PROJECT REPORT

ON

Data analysis Using Emerging Power Bi

Submitted in Partial fulfilment of the

Requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

Computer Science & Engineering

Submitted by

Ravikesh Kumar Singh(2103600100075)

Aditya Gupta (2103600100008)

Bipin kharwar(2103600100037)

Shubham Jaiswal(2103600100094)

Under the guidance

Of

Mrs Namita Srivastava

DEPARTMENT OF COMPUTER SCIENCE

GOEL INSTITUTE OF TECHNOLOGY AND MANAGEMENT, LUCKNOW

Affiliated to

DR. A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW,

May-2025

GOEL INSTITUTE OF TECHNOLOGY & MANAGEMENT, LUCKNOW

Affiliated

A. P. J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW, INDIA

BONAFIDE CERTIFICATE

This is to certify that the project report entitled "**REGRESSION ANALYSIS USING EMERGING POWER BI"** submitted by **Ravikesh Kumar Singh (2103600100075), Aditya Gupta (2103600100008)** , **Bipin kharwar (2103600100037), Shubham Jaiswal (2103600100094)** to Goel Institute of Technology, Lucknow in partial fulfillment of the requirement for the award of the degree of B-Tech in CSE is a record of project undertaken by him/her under my supervision. The report fulfills the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Mrs. Namita Srivastava

Assistant Professor CSE Department

(Project Guide)

Internal Examiner (s) **External Examiner (s)**

Science & Engineering

GOEL INSTITUTE OF TECHNOLOGY & MANAGEMENT

CERTIFCATE

Certify that the project entitled "**REGRESSION ANALYSIS USING EMERGING POWER BI"** submitted by **Ravikesh Kumar Singh (2103600100075), Aditya Gupta (2103600100008) , Bipin kharwar (2103600100037), Shubham Jaiswal (2103600100094)** in the partial fulfillment of the requirements for the award of the degree of the Bachelor of Technology (Computer Science & Engineering) of DR. A P J ABDUL KALAM TECHNICAL UNIVERSITY, is a record of student's own work carried our supervision and guidance. The project report embodies results of the original work and studies carried out by students and the contents do not form the basis for the award of any other degree to the candidate or to anybody else.

Mrs. Namita Srivastava
(Project Guide)

COMPUTER SCIENCE & ENGINEERING

GOEL INSTITUTE OF TECHNOLOGYAND & MANAGEMENT

DECLARATION OF CANDIDATE

We hereby declare that the project entitled "**REGRESSION ANALYSIS USING EMERGING POWER BI"** submitted by us in the partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (CS) of DR. A P J ABDUL KALAM TECHNICAL UNIVERSITY is record of our own work carried under the supervision and guidance of **Mrs. Namita Srivastava** to the best of our knowledge this project has not been submitted to DR. A P J ABDUL KALAM TECHNICAL UNIVERSITY or any other University or Institute for the award of any degree.

Ravikesh Kumar Singh
(2103600100075)

Aditya Gupta
(2103600100008)

Bipin kharwar
(2103600100037)

Shubham Jaiswal
(2103600100094)

PREFACE

The Topic covered under each are as follows:

1. Abstract
2. Introduction
3. Problem Statement
4. Literature Review
5. Methodology of solution
6. Steps for analytic
   - **6.1** Understanding Requirement
     - **6.1.1** Importing modules
     - **6.1.2** Reading of data
     - **6.1.3** declaring the dataset as a variable
     - **6.1.4** Detailing
     - **6.1.5** Description
   - **6.2** Design data Requirements
     - **6.2.1** Finding null tuples
     - **6.2.2** Removal of null tuples
     - **6.2.3** Managing null values according to attribute
   - **6.3** Processing data or Exploratory Data Analysis
     - **6.3.1** Performing analytics: model creation
     - **6.3.2** Model testing
     - **6.3.3** Test train splitting
     - **6.3.4** Model scoring
     - **6.3.5** Model comparision
   - **6.4** Visualization
     - **6.4.1** Visualization tools
     - **6.4.2** Visuals on excel
     - **6.4.3** Creating dashboard on excel
     - **6.4.4** Visualization on power BI
     - **6.4.5** Creation of dashboard on Power BI
7. Conclusion
8. Future Scope
9. Glossary
10. References

TABLE OF CONTENT

LIST OF TABLES

ACKNOWLEDGEMENT

We would like to express my deepest gratitude to all those who supported and guided me throughout the journey of completing my final year project.

First and foremost, We are extremely thankful to my project supervisor, [Supervisor's Name], for their invaluable guidance, encouragement, and support throughout the project. Their insights and constructive feedback have been crucial to the success of this work.

We would also like to thank B-tech in CSE at Goel Institute of technology and Management for providing the resources and a conducive environment for learning and research.

Our sincere thanks to all the faculty members and staff who have imparted their knowledge and offered their support during my academic journey.

We are grateful to my team members/classmates/friends who collaborated with me, shared ideas, and provided continuous support.

Last but not least, we would like to thank my family for their unconditional love, patience, and moral support during the entire course of my studies and project work.

This project would not have been possible without the contribution and encouragement of everyone mentioned above.

Ravikesh Kumar Singh
(2103600100075)

Aditya Gupta
(2103600100008)

Bipin kharwar
(2103600100037)

Shubham Jaiswal
(2103600100094)

1.Abstract

This research paper explores the application of data analytics to an Airbnb dataset containing 74,111 listings, focusing on variables such as room type, accommodates, bathrooms, cancellation policy, cleaning fee, instant book ability, review scores, bedrooms, beds, and log-transformed price. Using Python libraries including pandas, NumPy, and seaborn, we perform exploratory data analysis (EDA) to uncover trends and relationships within the data. The study highlights key statistical insights and identifies potential sources of human error that could impact data quality and analytical and implement KNN algorithm to treat outlier values . The findings provide a foundation for understanding pricing dynamics in the Airbnb market and underscore the importance of addressing human-induced inaccuracies in workflows.

2.Introduction

The process of making sense of something is called analysis, and the process of making sense of the data that is available is called data analytics. The management of data, which includes the gathering and storing of said data from a variety of sources, as well as the utilization of procedures, tools, and techniques to evaluate said data, is at the heart of this area. By analyzing data and making inferences from it, the purpose of data analytics is to derive correlations, obtain insights, and locate patterns. These actionable insights not only help firms with the decision making process, but they also help with generating predictions and boosting efficiency . Amidst market uncertainties and various geopolitical crises including the Russia-Ukraine conflict, the need for resilience has expanded beyond organizations to encompass governments, citizens, armed forces, education, and other stakeholders. Data analytics and sciences are playing an essential role in a wide range of socio-economic and political initiatives, such as managing the displacement and rehabilitation of refugees, mitigating climate change, reducing food waste, enhancing aid programs' effectiveness, and more. As Web 3.0 and the metaverse continue to grow and gain adoption, organizations and other entities must consider incorporating them into their data analytics initiatives. Additionally, with humans and artificial intelligence collaborating and complementing each other in unprecedented ways, the next wave of data analytics is expected to provide optimal insights and decision-making capabilities. This will result in competitive advantages and enable businesses to adhere to their key performance indicators

This study analyses an Airbnb dataset extracted from an Excel file (‘Air_BNB.xlsx’), as processed in a Jupyter Notebook environment. The dataset includes 74,111 records with 10 variables after dropping an identifier column (‘id’). The primary objective is to perform exploratory data analysis (EDA) to summarize the dataset's characteristics and explore potential relationships, while also considering the role of human error in data collection and preprocessing.

3. Problem statement

Regression analysis is a fundamental statistical method used to model relationships between a dependent variable and one or more independent variables. However, the presence of outliers—data points that deviate significantly from other observations—can adversely affect the performance of regression models, leading to biased estimates and reduced predictive accuracy.

The KNN algorithm, a non-parametric method, is sensitive to the local structure of the data. Outliers can distort the distance metrics used in KNN, thereby influencing the selection of neighbors and, consequently, the regression outcomes. Therefore, addressing outliers is crucial for enhancing the efficacy of KNN regression models.

It focuses to develop a robust regression model employing the k-Nearest Neighbors (KNN) algorithm, incorporating effective outlier detection and treatment strategies to improve prediction accuracy and model reliability.

4. Literature Review

4.1 Machine Learning

Machine learning is an artificial intelligence (AI) discipline geared toward the technological development of human knowledge. Machine learning allows computers to handle new situations via analysis, self-training, observation and experience.

Machine learning facilitates the continuous advancement of computing through exposure to new scenarios, testing and adaptation, while employing pattern and trend detection for improved decisions in subsequent (though not identical). situations.

Machine learning is often confused with data mining and knowledge discovery in databases (KDD), which share a similar methodology.

Machine learning algorithms are often categorized as supervised or unsupervised. Supervised algorithms require a data scientist or data analyst with machine learning skills to provide both input and desired output, in addition to furnishing feedback about the accuracy of predictions during algorithm training. Data scientists determine which variables, or features, the model should analyze and use to develop predictions. Once training is complete, the algorithm will apply what was learned to new data. Unsupervised algorithms do not need to be trained with desired outcome data. Instead, they use an iterative approach called deep learning to review data and arrive at conclusions.[5]

Unsupervised learning algorithms -- also called neural networks -- are used for more complex processing tasks than supervised learning systems, including image recognition, speech-to- text and natural language generation. These neural networks work by combing through millions of examples of training data and automatically identifying often subtle correlations between many variables. Once trained, the algorithm can use its bank of associations to interpret new data. These algorithms have only become feasible in the age of big data, as they require massive amounts of training data.

