

GadgetHive: Price Prediction and Recommendation

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
OF MINI PROJECT

SEMESTER VI

IN

INFORMATION TECHNOLOGY

BY

Ansari Mohmmad Waqiur Rahman

Rahul Chalke

Vanraj Pardeshi

Anurag Singh

UNDER THE GUIDANCE OF

Ms.Chhaya Dhavale

(Assistant Professor, Department of Information Technology)



INFORMATION TECHNOLOGY

DEPARTMENT XAVIER INSTITUTE OF

ENGINEERING UNIVERSITY OF MUMBAI

2022-2023

XAVIER INSTITUTE OF ENGINEERING

MAHIM CAUSEWAY, MAHIM, MUMBAI - 400016.

CERTIFICATE

This to certify that

Ansari Mohmmad Waqiur Rahman	(202003019)
Rahul Chalke	(201903007)
Vanraj Pardeshi	(202003030)
Anurag Singh	(202003055)

Have satisfactorily carried out the PROJECT work titled “GadgetHive: Price Prediction and Recommendation” in partial fulfilment of the Mini project of Sem-6 of Information Technology as laid down by the University of Mumbai during the academic year 2022-2023.

Prof. Meena Ugale
Head of Department

Prof.Chhaya Dhavale
Supervisor/Guide

Date:

Place: Mahim, Mumbai

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' Ideas or words have been included, I have adequately cited and referenced the original sources.

I also declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which thus have not been properly cited or from whom proper permission have not been taken when needed.

Ansari Mohmmad Waqiur Rahman (202003019)

Rahul Chalke (201903007)

Vanraj Pardeshi (202003030)

Anurag Singh (202003055)

Date:

ABSTRACT

The objective of GadgetHive is to provide users with accurate predictions of laptop prices and recommendations of smartphones based on various features such as RAM, processor, and other specifications. By utilizing advanced algorithms such as cosine similarity and random forest, the application aims to provide reliable and precise results to help users make informed decisions when purchasing laptops and smartphones. Additionally, GadgetHive seeks to simplify the process of selecting the right laptop or smartphone by streamlining the selection process and presenting users with a concise list of options that best match their preferences and requirements. The ultimate goal of GadgetHive is to enhance the user's experience by providing them with a personalized and efficient way of choosing the right device for their needs.

The project aims to answer questions such as:

- What is the price range of a specific laptop model based on its features?
- Which laptop models have the highest RAM and processor speed?
- Which smartphones are recommended based on specific features, such as camera quality, screen size, or battery life?
- What is the best smartphone match for a user's preferred specifications?

Overall, the results of the analysis provided by GadgetHive are expected to be reliable and useful for anyone looking to buy a laptop or a smartphone. By simplifying the selection process and providing personalized recommendations and predictions, GadgetHive helps users make informed decisions and choose the device that best fits their needs and budget.

ACKNOWLEDGEMENT

We would like to thank Fr. (Dr.) John Rose SJ (Director of XIE) for providing us with such an environment so as to achieve goals of our project and supporting us constantly.

We express our sincere gratitude to our Honourable Principal Dr. Y.D. Venkatesh for encouragement and facilities provided to us.

We would like to place on record our deep sense of gratitude to Prof. Meena Ugale, Head of Dept of Information Technology, Xavier Institute of Engineering, Mahim, Mumbai, for her generous guidance, help and useful suggestions.

With deep sense of gratitude, we acknowledge the guidance of our project guide Prof.Chhaya Dhavale. The time-to-time assistance and encouragement by him has played an important role in the development of our project.

We would also like to thank our entire Information Technology staff who have willingly cooperated with us in resolving our queries and providing us all the required facilities on time.

Ansari Mohmmad Waqiur Rahman (202003019) -----

Rahul Chalke (201903007) -----

Vanraj Pardeshi (202003030) -----

Anurag Singh (202003055) -----

XAVIER INSTITUTE OF ENGINEERING



Institute Vision

To nurture the joy of excellence in a world of high technology.

Institute Mission

To strive to match global standards in technical education by interaction with industry, continuous staff training and development of quality of life.



Xavier Institute of Engineering
Mahim, Mumbai 400016

Department of Information Technology

(Affiliated to University of Mumbai)

Department Vision

To nurture the joy of excellence in the world of Information Technology.

Department Mission

- M1: To develop the critical thinking ability of students by promoting interactive learning.
- M2: To bridge the gap between industry and institute and give students the kind of exposure to the industrial requirements in current trends of developing technology.
- M3: To promote learning and research methods and make them excel in the field of their study by becoming responsible while dealing with social concerns.
- M4: To encourage students to pursue higher studies and provide them awareness on various career opportunities that are available.



Xavier Institute of Engineering

Mahim, Mumbai 400016

Department of Information Technology

(Affiliated to University of Mumbai)

Program Educational Objectives (PEOs)

Information Technology Engineering Graduates shall be

PEO1: Employed as IT professionals, and shall engage themselves in learning, understanding, and applying newly developed ideas and technologies as their field of study evolves.

