**PROJECT REPORT**

**ON**

**Predicting House Prices with Machine Learning**

****

**SUBMITTED BY : DIVYANSHU MISHRA**

**YASH PARCHA**

**SAGAR**

**DEVANSH GARG**

**INSTITUTE OF TECHNOLOGY AND SCIENCE**

**MOHAN NAGAR, GHAZIABAD**

**SINCE 1995**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **1.** | **Introduction to Python and Machine Learning** | **3-4** |
| **2.** | **IMPORTING DATA** | **6** |
| **3.** | **Gathering Basic information** | **7-11** |
| **4.** | **Identification of Outliers using Box Plot and Outliers** | **12-17** |
| **5.** | **RESETING INDEX AFTER REMOVING OUTLIERS** | **18** |
| **6.** | **Dataset Trend Visualization** | **19-26** |
| **7.** | **Test and Train Splitting** | **27** |
| **8.** | **Logistic REGRESSOR** | **28** |
| **9.** | **ACCURACY AND PREDICTION** | **29** |
| **10.** | **DECISION TREE** | **30** |
| **11.** | **RANDOM FOREST** | **31-32** |
| **12.** | **K - Nearest Neighbor** | **33** |
| **13.** | **Gaussian Naive Bayes** | **34-35** |
| **14.** | **DATA COMPARISION** | **36** |

**Introduction to Python and Machine Learning**

**Introduction to python :**

Python is a general-purpose, dynamically typed, high-level, compiled and interpreted, garbage-collected, and purely object-oriented programming language that supports procedural, object-oriented, and functional programming. It was Created by Guido van Rossum and first released in 1991, Python emphasizes code readability and allows programmers to express concepts in fewer lines of code compared to languages like C++ or Java.

**Why learn Python?**

o Easy to use and Learn: Python has a simple and easy-to-understand syntax, unlike traditional languages like C, C++, Java, etc., making it easy for beginners to learn.

o Interpreted Language: Python does not require compilation, allowing rapid development and testing. It uses Interpreter instead of Compiler.

o Object-Oriented Language: It supports object oriented programming i.e(inheritance,encapsulation,polymorphism,abstraction) making writing reusable and modular code easy.

o Extensive Libraries : Python has a rich ecosystem of libraries and frameworks, such as NumPy, Pandas, and Matplotlib, which simplify tasks like data manipulation and visualization.

**Python Popular Frameworks and Libraries**

o Mathematics - NumPy, Pandas, etc.

o REST framework: a toolkit for building RESTful APIs

o MachineLearning – Numpy, Seaborn, Matplotlib etc.

**Where is Python used?**

o Data Science: Python is important in this field because it is easy to use and has powerful tools for data analysis and visualization like NumPy, Pandas, and Matplotlib.

o Machine Learning: Python is widely used for machine learning due to its simplicity, ease of use, and availability of powerful machine learning libraries.

**Introduction to Machine Learning**

Machine learning (ML) is a subfield of artificial intelligence (AI) that involves the development of algorithms and statistical models enabling computers to perform tasks without explicit instructions. Instead, these systems learn patterns and make decisions based on data. Machine learning is transforming various industries by automating complex processes, providing insights from large datasets, and creating new opportunities for innovation.

**Definition and Scope**

Machine learning leverages computational methods to improve performance on a given task over time with experience. This process involves:

1. **Data Collection:** Gathering large and diverse datasets.

2. **Data Preprocessing:** Cleaning and formatting data to be suitable for analysis.

3. **Model Selection:** Choosing an appropriate algorithm or model based on the task.

4. **Training:** Feeding the data into the model to learn patterns.

5. **Evaluation:** Assessing the model's performance using metrics and validation techniques.

6. **Deployment:** Implementing the model in real-world applications.

7. **Maintenance:** Continuously updating and refining the model as new data becomes available.

**Types of Machine Learning**

Machine learning techniques can be broadly categorized into three types:

**1. Supervised Learning:** The model is trained on a labeled dataset, meaning that each training example is paired with an output label. Common algorithms include:

o Linear Regression

o Decision Trees

o Support Vector Machines (SVM)

o Neural Networks

**2. Unsupervised Learning:** The model is provided with unlabeled data and must find inherent patterns or groupings. Common algorithms include:

o Clustering (e.g., K-Means, Hierarchical Clustering)

o Association Rules (e.g., Apriori, Eclat)

o Principal Component Analysis (PCA)

**3. Reinforcement Learning:** The model learns by interacting with an environment, receiving rewards or penalties based on its actions, and aims to maximize cumulative rewards. Key concepts include:

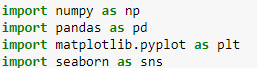
o Markov Decision Processes (MDP)

o Q-Learning

o Deep Q-Networks (DQN)