4.1.1 Types of machine learning problems

There are various ways to classify machine learning problems. Here, we discuss the most

obvious ones.

1. On basis of the nature of the learning "signal" or "feedback" available to a learning system

- **Supervised Learning**
    a) **Classification**: You train with images/labels. Then in the future you give a new image expecting that the computer will recognize the new object.
    b) **Market Prediction/Regression**: You train the computer with historical market data and ask the computer to predict the new price in the future.
- **Unsupervised learning**: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. It is used for clustering population in different groups. Unsupervised learning can be a goal in itself (discovering hidden patterns in data).

a) **Clustering:** You ask the computer to separate similar data into clusters, this is essential in research and science.
b) **High Dimension Visualization**: Use the computer to help us visualize high dimension data.
c) **Generative Models**: After a model captures the probability distribution of your input data, it will be able to generate more data. This can be very useful to make your classifier more robust.

A simple diagram which clears the concept of supervised and unsupervised learning is shown below

As you can see clearly, the data in supervised learning is labelled, whereas data in unsupervised learning is unlabelled.

- **Semi-supervised learning**: Problems where you have a large amount of input data and only some of the data is labelled, are called semi-supervised learning problems. These problems sit in between both supervised and unsupervised learning. For example, a photo archive where only some of the images are labelled, (e.g. dog, cat, person) and the majority are unlabelled.

Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The program is provided feedback in terms of rewards and punishments as it navigates its problem space

2. On the basis of "output" desired from a machine learned system

- **Classification:** Inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".
- **Regression:** It is also a supervised learning problem, but the outputs are continuous rather than discrete. For example, predicting the stock prices using historical data.

- **Clustering:** Here, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

- **Density estimation:** The task is to find the distribution of inputs in some space.
- **Dimensionality reduction:** It simplifies inputs by mapping them into a lower-dimensional space. Topic modelling is a related problem, where a program is

given a list of human language documents and is tasked to find out which documents cover similar topics.

4.1.2. List of Common Algorithms

- **k-Nearest Neighbour:** The k-nearest neighbours' algorithm (k-NN) is a non-parametric method used for classification and regression.[8] In both cases, the input

consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression:

I. In k-NN classification, the output is a class membership. An object is classified by a plurality vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours (k is a positive integer, typically small). If k-1, then the object is simply assigned to the class of that single nearest neighbour.

II. In k-NN regression, the output is the property value for the object. This value is the average of the values of k nearest neighbours.

- **Naive Bayes**: Naive Bayes is a simple, yet effective and commonly-used, machine learning classifier. It is a probabilistic classifier that makes classifications using the Maximum A Posteriori decision rule in a Bayesian setting. It can also be represented using a very simple Bayesian network. Naive Bayes classifiers have been especially popular for text classification, and are a traditional solution for problems such as spam detection.

- **Linear Regression:** Simple linear regression is useful for finding relationship between two continuous variables. One is predictor or independent variable and other is response or dependent variable. It looks for statistical relationship but not deterministic relationship. Relationship between two variables is said to be deterministic if one variable can be accurately expressed by the other. For example, using temperature in degree Celsius it is possible to accurately predict Fahrenheit. Statistical relationship is not accurate in determining relationship between two variables. For example, relationship between height and weight. The core idea is to obtain a line that best fits the data. The best fit line is the one for which total prediction error (all data points) are as small as possible. Error is the distance between the point to the regression line.

- **Support Vector Machines (SVM):** The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

- To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e. the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.
- **Neural Networks:** A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. Thus, a neural network is either a biological neural network, made up of real biological neurons, or an artificial neural network, for solving artificial intelligence (AI) problems. The connections of the biological neuron are modelled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1. or it could be -1 and 1.

4.1.3 Regression Analytics

Regression analytics is a statistical method used to examine the relationship between one dependent variable and one or more independent variables. Its primary goals are to model, understand, and predict numerical outcomes based on input data.

At its simplest form, regression answers the question: How does a change in one or more independent variables affect the dependent variable?

4.2.1. Basic Concept

Regression analytics attempts to find a mathematical equation that best fits a dataset. For example, in a simple linear regression, where one independent variable predicts a dependent variable , the relationship is modeled as:

Y = \beta_0 + \beta_1 X + \epsilon

- is the intercept (value of when)
- is the slope (change in for a one-unit change in)
- is the error term (the part of not explained by the model)

4.2.2. Multiple Linear Regression

When more than one independent variable is used, the model becomes:

Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon

This is called multiple linear regression, and it’s useful when multiple factors influence the target variable.

4.2.3. Purpose of Regression Analytics

Prediction: Forecasting future values based on current and historical data.
Estimation: Quantifying the effect of independent variables on the dependent variable.
Explanation: Understanding relationships between variables in real-world systems.
Control: Identifying key drivers that can be adjusted to influence outcomes.

4.2.4. Types of Regression Models

Linear Regression: Assumes a linear relationship.
Polynomial Regression: Models non-linear relationships using polynomial terms.
Logistic Regression: Used when the dependent variable is binary (e.g., yes/no).
Ridge and Lasso Regression: Regularized models to prevent overfitting when dealing with high-dimensional data.
Non-parametric Regression: No fixed functional form (e.g., decision trees or splines).

4.2.5. Assumptions in Linear Regression

For classical linear regression to produce valid results, several assumptions must be met:

Linearity: The relationship between predictors and response is linear.
Independence: Observations are independent of each other.
Homoscedasticity: The variance of errors is constant across all levels of .
Normality: The residuals (errors) are normally distributed.
No multicollinearity: Independent variables are not too highly correlated.
Violations of these assumptions can lead to biased or inefficient estimates.

4.2.6. Model Evaluation

Regression models are commonly evaluated using:

- R-squared ( ): Proportion of variance in the dependent variable explained by the model.
- Adjusted R-squared: Modified that accounts for the number of predictors.
- Mean Squared Error (MSE) or Root Mean Squared Error (RMSE): Measures the average squared difference between actual and predicted values.
- p-values and confidence intervals: Test the statistical significance of individual coefficients.

4.2.7. Applications of Regression Analytics

Regression analytics is widely used in various domains:

- Business: Forecasting sales, pricing strategies, customer lifetime value.
- Healthcare: Predicting disease progression or treatment outcomes.
- Economics: Estimating economic growth based on macroeconomic indicators.
- Social Sciences: Studying the impact of education, income, or demographics on social outcomes.

Summary

Regression analytics is a versatile and powerful method used to explore, model, and forecast relationships between variables. While simple in concept, the depth of theory behind regression—ranging from assumptions and estimation methods to evaluation and diagnostics—makes it a central tool in both theoretical and applied data analysis.

5.Methodology

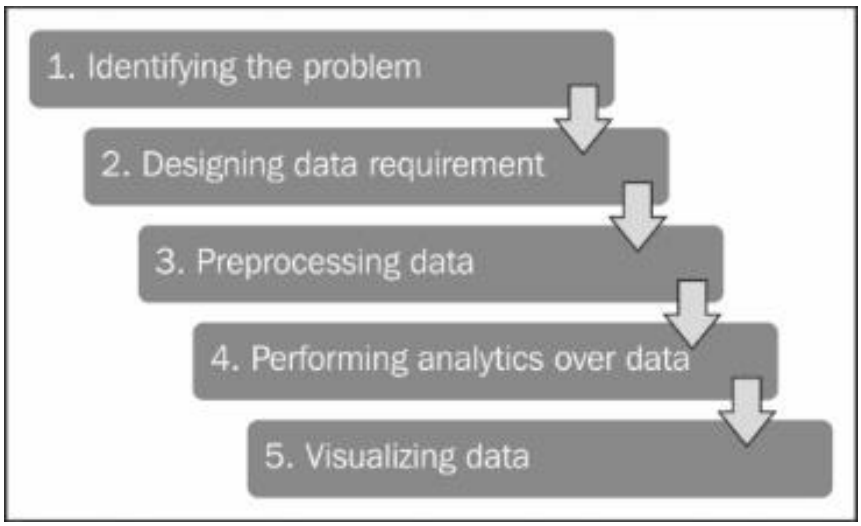


fig-1. Process of data analytics

6.Steps in data Analytics

6.1. Identifying Problem:

At this step we have to clear out all our requirements to make data planning according to the data need for analysis and decision making for our business requirement and future decisions. In this we use the steps to make data outlier removal to find data accuracy and comparision with replacement with mean , median and knn imputer.

In this we do first some steps for data reading and understanding:

6.1.1. importing the modules: now first we import all the important modules required for our data Figure 2

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

```
from sklearn import metrics
import matplotlib.pyplot as plt
import matplotlib.style
```

The image shows Python import statements used in our data analysis project. Here's a breakdown of the libraries being imported

- numpy as np: Used for numerical operations and working with arrays.
- pandas as pd: Used for data manipulation and analysis, especially with DataFrames.
- seaborn as sns: A statistical data visualization library built on top of matplotlib.
- LinearRegression from sklearn.linear_model: A machine learning model used for performing linear regression.
- metrics from sklearn: Provides tools for measuring model performance (e.g., mean squared error, accuracy).
- matplotlib.pyplot as plt: A widely used plotting library for creating static, interactive, and animated visualizations.
- matplotlib.style: Allows for changing the visual style of plots (e.g., 'ggplot', 'seaborn' themes).