PEO2: Competent to use the learnt knowledge successfully in the diversified sectors of industry, academia, research and work effectively in multidisciplinary environments.

PEO3: Aware of professional ethics and create a sense of social responsibility in building the nation/society.

Program Specific Outcomes (PSOs)

PSO1: Demonstrate the ability to analyze and visualize the business domain and formulate appropriate information technology solutions.

PSO2: Apply various technologies like Intelligent Systems, Data Mining, IOT, Cloud and Analytics, Computer and Network Security etc. for innovative solutions to real time problems.

Program Outcomes (POs)

Engineering Graduates will be able to

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions for complex problems.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Table of Content

Sr. No	Topic	Page No
A	Abstract	4
B	Acknowledgment	5
C	Department LOs, POs and PSOs	7
D	List of Images	12
1	Introduction 1. Problem Definition 2. Scope and importance of project	14
2	Hardware and Software requirements	15
3	Analysis and Planning	16
4	Implementation 1. Implementation Process 2. Results and Test Cases	17
5	Conclusion and Future Work	32
6	References	33

List of Images

Sr. No	Title	Page No
1	Concept Diagram	16
2	Data Preprocessing	17
3	Importing Libraries	18
4	Loading CSV File	19
5	Data.shape	19
6	Data.info()	20
7	Data Frame Profiling	20
8	Data.isnull().sum()	21
9	Checking app with rating more than 5	21
10	Shifting wrong data	22
11	Splitting the Last Updated Column	22
12	Data Visualization	23
13	Histogram	23
14	Scatter Plot	24
15	Bar Graph	24
16	Linear Regression	26
17	Linear Regression Accuracy	26
18	Creating Decision Tree	27
19	Decision Tree Accuracy	27
20	Creating SVM Regressor	28
21	Multiple Decision Boundaries	28
22	SVM Accuracy	28

23	Random Forest Algorithm	29
24	Creating Random Forest Algorithm	30
25	Random Forest Accuracy	30

1. INTRODUCTION

- **Problem Definition**

GadgetHive aims to solve the problem of users facing difficulties in selecting the right laptop or smartphone based on their requirements, preferences, and budget. With a vast number of options available in the market, users may feel overwhelmed and unsure of what features to prioritize, leading to confusion and frustration. GadgetHive uses advanced algorithms to provide accurate predictions of laptop prices and personalized recommendations of smartphones based on user preferences, simplifying the selection process and enabling users to make informed decisions.

- **Scope and importance of the project**

Scope:

- GadgetHive provides laptop price prediction and smartphone recommendation services based on various features and specifications.
- The application utilizes advanced algorithms such as cosine similarity and random forest to provide accurate predictions and recommendations.
- The platform can assist users in selecting the best device that fits their requirements and budget, simplifying the selection process.

Importance:

- GadgetHive addresses the problem of users feeling overwhelmed and uncertain when selecting the right laptop or smartphone for their needs, streamlining the selection process and saving time.
- The application provides personalized recommendations and predictions based on user preferences and requirements, ensuring that users make informed decisions when purchasing a device.
- By providing accurate predictions of laptop prices, GadgetHive can also help users save money by suggesting devices that offer the best value for their budget.
- GadgetHive's advanced algorithms and personalized recommendations make it an important tool for anyone looking to buy a laptop or smartphone, particularly those who are unfamiliar with the technical specifications and features of these devices.

2. HARDWARE AND SOFTWARE REQUIREMENTS

1. Hardware Requirements

- Core I3 Processor And Above
- At Least 4gb Ram
- At Least 128 Gb Storage

2. Software Requirements

- Python 3.6.10 Or Above
- Jupyter Lab / Notebook
- Google Colab
- Packages : Numpy , Pandas, Scikit Learn, Matplotlib, Seaborn

