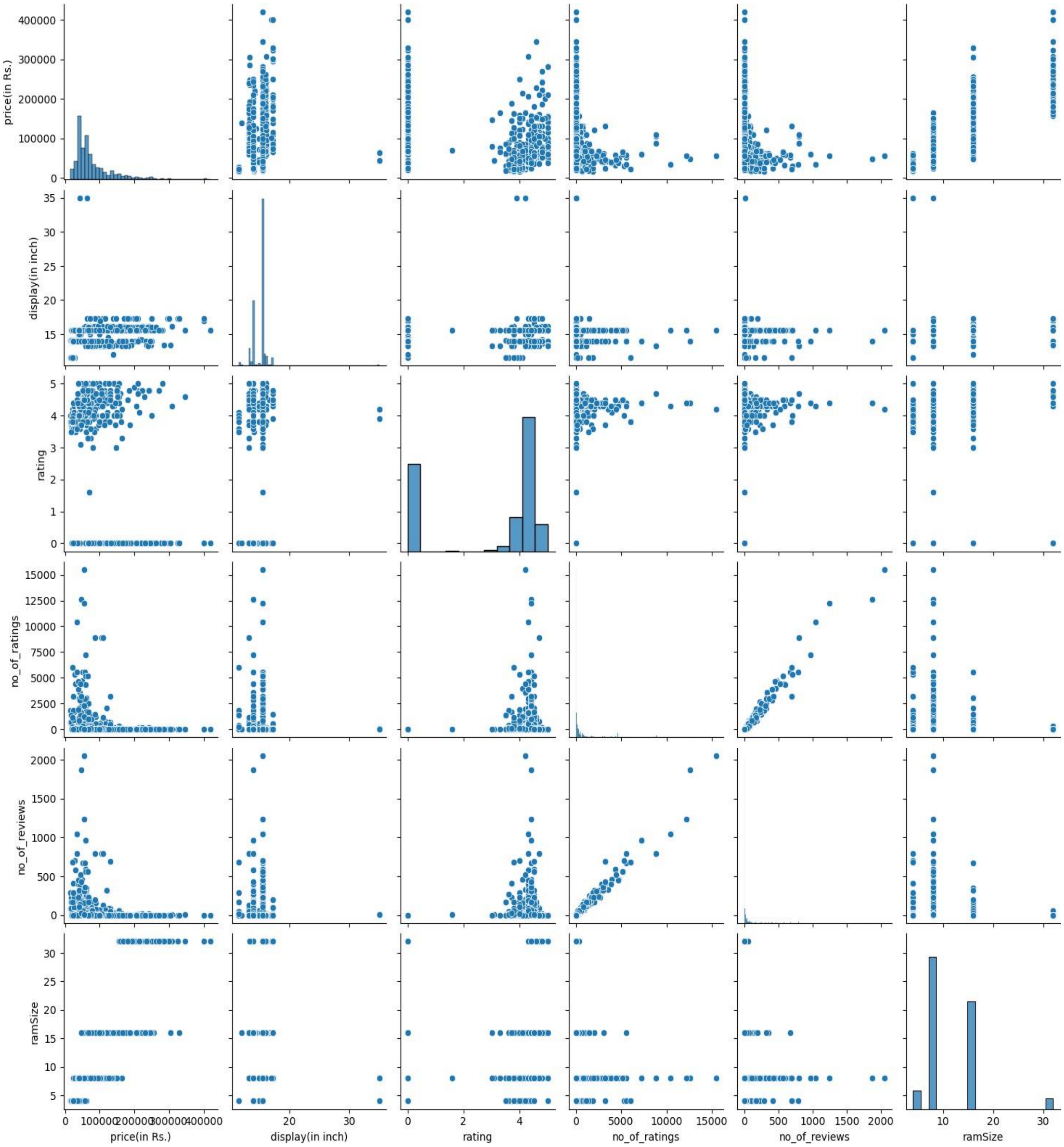
Most Rated



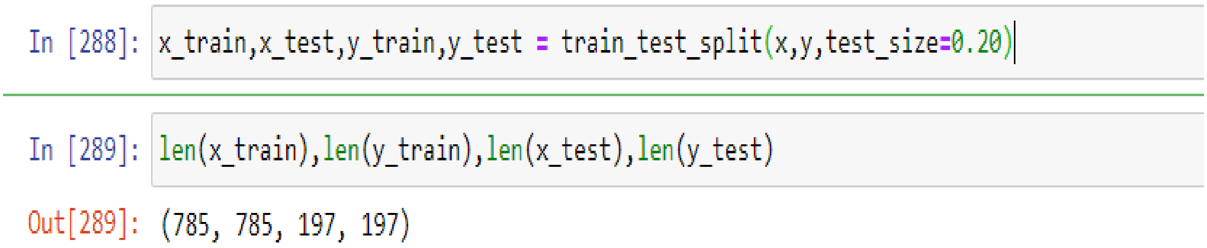
Most Reviews



Dependency of Numerical features on each other

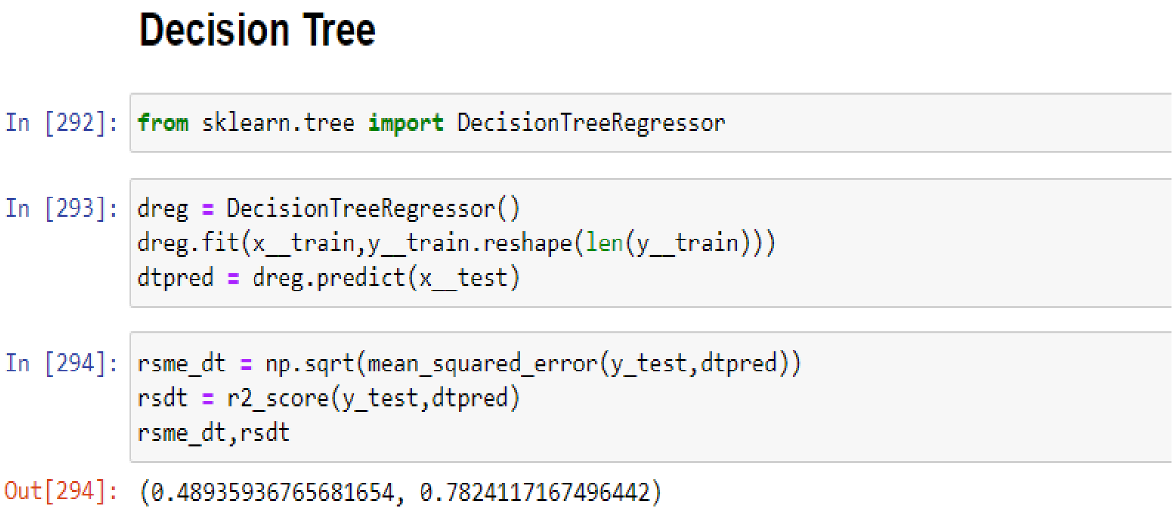


Train and Test Dataset