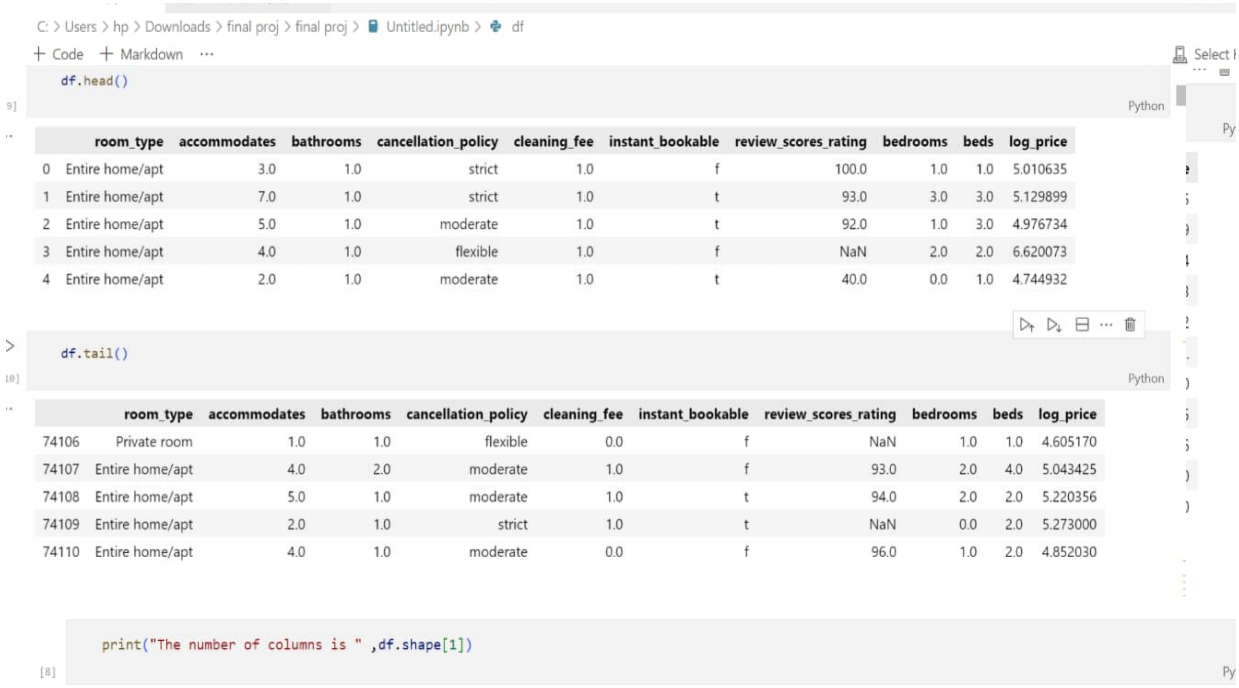
6.1.2 Reading of data: in this first we read the data set of xlsx format of 71114 rows provided by air bnb hotel services and it is done with the help of pandas library of python.

```
[3]  ## importing data

[4]  df = pd.read_excel('Air_BNB.xlsx')

[5]  df
```

	id	room_type	accommodates	bathrooms	cancellation_policy	cleaning_fee	instant_bookable	review_scores_rating	bedrooms	beds
0	6901257	Entire home/apt	3.0	1.0	strict	1.0	f	100.0	1.0	1.0
1	6304928	Entire home/apt	7.0	1.0	strict	1.0	t	93.0	3.0	3.0
2	7919400	Entire home/apt	5.0	1.0	moderate	1.0	t	92.0	1.0	3.0
3	13418779	Entire home/apt	4.0	1.0	flexible	1.0	f	NaN	2.0	2.0
4	3808709	Entire home/apt	2.0	1.0	moderate	1.0	t	40.0	0.0	1.0

Figure 3

6.1.3 declaring the dataset as a variable : now we declare our dataset as a variable df(df stands for data frame).



	room_type	accommodates	bathrooms	cancellation_policy	cleaning_fee	instant_bookable	review_scores_rating	bedrooms	beds	log_price
0	Entire home/apt	3.0	1.0	strict	1.0	f	100.0	1.0	1.0	5.010635
1	Entire home/apt	7.0	1.0	strict	1.0	t	93.0	3.0	3.0	5.129899
2	Entire home/apt	5.0	1.0	moderate	1.0	t	92.0	1.0	3.0	4.976734
3	Entire home/apt	4.0	1.0	flexible	1.0	f	NaN	2.0	2.0	6.620073
4	Entire home/apt	2.0	1.0	moderate	1.0	t	40.0	0.0	1.0	4.744932

	room_type	accommodates	bathrooms	cancellation_policy	cleaning_fee	instant_bookable	review_scores_rating	bedrooms	beds	log_price
74106	Private room	1.0	1.0	flexible	0.0	f	NaN	1.0	1.0	4.605170
74107	Entire home/apt	4.0	2.0	moderate	1.0	f	93.0	2.0	4.0	5.043425
74108	Entire home/apt	5.0	1.0	moderate	1.0	t	94.0	2.0	2.0	5.220356
74109	Entire home/apt	2.0	1.0	strict	1.0	t	NaN	0.0	2.0	5.273000
74110	Entire home/apt	4.0	1.0	moderate	0.0	f	96.0	1.0	2.0	4.852030

Detailing: now we read our file in various dimensions.

5.1.5

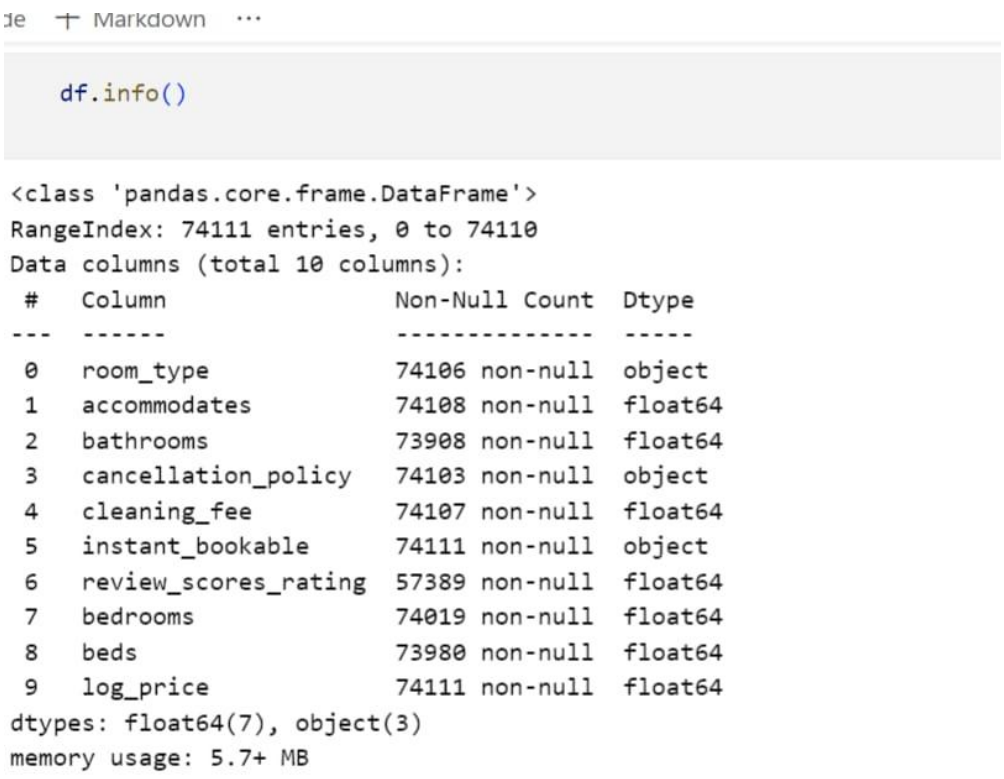


Figure 4

Description: now we find all detailing knowledge for file like mean, median , mode ,min, max, standerd deviation et.c.

```
round(df.describe(),2)
```

	accommodates	bathrooms	cleaning_fee	review_scores_rating	bedrooms	beds	log_price
count	74108.00	73908.00	74107.00	57389.00	74019.00	73980.00	74111.00
mean	3.16	1.24	0.73	94.07	1.27	1.71	4.78
std	2.15	0.58	0.44	7.84	0.85	1.25	0.72
min	1.00	0.00	0.00	20.00	0.00	0.00	0.00
25%	2.00	1.00	0.00	92.00	1.00	1.00	4.32
50%	2.00	1.00	1.00	96.00	1.00	1.00	4.71
75%	4.00	1.00	1.00	100.00	1.00	2.00	5.22
max	16.00	8.00	1.00	100.00	10.00	18.00	7.60

Figure 5

6.2 Design data Requirements:

Here we design our data requirements to treat them for basic functionalities.

6.2.1 Finding null tuples: here we find all count of null tuples available in our data set

Command

df2.isnull().sum()

Interpretation:

This command checks for missing (null) values in each column of the DataFrame df2 and returns the sum of null values per column.