3. ANALYSIS AND PLANNING

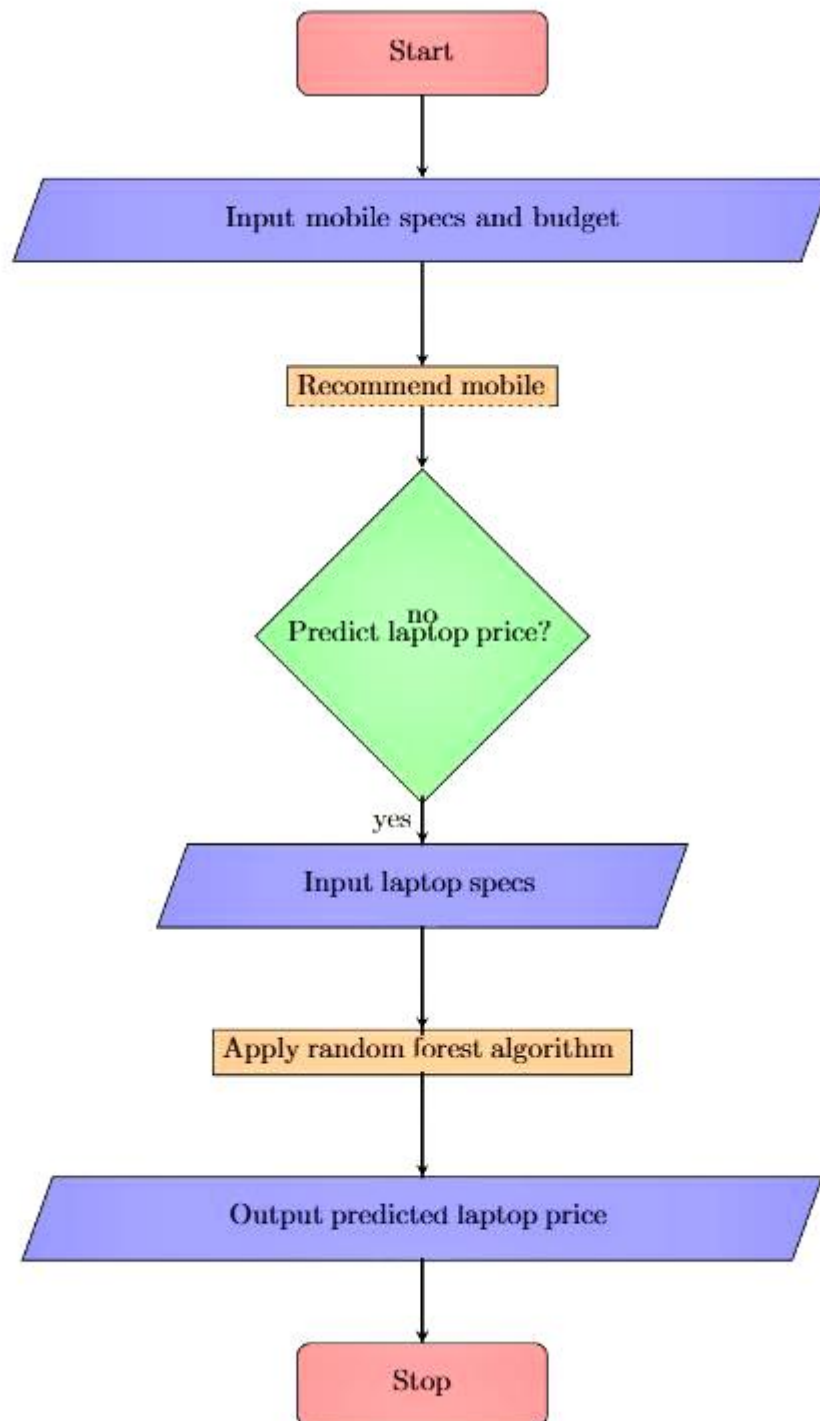


Figure 1 Concept Diagram

4. IMPLEMENTATION METHODOLOGY

Implementation Process

The implementation process we further divide into three phases

1. Data Pre-processing
2. Data Analysis
3. Model Creation
4. Checking Accuracy

Data Pre-processing:

Data pre-processing is a data mining technique which is used to transform the raw data in a useful and efficient format.

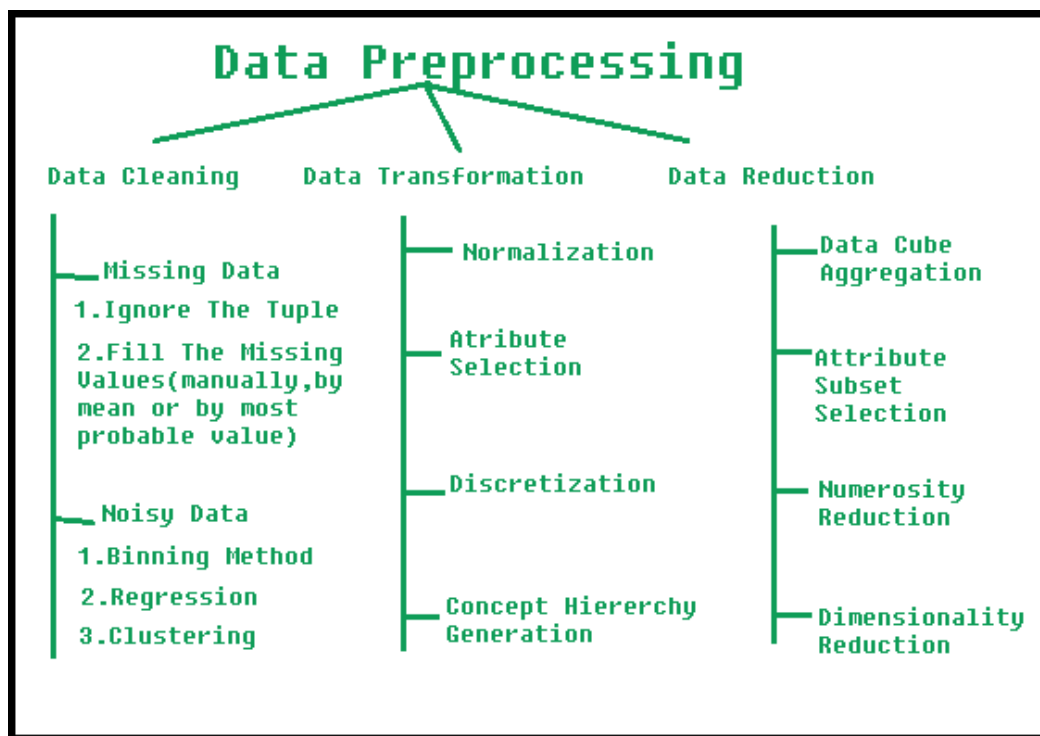


Figure 2 Data Preprocessing

Steps Involved in Data Pre-processing:

1. Data Cleaning:

(a) Missing Data:

This situation arises when some data is missing in the data. It can be handled in various ways. Some of them are: Ignore the tuples: This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

(b) Noisy Data:

Noisy data is meaningless data that can't be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways:

Binning Method: This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segment is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the task.

2. Data Reduction:

Since data mining is a technique that is used to handle huge amounts of data. While working with a huge volume of data, analysis became harder in such cases. In order to get rid of this, we use data reduction techniques. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

- i. Data Cube Aggregation:
- ii. Numerosity Reduction:
- iii. Dimensionality Reduction:

Data pre-processing in Googleplaystore.csv:

First importing all libraries:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # this is used for the plot the graph
import seaborn as sns # used for plot interactive graph.
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
%matplotlib inline
```

Figure 3 Importing Libraries

Loading the csv file:

In [2]:	df = pd.read_csv('laptop_data.csv')											
In [3]:	df.head()											
Out[3]:	Unnamed: 0	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	71378.6832
1	1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	47895.5232
2	2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	30636.0000
3	3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	135195.3360
4	4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	96095.8080

Figure 4 Loading CSV File

Pandas Data Frame data.shape Method

The **data.shape** attribute is a built-in attribute in the Pandas library of Python programming language. It is used to get the dimensions or shape of a Pandas DataFrame, which is represented as a tuple containing the number of rows and columns.

```
In [4]: df.shape
Out[4]: (1303, 12)
```

Figure 5 data.shape

Pandas Data Frame data.info() Method

The data.info () method is a built-in method in the Pandas library of Python programming language. It is used to print a concise summary of a Pandas Data Frame, including information about the column names, data types, non-null values, and memory usage.

```
In [5]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1303 entries, 0 to 1302
Data columns (total 12 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   Unnamed: 0      1303 non-null   int64  
1   Company         1303 non-null   object  
2   TypeName        1303 non-null   object  
3   Inches          1303 non-null   float64 
4   ScreenResolution 1303 non-null   object  
5   Cpu             1303 non-null   object  
6   Ram             1303 non-null   object  
7   Memory          1303 non-null   object  
8   Gpu             1303 non-null   object  
9   OpSys           1303 non-null   object  
10  Weight          1303 non-null   object  
11  Price           1303 non-null   float64 
dtypes: float64(2), int64(1), object(9)
memory usage: 122.3+ KB
```

Figure 6 data.info()

Pandas Data Frame profiling Method

Here we use pandas profiling to analyse whole set of data. Pandas profiling provides analysis like type, unique values, missing values, quantile statistics, mean, mode, median, standard deviation, sum, skewness, frequent values, histograms, correlation between variables, count, heatmap visualization, etc.

```
import pandas_profiling
pandas_profiling.ProfileReport(data)
```

Overview

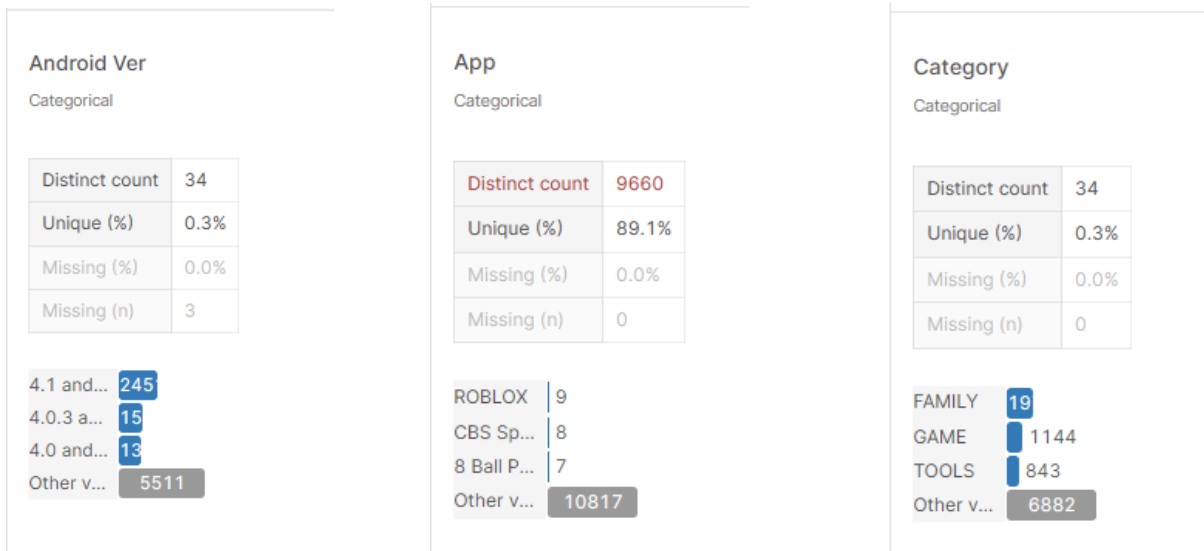
Dataset info

Number of variables	13
Number of observations	10841
Total Missing (%)	1.1%
Total size in memory	1.1 MiB
Average record size in memory	104.0 B

Variables types

Numeric	1
Categorical	12
Boolean	0
Date	0
Text (Unique)	0
Rejected	0
Unsupported	0

Figure 7 Data frame profiling



1. Data analysis

Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.