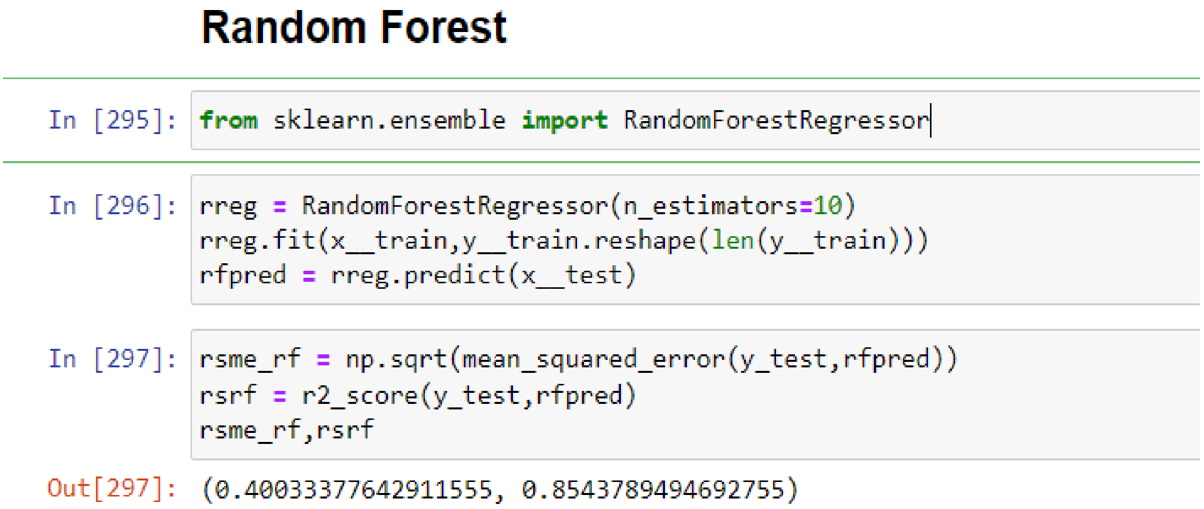
```
In [288]: x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.20)

In [289]: len(x_train),len(y_train),len(x_test),len(y_test)

Out[289]: (785, 785, 197, 197)
```