Summary of Missing Values:

- room_type: 5 missing
- accommodates: 3 missing
- bathrooms: 195 missing
- cancellation_policy: 8 missing
- cleaning_fee: 4 missing

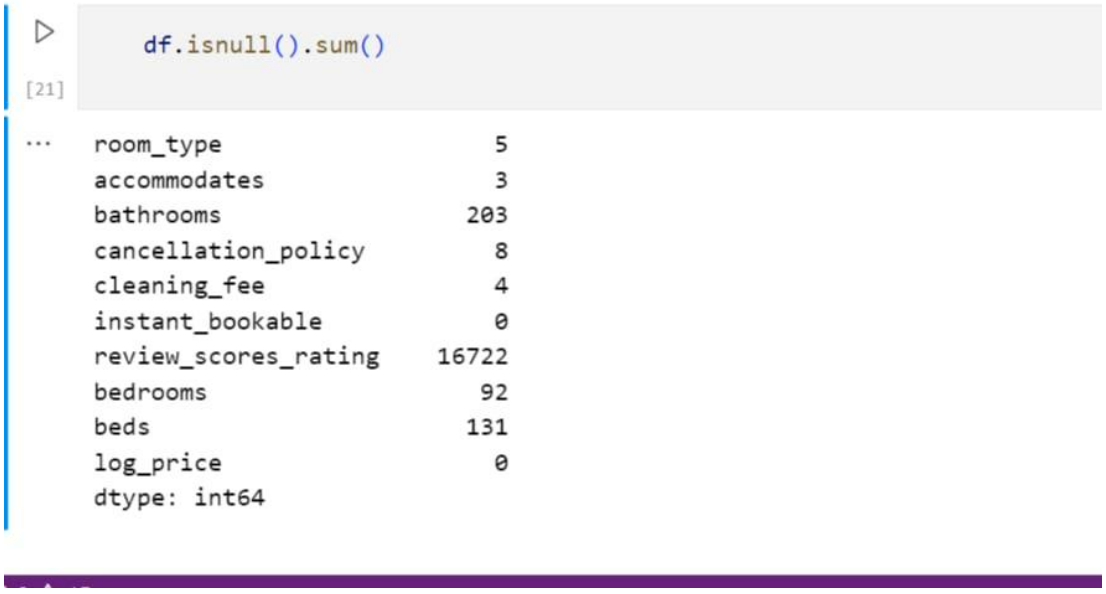
- instant_bookable: 0 missing
- review_scores_rating: 10,215 missing ❗ (significantly high)
- bedrooms: 92 missing
- beds: 125 missing
- log_price: 0 missing

Notes:

review_scores_rating has the most missing values by far, which may need special attention like imputation or column exclusion.

instant_bookable and log_price have no missing data.

The data type of the result is int64.



```
[21]  df.isnull().sum()

...   room_type                 5
      accommodates              3
      bathrooms               203
      cancellation_policy       8
      cleaning_fee              4
      instant_bookable          0
      review_scores_rating  16722
      bedrooms                 92
      beds                    131
      log_price                 0
      dtype: int64
```

Figure 5

6.2.2 Removal of null tuples: here we remove all null tuples from our dataset

6.2.3 Count of null tuples according to each atttibutes after

```
df2.drop_duplicates(inplace = True)
```

```
df2
```

	room_type	accommodates	bathrooms	cancellation_policy	cleaning_fee	instant_bookable	review_scores_rating	bedrooms	beds	log_price
0	Entire home/apt	3.0	1.0	strict	1.0	f	100.0	1.0	1.0	5.010635
1	Entire home/apt	7.0	1.0	strict	1.0	t	93.0	3.0	3.0	5.129899
2	Entire home/apt	5.0	1.0	moderate	1.0	t	92.0	1.0	3.0	4.976734
3	Entire home/apt	4.0	1.0	flexible	1.0	f	NaN	2.0	2.0	6.620073
4	Entire home/apt	2.0	1.0	moderate	1.0	t	40.0	0.0	1.0	4.744932
...	...	...	...	...	...	...	...	...	...	...
74104	Entire home/apt	2.0	1.0	strict	1.0	f	100.0	1.0	1.0	4.356709
74107	Entire home/apt	4.0	2.0	moderate	1.0	f	93.0	2.0	4.0	5.043425
74108	Entire home/apt	5.0	1.0	moderate	1.0	t	94.0	2.0	2.0	5.220356
74109	Entire home/apt	2.0	1.0	strict	1.0	t	NaN	0.0	2.0	5.273000
74110	Entire home/apt	4.0	1.0	moderate	0.0	f	96.0	1.0	2.0	4.852030

54117 rows × 10 columns

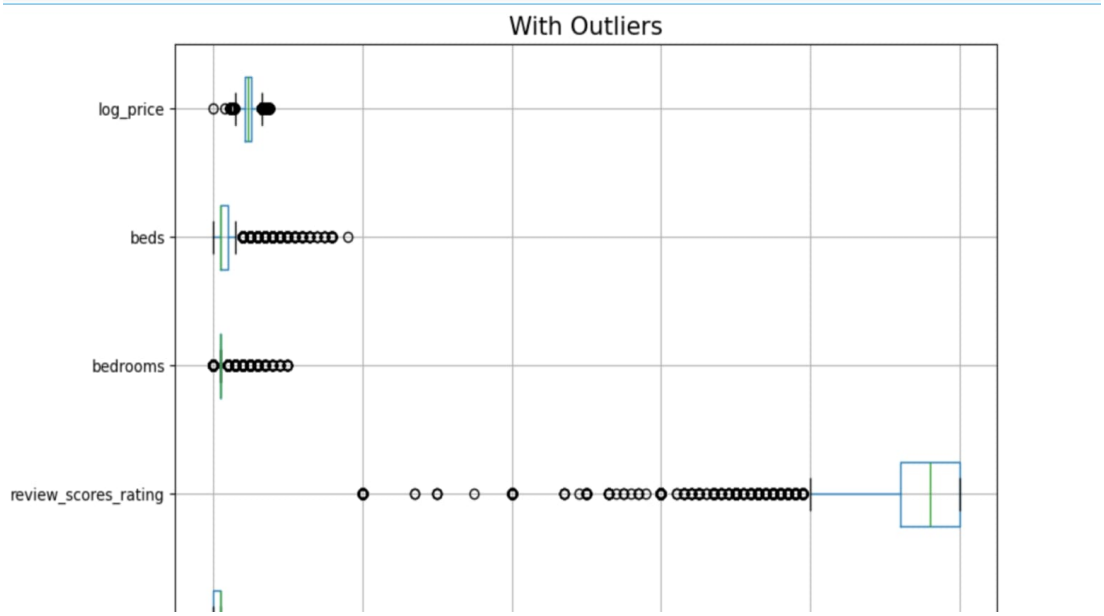
removing duplicates

```
df2.isnull().sum()
```

```
room_type                  5
accommodates               3
bathrooms                195
cancellation_policy        8
cleaning_fee               4
instant_bookable           0
review_scores_rating    10215
bedrooms                  92
beds                     125
log_price                  0
dtype: int64
```

Figure 6

6.2.4 finding outliers using box plot: here we find specific range of outliers in box plot.



6.2.5 Copying dataset: now we copy our dataset as df2, df_mean,df_med, df_knn. This Python code using the copy() method from pandas. Here's what each line does:

code

df_med_out = df_med.copy()

df_knn_out = df_knn.copy()

Explanation:

- **df_med and df_knn:** These are likely two existing pandas DataFrames.
- **.copy():** Creates a deep copy of the DataFrame. This means that df_med_out and df_knn_out are independent copies of df_med and df_knn respectively.

Any changes made to df_med_out or df_knn_out will not affect the original DataFrames (df_med or df_knn), and vice versa.

Use Case:

This approach is for:

- Preserve the original data.
- Experiment with data transformations (e.g., outlier removal, imputation, etc.) in separate copies.
- Here we are preparing three separate datasets (df_mean_out, df_med_out, df_knn_out) likely for further preprocessing or model training.

6.3. Processing data: here we desing data processing methodology like data oulier removal function, knn imputer, changing outlier with mean, median, and knn value.

6.3.1 Design of oulier removal function:here we design function for outlier removal and deletion.

Exploratory data analysis

```
def remove_outlier(col):
    sorted(col)
    Q1,Q3=np.percentile(col,[25,75])
    IQR=Q3-Q1
    lower_range= Q1 - (1.5 * IQR)
    upper_range= Q3 + (1.5 * IQR)
    return lower_range, upper_range
```

```
remove_outlier(df_mean['review_scores_rating'])
```

6.3.2 implementation of outlier function

```
remove_outlier(df_mean['review_scores_rating'])
```

```
(83.0, 107.0)
```

6.3.3 Removing from KNN data set and copying in new variable

This is a Python code snippet for handling missing data in a DataFrame and visualizing outliers.

First Block:

- Imports **KNNImputer from sklearn.impute** to impute missing values using the k-Nearest Neighbors algorithm.
- Creates an imputer with n_neighbors=5.Fits and transforms the DataFrame df2 to impute missing values, storing the result in df_imputed.
- Converts the imputed data back into a pandas DataFrame df_knn with the same columns as df2.