```
data.isnull().sum()
```

```
App      0
Category 0
Rating   1474
Reviews  0
Size      0
Installs  0
Type      1
Price     0
Content Rating  1
Genres    0
Last Updated  0
Current Ver  8
Android Ver  3
dtype: int64
```

Figure 8 `data.isnull().sum()`

while pre-processing and viewing the data we observe that in this Datasets the maximum rating of an app is 19 which is not possible yet, we know that the maximum rating of an app in the google play store is 5. >>so we confinity says that something wrong.

```
data[data['Rating'] == 19]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Cont Ratin
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	1,000+	Free	0	Everyone	NaN

Figure 9 Checking app with rating more than 5

By viewing this we say that in this row all data are placed wrong by one position so we shift the data to the right hand side and make correct position

```
data.iloc[10472,1:] = data.iloc[10472,1:].shift(1)
data[10471:10473]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price
10471	Xposed Wi-Fi-Pwd	PERSONALIZATION	3.5	1042	404k	100,000+	Free	0
10472	Life Made Wi-Fi Touchscreen Photo Frame	NaN	1.9	19	3.0M	1,000+	Free	0

Figure 10 Shifting wrong data

We see that the app 'Life Made Wi-Fi Touchscreen Photo Frame' does not have any category, when we search this app on play store we found that it belong to lifestyle category Here we added 2 more columns in the data set by splitting the last updated attribute, by doing this we find that in which year apps are added or updated on playstore.

```
data["Last Updated"] = pd.to_datetime(data['Last Updated'])
data['year_added'] = data['Last Updated'].dt.year
data['month_added'] = data['Last Updated'].dt.month
```

```
data.head(2)
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19M	10,000+	Free	0	Everyone
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14M	500,000+	Free	0	Everyone

Figure 11 Splitting the Last Updated Column

Data Visualization:

```
In [14]: df.head()
```

Out[14]:	Company	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price
0	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	71378.6832
1	Apple	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	47895.5232
2	HP	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8	256GB SSD	Intel HD Graphics 620	No OS	1.86	30636.0000
3	Apple	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16	512GB SSD	AMD Radeon Pro 455	macOS	1.83	135195.3360
4	Apple	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	96095.8080

Figure 12 Data Visualization

Histogram:

```
In [28]: df['TypeName'].value_counts().plot(kind='bar')
```

```
Out[28]: <AxesSubplot:>
```

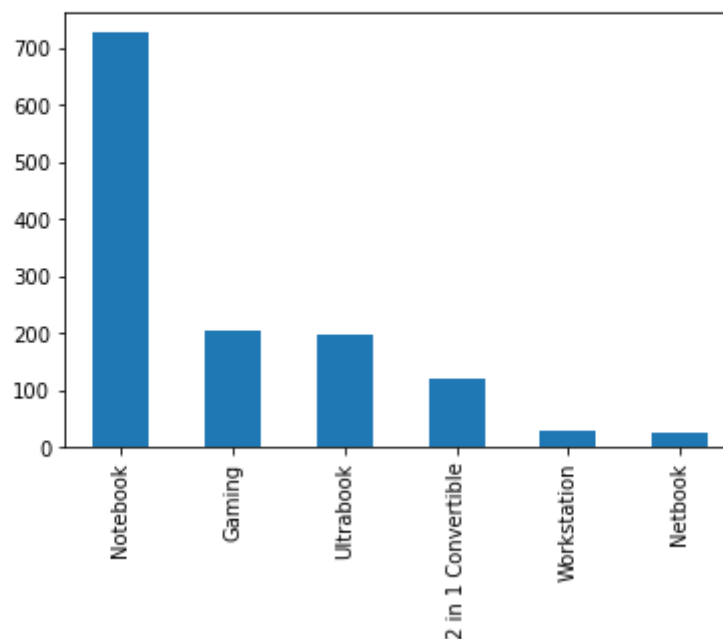


Figure 13 Histogram

From the above Fig. Relationship between Laptop TypeName is depicted. It is clearly seen that “Notebook” Type laptop has highest price following by “Gaming” Laptop and following by other types of laptop.