RSME and R2score



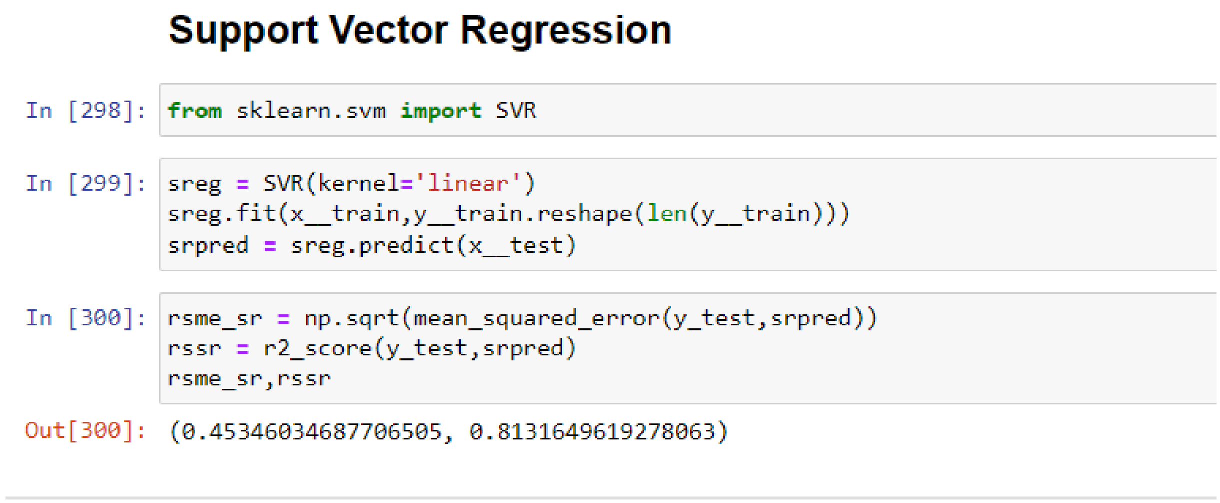
Decision Tree

```
In [292]: from sklearn.tree import DecisionTreeRegressor

In [293]: dreg = DecisionTreeRegressor()
          dreg.fit(x__train,y__train.reshape(len(y__train)))
          dtpred = dreg.predict(x__test)

In [294]: rsme_dt = np.sqrt(mean_squared_error(y_test,dtpred))
          rsdt = r2_score(y_test,dtpred)
          rsme_dt,rsdt

Out[294]: (0.48935936765681654, 0.7824117167496442)
```



Random Forest

```
In [295]: from sklearn.ensemble import RandomForestRegressor

In [296]: rreg = RandomForestRegressor(n_estimators=10)
          rreg.fit(x__train,y__train.reshape(len(y__train)))
          rfpred = rreg.predict(x__test)

In [297]: rsme_rf = np.sqrt(mean_squared_error(y_test,rfpred))
          rsrf = r2_score(y_test,rfpred)
          rsme_rf,rsrf

Out[297]: (0.40033377642911555, 0.8543789494692755)
```



Support Vector Regression

```
In [298]: from sklearn.svm import SVR

In [299]: sreg = SVR(kernel='linear')
          sreg.fit(x__train,y__train.reshape(len(y__train)))
          srpred = sreg.predict(x__test)

In [300]: rsme_sr = np.sqrt(mean_squared_error(y_test,srpred))
          rssr = r2_score(y_test,srpred)
          rsme_sr,rssr

Out[300]: (0.45346034687706505, 0.8131649619278063)
```

From comparison we can see that Random forest algorithm has the highest r2score and lowest root square mean error. So, the random forest algorithm is the best to predict the price of the laptops.

CHAPTER 5

DECLARATION

I, Amrutha R. a student of 7th semester BE, Computer Science and Engineering department, Bangalore Institute of Technology , Bengaluru hereby declare that internship project work entitled "LAPTOP SELECTION ANALYSIS" has been carried out by me at Prinston Smart Engineers,Bengaluru and submitted in partial fulfillment of the course requirement for the award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi, during the academic year 2022-2023.

I also declare that, to the best of my knowledge and belief, the work reported here is not from the part of dissertation on the basis of which a degree or award was conferred on an earlier occasion on this by any other student.

Place: Bengaluru

Amrutha R.
[1BI19CS020]

CHAPTER 6

CONCLUSION/FUTURE ENHANCEMENT

As laptops become an essential tool for work, education, and entertainment, it can be challenging for consumers to choose the right laptop for their needs. With the proposed machine learning model one can predict the price of the model based on the specifications.

The r2 score of decision tree, random forest, and support vector regression is 0.7824, 0.8543, and 0.8131 respectively. From this we can conclude that the random forest algorithm predicts more precisely compared to other algorithms.

I have found the important features which could play a vital role in laptop selection and non influential features as well. I studied the report of prediction carefully. I can expand the existing system with additional analysis methods and implementation with neural networks and deep learning.

CHAPTER 7

REFERENCES

- https://www.kaggle.com/datasets/rajugc/laptop-selection-dataset
- https://pandas.pydata.org/docs/
- https://matplotlib.org/stable/index.html
- https://www.academia.edu/69591584/Laptop_Price_Prediction_using_Machine_Learning