Second Block:

- Identifies columns in df with data types uint8 (unsigned 8-bit integer) or bool (boolean).
- Uses matplotlib.pyplot (aliased as plt) to create a boxplot of these columns to visualize outliers.
- Sets the figure size to 10x10 inches, labels the title as "With Outliers" with a font size of 16, and displays the plot.The code is likely part of a data preprocessing pipeline to handle missing values and inspect outliers in a dataset.

```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=5)
df_imputed=imputer.fit_transform(df2)
df_knn = pd.DataFrame(data = df_imputed,columns=df2.columns)
```

```
cont=df.dtypes[(df.dtypes != 'uint8') & (df.dtypes != 'bool')].index
plt.figure(figsize=(10,10))
df[cont].boxplot(vert = 0)
plt.title('With Outliers',fontsize = 16)
plt.show()
```

6.3.4 Doing z score analysis through heat map

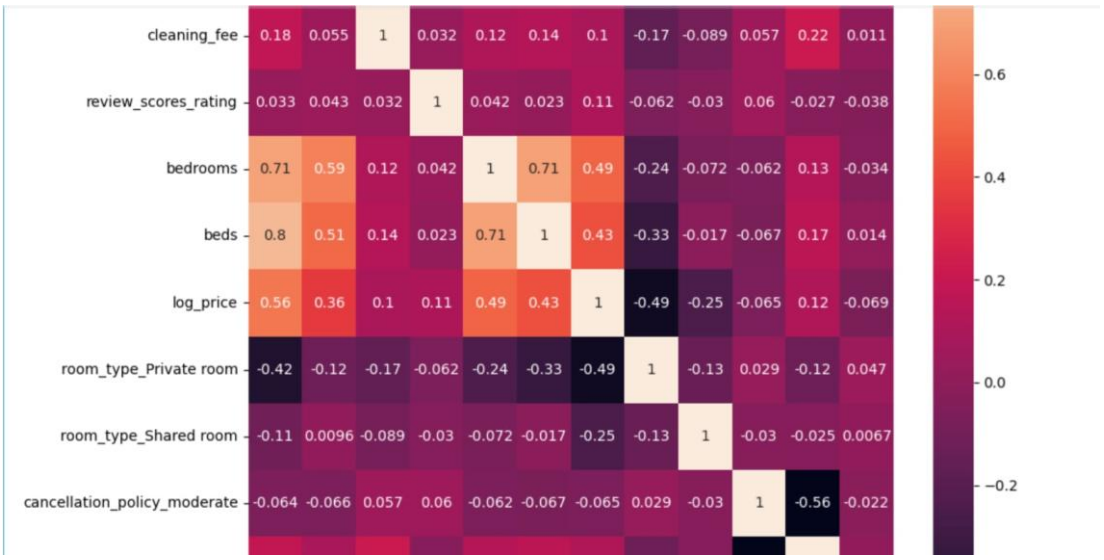


Figure 7

This is a correlation heatmap, which visually represents the correlation matrix between different features (columns) in a dataset. Each cell in the heatmap shows the Pearson correlation coefficient between two variables, which ranges from -1 to 1:

Indications:

- 1 indicates a perfect positive correlation.
- 1 indicates a perfect negative correlation.
- 0 indicates no correlation.

Observations from the Heatmap:

Strong Positive Correlations:

1. beds and bedrooms: 0.71
2. beds and log_price: 0.43
3. bedrooms and log_price: 0.49
4. beds and cleaning_fee: 0.14
5. cleaning_fee and log_price: 0.56

Strong Negative Correlations:

1. room_type_Private room and log_price: -0.49

2. cancellation_policy_moderate and room_type_Private room: -0.56
3. room_type_Private room and beds: -0.33
4. room_type_Private room and bedrooms: -0.24

Low or No Correlation:

review_scores_rating is nearly uncorrelated with most features.

room_type_Shared room has weak correlations with other variables.

Interpretation of Key Features:

1. Listings with more beds and bedrooms tend to have higher prices.
2. A Private room is negatively associated with the price, number of beds, and bedrooms, suggesting that such listings are cheaper and smaller.
3. Listings with a moderate cancellation policy tend to not be private rooms (negative correlation).

Color Gradient:

The color bar on the side explains the gradient: dark colors (black/purple) indicate negative correlations, and bright colors (orange/red) indicate positive correlations.

Use Case:

This kind of heatmap is often used in exploratory data analysis (EDA) to understand relationships between variables and to inform feature selection for machine learning models.

6.4 Performing analytics: here we perform advance analytics like model training and model score finding and comparision.

6.4.1 creation of our model

Here the data is being prepared for training a machine learning model, specifically for predicting log_price. The code is executed in Visual Studio Code (VS Code) with the Python extension enabled.

What's in the Code:

1. Feature-Target Split

```
X = df_mean_out.drop('log_price', axis=1)

y = df_mean_out[['log_price']]
```

- X is the feature set — all columns except 'log_price'.
- y is the target variable — the 'log_price' column.
- df_mean_out likely refers to a cleaned dataset (possibly with missing values filled using mean imputation, and outliers present).

2. Checking Target Data

```
y.head()
```

Displays the first 5 values of the log_price column:

log_price

0 5.010635

1 5.129899

2 4.976734

3 6.620073

4 4.744932

These are log-transformed price values, useful for reducing **skewness** in price prediction problems.

3. Train-Test Split

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

- Splits the data into 70% training and 30% testing.

- random_state=1 ensures reproducibility Summary:
- This block of code is setting up the dataset to train a regression model. It uses df_mean_out:
- Features (X) and target (y) are separated.
- A preview of y shows log-transformed price values.

The data is split into training and testing sets.

```
47]  from sklearn.model_selection import train_test_split
     X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
```

```
45]  X = df_mean_out.drop('log_price',axis=1)
     y = df_mean_out[['log_price']]
```

```
X = df_mean_out.drop('log_price',axis=1)
y = df_mean_out[['log_price']]
```

45] Python

```
y.head()
```

46] Python

	log_price
0	5.010635
1	5.129899
2	4.976734
3	6.620073
4	4.744932

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
```

47] Python

⚠ 15 Spaces: 4 {} ⚠ Select Interpreter Cell 62 of 73

6.4.2 performing linear regression impution

This image shows a code snippet from a Jupyter notebook where a Linear Regression model is being trained using scikit-learn, and its R^2 score is calculated on the training dataset.

Code :

```
regression_model = LinearRegression()

regression_model.fit(X_train, y_train)
```

- A Linear Regression model is initialized and trained using the training dataset (X_train, y_train).
- The model is assigned to the variable regression_model.

```
regression_model.score(X_train, y_train)
```

This computes the R^2 score on the training set.

Output shown: 0.5026453886994027

Interpretation:

- The R^2 score (~0.50) means the model explains about 50% of the variance in the training data. This is a moderate performance, and may indicate:
- There could be outliers or noise in the data.
- Features may not be strongly predictive of the target (log_price).

- Further feature engineering or model improvement might be needed.

```
regression_model = LinearRegression()
regression_model.fit(X_train,y_train)
```

```
LinearRegression
LinearRegression()
```

```
# df_mean_out
regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

6.4.3 model scoring

```
# df_mean_out
regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

This image shows running a Python script that applies machine learning using scikit-learn. The key operations shown involve training a regression model with and with outlier treatment.

Key Points from the Image:

1. Model: regression_model

The same model is trained in three different cases with varying data preprocessing steps.

- df_mean_out (Possibly Outliers Present)

- regression_model1.score(X_train, y_train)
- Shows training score (R^2): 0.5026
- This suggests that the model explains about 50% of the variance in the training set.
- Variable regression_model is used.

Summary:

This notebook is testing how different preprocessing strategies (handling missing values and outliers) affect the performance of a regression model predicting log_price. The score reported is the R^2 score, showing model performance on the training data.

6.4.4 Removing from med & knn data set and copying in new

```
#med  outlier not treated
X = df_med.drop('log_price',axis=1)
y = df_med[['log_price']]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
regression_model.fit(X_train,y_train)

regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

variable

Here linear regression models are trained using different imputed datasets, both without treating outliers. The goal is to observe how the choice of imputation (median vs. KNN) affects model performance on training data.

Median Imputation (df_med)

#med outlier not treated

X = df_med.drop('log_price', axis=1)

y = df_med[['log_price']]

regression_model.score(X_train, y_train)

- df_med is a dataset where missing values were likely imputed using median.
- Outliers are not removed.
- Train-test split is 70-30.

The model achieves an R^2 score of:

0.5026453886994027

KNN Imputation (df_knn)

#knn outlier not treated

X = df_knn.drop('log_price', axis=1)

y = df_knn[['log_price']]

regression_model.score(X_train, y_train)

- df_knn is a dataset where missing values were imputed using KNN (k-nearest neighbors).
- Again, outliers are not treated.

The model gives the exact same R^2 score:

0.5026453886994027

Key Observations:

- Both median and KNN imputation lead to the same training score ($R^2 \approx 0.5026$).
- This suggests that the imputation method does not significantly affect model performance — at least on the training set — when outliers are not treated.
- It's also possible that the imputed values ended up being quite similar, or that missing values weren’t widespread.