Scatter Plot:

```
In [31]: sns.scatterplot(x=df['Inches'],y=df['Price'])
```

```
Out[31]: <AxesSubplot:xlabel='Inches', ylabel='Price'>
```

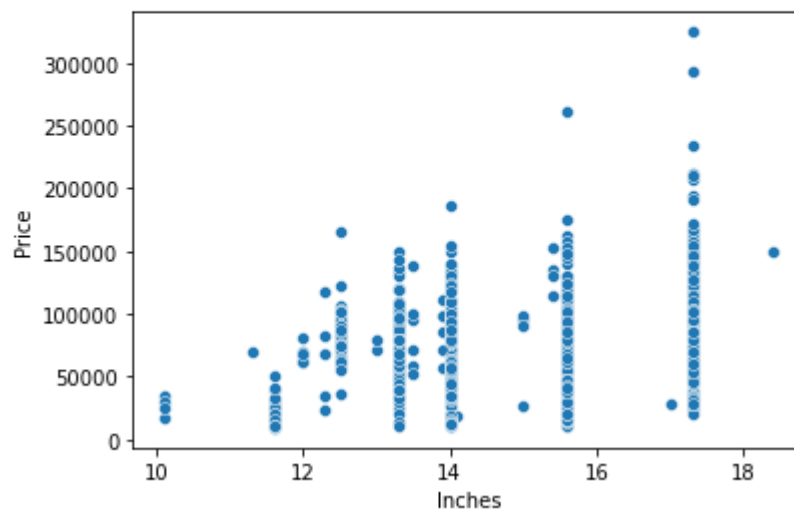


Figure 14 Scatter Plot

In the above fig We have drawn Scatter Plot to depict the relationship between Laptop Price and Screen Size. It is clearly seen that Laptop Price Increases as Screen Size increases.

Bar Graph:

```
In [29]: sns.barplot(x=df['TypeName'],y=df['Price'])  
plt.xticks(rotation='vertical')  
plt.show()
```

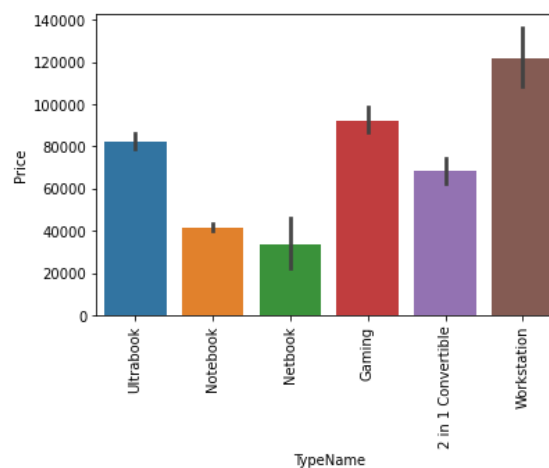


Figure 15 Bar Graph

Regression Techniques:

- 1) **Linear Regression:** Linear regression is a simple and commonly used regression technique that aims to model the relationship between a dependent variable (Y) and one or more independent variables (X) as a linear function. The model assumes that the relationship between Y and X is linear and tries to find the best-fitting line that minimizes the sum of the squared residuals between the predicted and actual values of Y.
- 2) **K-Neighbours Regressor:** K-Nearest Neighbours (KNN) Regressor is a non-parametric algorithm that can be used for both regression and classification problems. It works by finding the k-nearest neighbours to a given data point and using their average (or median) value as the predicted output.
- 3) **Decision Tree Regressor:** Decision Tree Regressor is a tree-based algorithm that recursively partitions the feature space into smaller regions based on the values of the features. It works by repeatedly splitting the data based on the feature that provides the best split, according to some criterion (e.g., minimizing the sum of squared errors).
- 4) **Random Forest Regressor:** Random Forest Regressor is an ensemble algorithm that combines multiple decision trees to improve the accuracy and robustness of the predictions. It works by training multiple decision trees on random subsets of the data and features, and then aggregating their predictions to make the final prediction.
- 5) **Gradient Boosting Regressor:** Gradient Boosting Regressor is another ensemble algorithm that combines multiple weak learners (usually decision trees) to create a strong predictor. It works by training each weak learner to correct the errors of the previous one, and then aggregating their predictions to make the final prediction.
- 6) **AdaBoost Regression:** AdaBoost Regression is another ensemble algorithm that combines multiple weak learners (usually decision trees) to create a strong predictor. It works by assigning weights to the training instances based on their difficulty to predict, and then training each weak learner to focus on the instances that were misclassified by the previous ones.
- 7) **Extra Trees Regressor:** Extra Trees Regressor is another ensemble algorithm that is similar to Random Forest Regressor. It works by training multiple decision trees on random subsets of the data and features, but with a different twist: it chooses the split points for each feature at random, rather than searching for the best split.

We have used different algorithms on the chosen dataset as follows:

1.Linear Regression: Linear regression is a statistical technique used to model the relationship between a dependent variable and one or more independent variables. It is commonly used to make predictions or estimate the value of the dependent variable based on the value of the independent variable(s).