Compare these with:

- Imputation + outlier treatment (df_med_out, df_knn_out, etc.)
- Evaluate test set performance using regression_model.score(X_test, y_test)

6.4.5 Model comparision on scoring

```
#knn  outlier not treated
X = df_knn.drop('log_price',axis=1)
y = df_knn[['log_price']]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
regression_model.fit(X_train,y_train)

regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

```
#knn  outlier treated
X = df_knn_out.drop('log_price',axis=1)
y = df_knn_out[['log_price']]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
regression_model.fit(X_train,y_train)

regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

```
#med  outlier not treated
X = df_med.drop('log_price',axis=1)
y = df_med[['log_price']]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
regression_model.fit(X_train,y_train)

regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

```
#knn  outlier not treated
X = df_knn.drop('log_price',axis=1)
y = df_knn[['log_price']]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test =train_test_split(X, y, test_size = 0.3, random_state =1)
regression_model.fit(X_train,y_train)

regression_model.score(X_train,y_train)
```

```
0.5026453886994027
```

```
#knn  outlier treated
```

6.5 Data visualization: Creating data visualizations is an effective way to transform raw data into interactive and insightful reports. Here we perform data visualization on two open source and most popular softwares:

1. MS Excel
2. Power BI

6.5.1 📈 Data Visualization in Microsoft Excel – A Complete Overview

Microsoft Excel is a powerful tool for data analysis and visualization, offering a range of chart types and features to represent data clearly and effectively. Here's a comprehensive breakdown of data visualization in Excel:

🧰 Key Data Visualization Tools in Excel

1. Charts

Excel offers a variety of built-in chart types:

- Column/Bar Charts: Compare categories (e.g., sales by region).
- Line Charts: Show trends over time (e.g., revenue growth)
- Pie/Doughnut Charts: Show proportions (e.g., market share)
- Scatter Plots: Analyze relationships between two variables.
- Area Charts: Emphasize volume over time.
- Combo Charts: Combine two chart types (e.g., line and column).
- Histogram: Show frequency distributions.
- Box & Whisker, Waterfall, Funnel (Excel 2016+): Specialized analytics.

📌 Access:

- Insert → Charts Group
- Or use Recommended Charts for automatic suggestions.

2. Pivot Tables & Pivot Charts

Summarize large datasets easily.

Add interactivity with slicers and timelines.

Can be used to create dynamic dashboards.

📌 Access:

Insert → PivotTable / PivotChart

3. Slicers & Timelines

Slicers: Provide clickable filters for PivotTables and PivotCharts.

Timelines: Date-based filter for time series data.

📌 Access:

Insert → Slicer or Insert → Timeline

4. Conditional Formatting

- Visualize data directly in cells using:
- Color scales
- Data bars
- Icon sets

Great for heatmaps, status indicators, etc.

📌 Access:

Home → Conditional Formatting

5. Sparklines

Tiny, in-cell charts to show trends.

Types: Line, Column, Win/Loss

📌 Access:

Insert → Sparklines

6. Maps

Use Filled Map chart to visualize geographical data.

📌 Access:

Insert → Maps (Available in Excel 2019 and later

7. Form Controls / ActiveX Controls

Add interactive elements like

- Drop-downs
- Checkboxes
- Sliders

Useful for dashboards and user input.

📌 Access:

Developer Tab → Insert → Form Controls

🖌️ Customization & Design Tools

- Chart Elements (titles, labels, legends, gridlines)
- Chart Styles & Layouts
- Axis formatting (scales, number formats)

- Data Labels and Trendlines
- Custom Colors and Fonts
- Themes to maintain visual consistency

📌 Access:

Click on a chart → Chart Design & Format tabs

🧱 Dashboard Design in Excel

- Combine multiple charts and controls.
- Use slicers to link all charts.
- Format shapes, colors, and layout for professional appearance.
- Hide gridlines and headings for a clean look.

✅ **Best Practices for Excel Data Visualization**

Principle	Explanation
Keep it simple Avoid clutter	Focus on the message
Use the right chart	Choose a chart type that fits your data.
Label clearly	Always include axis titles, legends, and labels
Use color wisely	Use contrasting colors to highlight key data.
Tell a story	Guide the viewer through your findings.

6.5.2 Creating dashboard on MS Excel:

Here an Excel dashboard named "Airbnb Price & Listing Dashboard", created to analyze various factors related to Airbnb listings. Below is a breakdown of all the visualizations, filters (slicers), and tools used in the dashboard:

🔧 **Tools & Features Used:**

- Microsoft Excel
- Pivot Tables & Charts
- Slicers (for interactive filtering)
- Dashboard Design Tools (for layout, colors, etc.)

🎯 **Dashboard Title**

"Airbnb Price & Listing Dashboard" – Clearly stated at the top to represent the focus of the data.

📊 **Visualizations and Their Descriptions:**

1. **Average Price by Room Type (Top Left - Bar Chart)**

Chart Type: Bar Chart (Vertical Columns)

Visualizes the average price for each room type:

Entire home/apt

Private room

Shared room

Background has a light red/pink theme for emphasis.

2. Average Review Score by Room Type (Top Center - 3D Bar Chart)

Chart Type: 3D Column Chart

Displays total or average review score for different room types.

Color-coded with golden brown bars.

3. Average Price by Bedrooms (Bottom Left - Line Chart)

Chart Type: Line Chart

Shows how the average price varies by the number of bedrooms (0 to 10).

Useful for identifying trends based on listing size.

4. Cleaning Fee Impact on Average Price (Bottom Center - Bar Chart)

Chart Type: Horizontal Bar Chart

Compares listings with or without a cleaning fee.

Indicates that listings with cleaning fees generally have higher prices.

5. Count of Listing by Cancellation Policy (Bottom Right - Pie Chart)

Chart Type: Pie Chart

Displays the distribution of listings across three cancellation policies:

Flexible

Moderate

Strict

Useful for understanding user preference or host offerings.

🔍 Filters (Slicers on the Left Side):

These slicers allow dynamic filtering of the dashboard data based on specific criteria:

1. room_type – Filters data by:

Entire home/apt Private room

2. bedrooms – Filters based on the number of bedrooms (0 to 13 visible).

3. cancellation_policy – Currently filtered to flexible.

4. instant_bookable – Filter for listings that allow instant booking (t stands for true).

5. cleaning_fee – Two options:

TRUE: Listings with a cleaning fee

FALSE: Listings without a cleaning fee

🧩 Sheet & File Info:

Sheet name: DASHBOARD

File: Likely contains multiple sheets including Air_BNB, Sheet3, Sheet4, etc.

6.5.3 Data visualization on Power BI

Certainly! Here's a comprehensive guide on data visualization in Power BI, one of the most powerful business intelligence tools by Microsoft.