In simple linear regression, there is only one independent variable, which is used to predict the value of the dependent variable. The relationship between the two variables is assumed to be linear, which means that a straight line can be used to approximate the relationship between the two variables.

The goal of linear regression is to find the line of best fit that minimizes the distance between the predicted values and the actual values of the dependent variable. This line is determined by estimating the values of the intercept and slope coefficients, which describe the position and slope of the line, respectively. Once the line of best fit has been determined, it can be used to make predictions about the dependent variable based on the values of the independent variable(s).

```
In [145... step1 = ColumnTransformer(transformers=[
    ('col_tnf',OneHotEncoder(sparse=False,drop='first'),[0,1,7,10,11])
],remainder='passthrough')

step2 = LinearRegression()

pipe = Pipeline([
    ('step1',step1),
    ('step2',step2)
])

pipe.fit(X_train,y_train)

y_pred = pipe.predict(X_test)

print('R2 score',r2_score(y_test,y_pred))
print('MAE',mean_absolute_error(y_test,y_pred))
```

Figure 16 Linear Regression

```
R2 score 0.8073277448418521
MAE 0.21017827976429174
```

Figure 17 Linear Regression Accuracy

2. Decision Tree Classifier: Decision tree is similar to the human decision-making process and so that it is easy to understand. Shannon invented the concept of entropy which measures the impurity if the input set. There are many benefits of decision tree. It can easily capture Non-linear pattern. The decision tree has no assumptions about distribution because of the non-parametric nature of the algorithm. To avoid biased tree, we balanced the data set before creating the model. Decision tree can handle uncertainty in both building and classification procedures. A decision tree classifier is a tree in which internal nodes are labelled by features. ... The classifier categorizes an object x_i by recursively testing for the weights that the features labelling the internal nodes have in vector x_i , until a leaf node is reached. The label of this node is then assigned to x_i

```
In [191... step1 = ColumnTransformer(transformers=[
    ('col_tnf', OneHotEncoder(sparse=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = DecisionTreeRegressor(max_depth=8)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)

print('R2 score', r2_score(y_test, y_pred))
print('MAE', mean_absolute_error(y_test, y_pred))
```

Figure 18 Creating Decision Tree

```
R2 score 0.8466456692979233
MAE 0.1806340977609143
```

Figure 19 Decision Tree Accuracy

3. SVM Regressor: A support vector machine (SVM) is a type of supervised machine learning classification algorithm which outputs an optimal hyperplane that categorizes new examples given labelled training data. SVMs were introduced initially in 1960s and were later refined in 1990s. However, it is only now that they are becoming very popular, owing to their ability to achieve outstanding results.

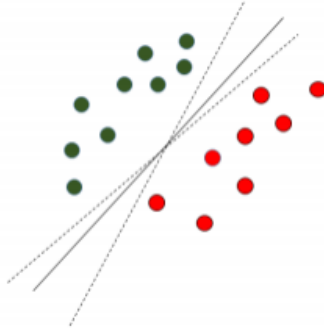


Figure 20 Multiple Decision Boundaries

```
In [213... step1 = ColumnTransformer(transformers=[
    ('col_tnf', OneHotEncoder(sparse=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = SVR(kernel='rbf', C=10000, epsilon=0.1)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)

print('R2 score', r2_score(y_test, y_pred))
print('MAE', mean_absolute_error(y_test, y_pred))
```

Figure 21 Creating SVM Regressor

```
R2 score 0.8083180902257614
MAE 0.20239059427481307
```

Figure 22 SVM Accuracy

4. Random Forest

Random forest is a tree-based algorithm which involves building several trees (decision trees), then combining their output to improve generalization ability of the model. The method of combining trees is known as an ensemble method. Ensembling is nothing but a combination of weak learners (individual trees) to produce a strong learner. Definition: A random forest is a classifier consisting of a collection of tree structured classifiers $h(x, \Theta_k)$, $k = 1, \dots$ where the Θ_k are independent identically distributed (i.i.d) random vectors and each tree casts a unit vote for the most popular class at input. Random Forest Algorithm: The following are the basic steps involved in performing the random forest algorithm:

- Pick N random records from the dataset.
- Build a decision tree based on these N records.
- Choose the number of trees you want in your algorithm and repeat steps (i) and (ii).

In case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

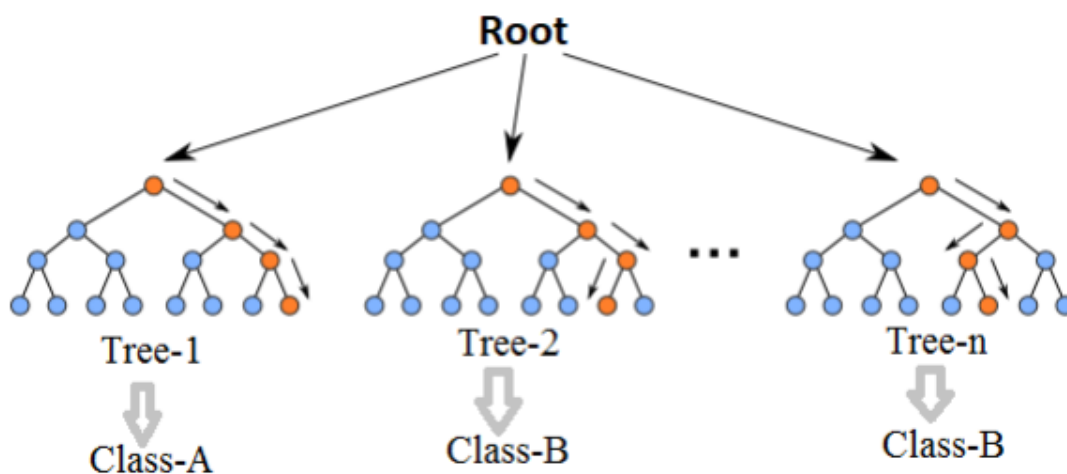


Figure 23 Random Forest Algorithm

What

ensemble does is take the mode (maximum occurring class) of the output produced by n different trees to create a better model. To say it in simple words: Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. The random forest is a classification algorithm consisting of many decisions trees. It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree.

```
In [306... step1 = ColumnTransformer(transformers=[
    ('col_tnf', OneHotEncoder(sparse=False, drop='first'), [0,1,7,10,11])
], remainder='passthrough')

step2 = RandomForestRegressor(n_estimators=100,
                             random_state=3,
                             max_samples=0.5,
                             max_features=0.75,
                             max_depth=15)

pipe = Pipeline([
    ('step1', step1),
    ('step2', step2)
])

pipe.fit(X_train, y_train)

y_pred = pipe.predict(X_test)

print('R2 score', r2_score(y_test, y_pred))
print('MAE', mean_absolute_error(y_test, y_pred))
```

Figure 24 Creating Random Forest

```
R2 score 0.8873402378382488
MAE 0.15860130110457718
```

Figure 25 Random Forest Algorithm Accuracy

DATA COLLECTION:

Data collection is the process of gathering and measuring data, information or any variables of interest in a standardized and established manner that enables the collector to answer or test hypothesis and evaluate outcomes of the particular collection.

There are total 11 columns and 1303 rows in this dataset. Information about columns is as follows:

- **Company:** Laptop Brand Name
- **TypeName:** Laptop Type Name
- **Inches:** Laptop Screen size in Inches
- **ScreenResolution:** Screen Resolution
- **CPU:** Laptop CPU Specs
- **RAM:** Laptop RAM in GB
- **Memory:** Laptop Storage in SSD/HDD
- **GPU:** Laptop GPU
- **OpSys:** Operating system
- **Weight:** Laptop Weight.

Data columns (total 13 columns):

- Company 10841 non-null object
- TypeName 10841 non-null object
- Inches 9367 non-null float64
- ScreenResolution 10841 non-null object
- CPU 10841 non-null object
- RAM 10841 non-null object
- Memory 10840 non-null object
- GPU 10841 non-null object
- OpSys 10840 non-null object
- Weight 10841 non-null object

- **CONCLUSION AND FUTURE WORK**

In conclusion, GadgetHive is a valuable tool for anyone looking to buy a laptop or smartphone. By providing accurate predictions of laptop prices and personalized recommendations based on user preferences, GadgetHive streamlines the selection process and helps users make informed decisions. The use of advanced algorithms such as cosine similarity and random forest ensures that the recommendations and predictions provided by GadgetHive are accurate and reliable.

For future work, GadgetHive could expand its services to include recommendations for other technology devices, such as tablets, smartwatches, and gaming consoles. The platform could also incorporate user reviews and feedback to further enhance the accuracy and reliability of its recommendations. Additionally, GadgetHive could explore partnerships with retailers and manufacturers to offer users exclusive deals and discounts on recommended devices. Overall, GadgetHive has great potential to evolve and improve in the future, providing users with even better and more personalized recommendations and predictions.

In addition to expanding its services and incorporating user reviews and feedback, GadgetHive could also explore the use of machine learning techniques to further enhance its algorithms. For instance, the platform could use natural language processing (NLP) to better understand user preferences and requirements, allowing it to provide even more personalized recommendations. GadgetHive could also explore the use of reinforcement learning to optimize its recommendation engine based on user interactions.

Overall, there are many opportunities for GadgetHive to evolve and improve in the future, providing users with an even more personalized and efficient way of choosing the right device.

6. REFERENCES

1. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
2. Brownlee, J. (n.d.). Machine Learning Mastery.
Retrieved from <https://machinelearningmastery.com/>
3. Kaggle. (n.d.). Kaggle.
Retrieved from <https://www.kaggle.com/>
4. GitHub. (n.d.). GitHub.
Retrieved from <https://github.com/>
5. Towards Data Science. (n.d.). Towards Data Science.
Retrieved from <https://towardsdatascience.com/>