💡 What is Power BI?

Power BI is a business analytics tool that allows you to:

- Connect to various data sources
- Transform and model data
- Create interactive visual reports and dashboards
- Share insights with teams or online

It's designed to help non-technical users visualize data with ease and gain insights quickly.

🧰 Key Features of Data Visualization in Power BI

Feature	Description
Interactive Dashboards	Combine visuals, slicers, filters on one page
Drill-Down Capability	Click into a data point to explore deeper
Cross Filtering	Click on one visual to affect others
Cross Filtering	Auto-refresh dashboards from live data sources
Custom Visuals	Use community or third-party visuals from AppSource

📊 Common Visualizations in Power BI

Visual Type	Best Used For
Bar/Column Chart	Compare values across categories
Line/Area Chart	Show trends over time
Pie/Donut Chart	Show parts of a whole
Table/Matrix	Display raw or summarized data in rows/columns
Cards/KPIs	Show summary statistics (e.g., totals, averages)
Scatter/Bubble Chart	Analyze relationships and distribution
Gauge/Funnel Chart	Progress or conversion funnels
Map Visuals	Show geographic data by region or location
Decomposition Tree	Drill down into hierarchies
Waterfall Chart	Show cumulative effects of values (e.g., profit flow)

🪄 How to Create Visuals in Power BI (Step-by-Step)

1. Connect to Data

Excel, SQL Server, Web, APIs, Azure, etc.

2. Transform Data

Use Power Query Editor to clean, reshape, merge, or split data.

3. Model Relationships

Define how tables relate (primary/foreign keys).

4. Create Visuals

Drag and drop fields into the canvas and choose a visual type.

5. Customize Visuals

Change colors, add titles, tooltips, filters, and sorting.

6. Create Filters & Slicers

Add slicers for date, category, region, etc.

7 Publish & Share

Publish to Power BI Service and share via web or mobile.

🔍 **Interactivity Features**

Feature	Use
Slicers	Add interactive filters to dashboards
Drill Through	Right-click to go to a detailed page
Bookmarks	Save different views or filter states
Tooltips	Show extra data when hovering
Drill Down/Up	Explore data by levels (e.g., year > month > day)

🛠 **Examples of Power BI Dashboard Use Cases**

Industry	Use Case
Sales	Sales performance, pipeline tracking
Finance	Profit & loss, expense analysis
Operations	Supply chain metrics, efficiency tracking
Marketing	Campaign ROI, audience segmentation
HR	Headcount, attrition rate, diversity stats

Power BI vs Excel for Visualization

Feature	Power BI	Excel
Real-time dashboards	✅	❌
Interactive visuals	✅	❌ (limited)
Large data handling	✅	❌
Data modeling (DAX)	✅	limited
Ease of sharing online	✅	❌
Familiar interface	❌ (new UI)	✅

6.5.4 Creation of dashboard on Power Bi

The dashboard which is titled "AIR BNB ANALYSIS", and it is created using Microsoft Power BI. It visually represents various metrics and dimensions related to Airbnb listings. Here's a detailed breakdown of what each section of the dashboard shows:

Dashboard Components

1. Filters (Left Sidebar):

Room Types:

- Entire home/apt
- Private room
- Shared room

These can be selected to filter the visualizations based on room preference.

Instant Bookable (instant_bookable):

- t (true)
- f (false)

Allows filtering listings based on whether they can be instantly booked or not.

2. Visualization Tiles:

a. Booking of Rooms (Top Center - Bar Chart):

Shows the count of accommodations based on selected room type.

In the current filter view, only "Shared room" is selected and displayed.

b. Sum of Log Price (Top Middle - Card):

Displays the total (or possibly average) logarithmic price of listings: 2K.

c. Ratings (Top Right - Card):

Shows total ratings: 2K.

d. Cancellation Policy (Bottom Left - Pie Chart):

Shows the distribution of cancellation policies:

- Flexible: 44.24%
- Strict: 37.68%
- Moderate: 18.08%

e. Accommodation Breakdown (Bottom Right - Donut Chart):

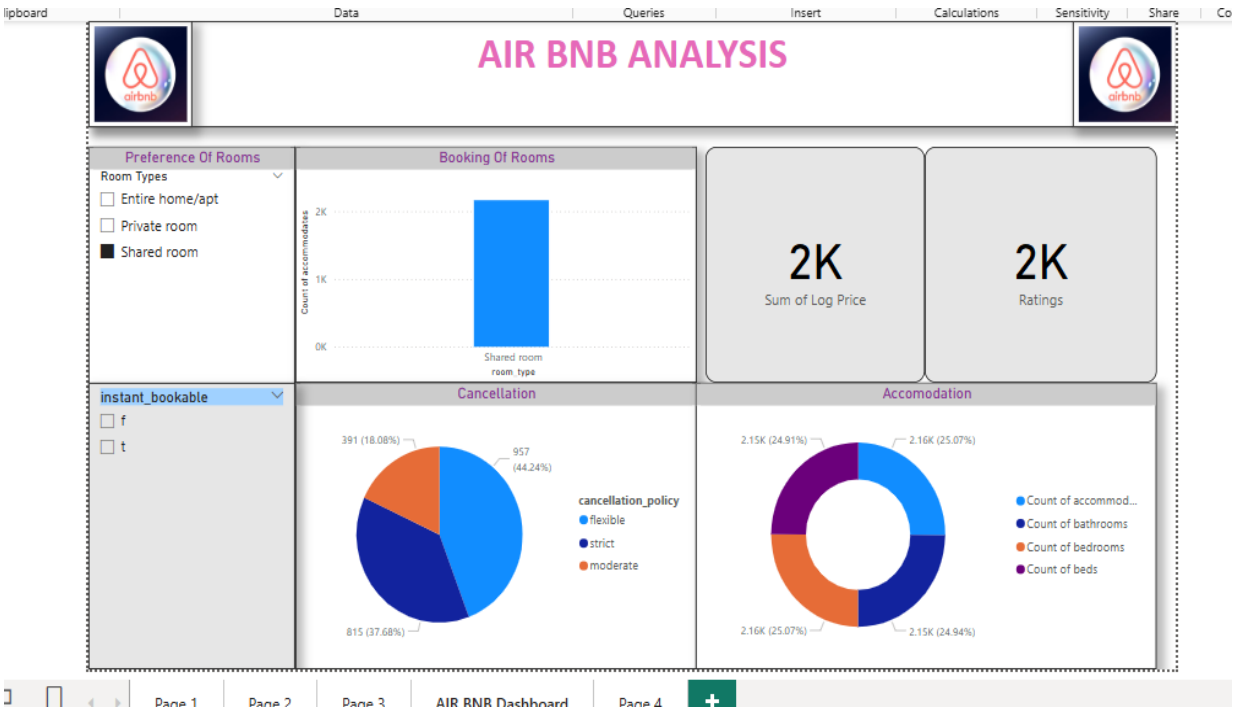
Distribution of:

- Count of accommodations
- Count of bathrooms
- Count of bedrooms
- Count of beds

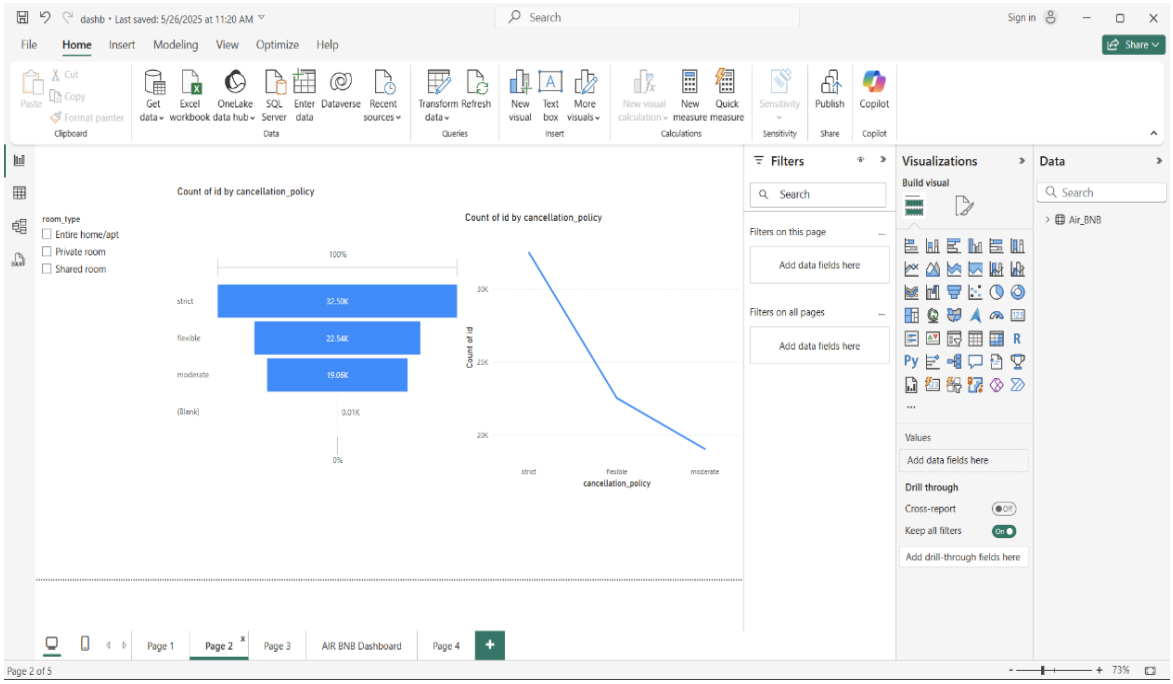
Each is represented equally at about 25% each, indicating balanced data distribution.

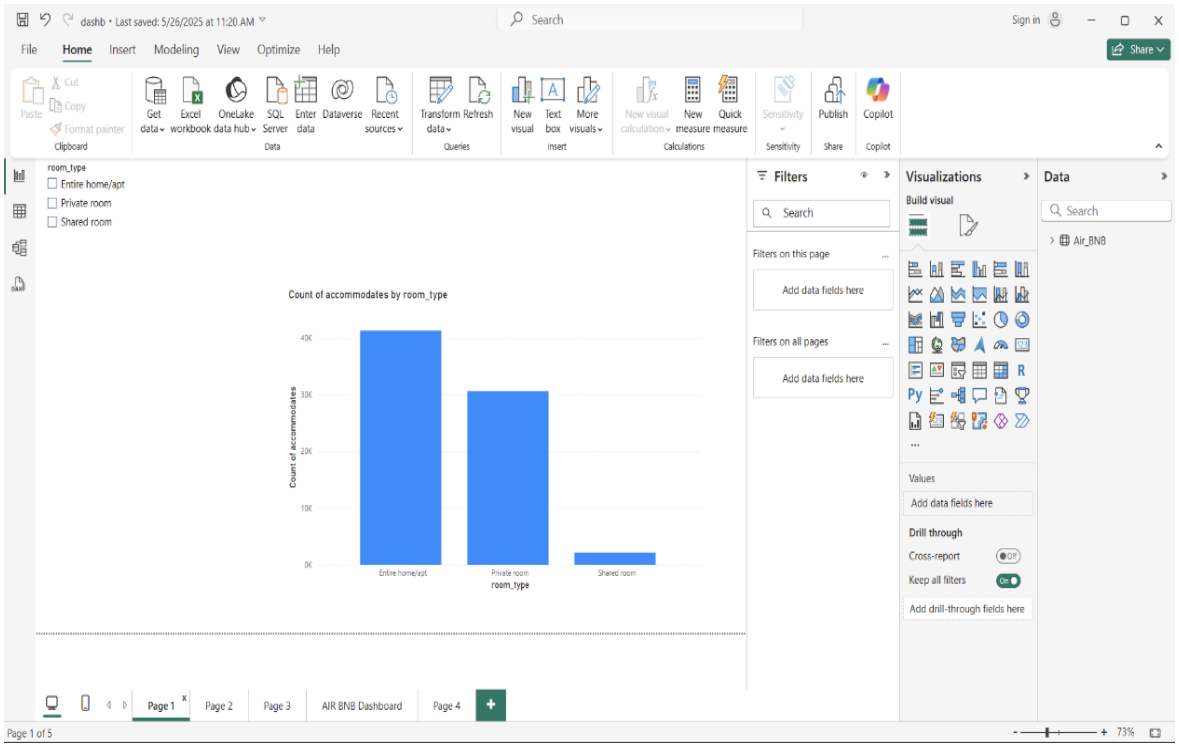
Observations:

- The visual theme is clean and uses a light background with readable font sizes.
- The main metrics (price and ratings) are clearly highlighted.
- Filters allow for dynamic updates based on room types and booking options.
- Pie and donut charts offer quick insight into categorical data like cancellation policies and property features.



Screenshot of Power bi Report




dashb • Last saved: 5/26/2025 at 11:20 AM
Search
Sign in
File
Home
Insert
Modeling
View
Optimize
Help
Share
Paste
Cut
Copy
Format painter
Clipboard
Get data
Excel workbook
OneLake data hub
SQL Server
Enter data
Dataverse
Recent sources
Data
Transform data
Refresh
Queries
New visual
Text box
More visuals
Insert
New visual calculation
New measure
Quick measure
Calculations
Sensitivity
Publish
Share
Copilot
room_type
Entire home/apt
Private room
Shared room
Count of accommodates by room_type
Count of accommodates
40K
30K
20K
10K
0K
Entire home/apt
Private room
Shared room
room_type
Filters
Search
Filters on this page
Add data fields here
Filters on all pages
Add data fields here
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Off
Keep all filters
On
Add drill-through fields here
Data
Search
Air_BNB
Page 1
Page 2
Page 3
AIR BNB Dashboard
Page 4
Page 1 of 5
73%

Images of charts on comparisions

7.Conclusion

The Power BI project has effectively demonstrated the ability to convert large volumes of raw data into insightful, easy-to-understand visual reports and dashboards. Through the use of Power BI's interactive tools and advanced analytical features, we were able to identify key trends, highlight performance indicators, and support data-driven decision-making.

Throughout the project, we explored the full workflow—from data import and transformation using Power Query to building dynamic dashboards using DAX functions and visualization elements. The end result was a comprehensive and user-friendly reporting system that could be easily interpreted by both technical and non-technical users.

This project emphasizes the growing importance of business intelligence tools like Power BI in today's data-driven environments and provides a solid foundation for future developments in real-time reporting and predictive analytics.\

Overall, the integration of machine learning techniques with business intelligence tools offered a comprehensive approach to data analytics. This project has not only enriched our understanding of regression modeling but also equipped us with practical skills to handle real-world data and generate actionable insights, laying a strong foundation for future endeavors in data science and analytics.

8.Future scope

The integration of machine learning algorithms with powerful data visualization tools such as Power BI opens vast opportunities for expansion and enhancement in future projects. Based on the outcomes and learning from this study, several directions can be explored to extend the scope and impact of the current work:

1. Advanced Machine Learning Models

While this project focused on linear regression and KNN-based techniques, future work can explore more advanced regression models such as Random Forest Regression, Gradient Boosting, or Neural Networks to improve prediction accuracy and handle non-linear relationships more effectively.

2. Real-Time Data Integration

The current analysis is based on static datasets. Future implementations can connect Power BI dashboards to real-time data sources (e.g., APIs, live databases) for dynamic updates and real-time decision-making capabilities.

3. Automated Data Pipeline with ETL

Building an automated Extract, Transform, Load (ETL) pipeline can streamline the data preprocessing phase, enabling seamless and efficient handling of large, continuously updated datasets.

4. Deployment as a Web-Based Dashboard

The insights generated can be deployed as an interactive web application or cloud dashboard accessible to stakeholders, enabling broader usability and real-time access to predictive analytics.

5. Incorporating Geospatial Analysis

Integrating map-based visuals and geolocation features can uncover geographic trends and patterns in Airbnb pricing, which would be highly useful for businesses targeting location-specific marketing and service improvements.

6. User Personalization and Recommendation System

Future models could incorporate user behavior data to develop personalized recommendations, enhancing user experience and increasing engagement in platforms similar to Airbnb.

7. Scalability with Big Data Tools

The project can be extended using big data frameworks like Apache Spark or Hadoop to manage and analyze even larger datasets efficiently, allowing for industrial-scale deployment.

8. Enhanced Power BI Capabilities with DAX and AI Visuals

Leveraging Data Analysis Expressions (DAX) and integrating AI-powered visuals in Power BI can lead to more powerful analytics and deeper insights.

In conclusion, the foundation laid by this project sets the stage for a wide array of future enhancements that combine data science, machine learning, and business intelligence. These developments can significantly benefit industries such as hospitality, e-commerce, real estate, and smart city planning.

9.Glossary

Term	Definition
Power BI	A business analytics tool by Microsoft used for interactive data visualization and business intelligence reporting.
Dashboard	A collection of visual elements (charts, KPIs, slicers) that display key metrics and trends.
DAX (Data Analysis Expressions)	A formula language used in Power BI to perform calculations and data manipulation.
Data Model	The structured representation of data tables and relationships in Power BI.
Dataset	A collection of data imported into Power BI from various sources for analysis.
Power Query	A data transformation tool in Power BI used to clean, reshape, and merge data.
Visualization	Graphical representation of data (e.g., charts, maps, gauges) used to convey insights.
KPI (Key Performance Indicator)	A measurable value that indicates how well an individual or organization is achieving a key business objective.
Slicer	A filtering tool in Power BI that allows users to select values to refine data visuals.
Measure	A calculated field in Power BI created using DAX to perform aggregate or dynamic calculations.
Report	A detailed view consisting of multiple pages of visuals derived from a dataset.

10. Reference

[1] Baig, M. I., Shuib, L., & Yadegaridehkordi, E. (2020). Big data in education: A state of the art, limitations, and future research directions. International Journal of Educational Technology in Higher Education, 17(1), 44. https://doi.org/10.1186/s41239-020-00223-0

[2] Katkar, S. V., Kharade, S. K., Kharade, K. G., & Kamat, R. K. (2020). Integration of Technology for Advancement in Supply Chain Management. In New Paradigms in Business Management Practices (Vol. 3, pp. 116–123). Amazon Publication.

[3] Kharade, K. G., Kharade, S. K., Sonawane, V. R., Bhamre, S. S., Katkar, S. V., & Kamat, R. K. (2021). IoT Based Security Alerts for the Safety of Industrial Area. In M. Rajesh, K. Vengatesan, M. Gnanasekar, Sitharthan.R, A. B. Pawar, P. N. Kalvadekar, & P. Saiprasad (Eds.), Advances in Parallel Computing. IOS Press. https://doi.org/10.3233/APC210185

[4] Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit: The Defi nitive Guide to Dimensional Modeling (3rd ed.). Wiley.

[5] Maheshwari, A. (2014). Data Analytics Made Accessible.

[6] Sonavane, A. K. K. (2021). Study of Emerging Role of Data Science in Business Intelligence. Design Engineering, 6.

[7] Marchena Sekli, G. F., & De La Vega, I. (2021). Adoption of Big Data Analytics and Its Impact on Organizational Performance in Higher Education Mediated by Knowledge Management. Journal of Open Innovation: Technology, Market, and Complexity, 7(4), 221. https://doi.org/10.3390/joitmc7040221

[8] Maroufkhani, P., Wagner, R., Wan Ismail, W. K., Baroto, M. B., & Nourani, M. (2019). Big Data Analytics and Firm Performance: A Systematic Review. Information, 10(7), 226. https://doi.org/10.3390/info10070226

[9] Naikwadi, B. H., Kharade, K. G., Yuvaraj, S., & Vengatesan, K. (2021). A Systematic Review of Blockchain Technology and Its Applications. In Recent Trends in Intensive Computing (pp. 467–473). IOS Press.

[10] Patil, S., Mujawar, A., Kharade, K. G., Kharade, S. K., Katkar, S. V., & Kamat, R. K. (2022). Drowsy Driver Detection Using Opencv And Raspberry Pi3. Webology, 19(2), 6003–6010.

[11] Popovič, A., Hackney, R., Tassabehji, R., & Castelli, M. (2018). The impact of big data analytics on firms' high value business performance. Information Systems Frontiers, 20(2), 209–222. https://doi.org/10.1007/s10796-016-9720-4

[12] Sonavane, A. K. K. (2021). An In-Depth Study of Retail Sales Trend and Pattern based on Exploratory Data Analysis. Design Engineering, 6313-6327.

[13] Prathima, Ch., Muppalaneni, N. B., & Kharade, K. G. (2022). Deduplication of IoT Data in Cloud Storage. In Ch. Satyanarayana, X.-Z. Gao, C.-Y. Ting, & N. B. Muppalaneni (Eds.), Machine Learning and Internet of Things for Societal Issues (pp. 147–157). Springer Nature Singapore. https://doi.org/10.1007/978-981-16-5090-1_13

[14] Sarker, I. H. (2021). Data Science and Analytics: An Overview from Data-Driven Smart Computing, Decision-Making and Applications Perspective. SN Computer Science, 2(5), 377. https://doi.org/10.1007/s42979-021-00765-8

[15] Swami, A., Patil, A., & Kharade, K. G. (2019). Applications of IoT for Smart Agriculture or Farming. International Journal of Research and Analytical Reviews, 6(2), 537–540.