**IMPORTING DATA**

**. Libraries**

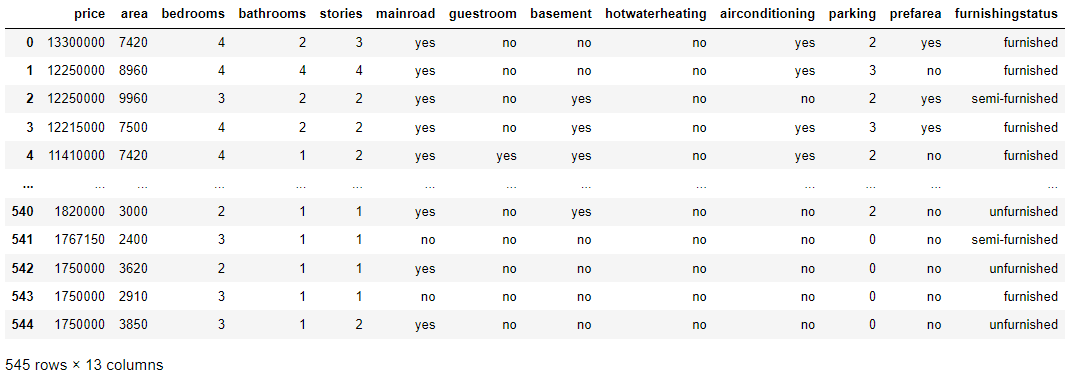
****

Pandas: It provides data structures like Data Frames and Series to handle and analyze data efficiently. NumPy is a library for numerical computing in Python. Seaborn is a statistical data visualization library. Matplotlib is a plotting library for Plotting graphs, histograms, scatter plots, and customizing visualizations.

**. Dataset Read**

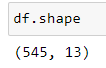
****

pd. read\_csv(' Housing.csv') is a function in Pandas that reads a comma-separated values (CSV) file into a Data Frame df. This function is reading the weather\_classification\_data.



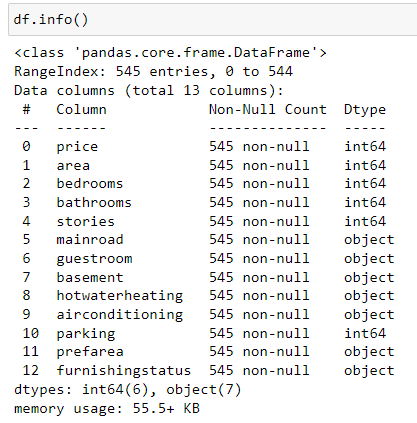
**Gathering Basic information**

**df. shape**



df. shape returns a tuple representing the dimensionality of the Data Frame. This dataset contains 545 rows and 13 columns.

**df.info ()**



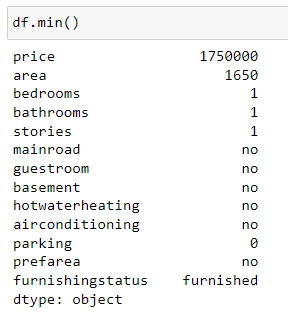
df.info () is a function that returns the count of rows, null status and the datatype of each column. Example The column price has 545 rows each containing non null values and datatype is int64.

**df. count ()**

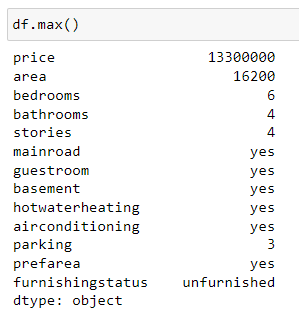


df. count () function is used to return the number of rows in each column in the dataset. Example area has 545 rows of data.

**df.min()**

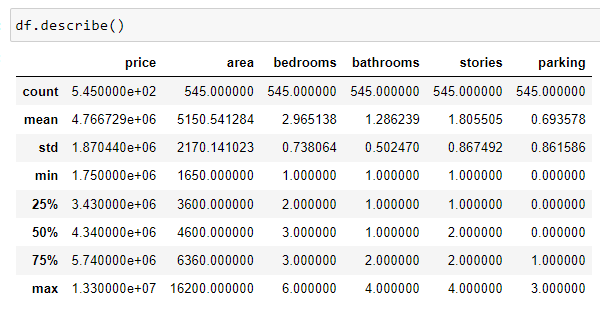


df.min () function is used to return the minimum value from each column. For example, area has 1650 as the minimum value in the entire column.



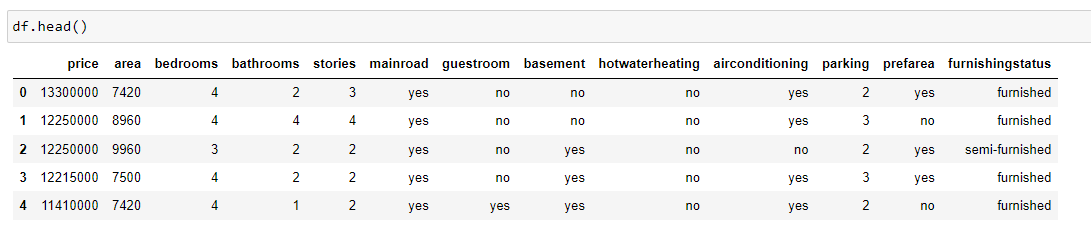
df.max () function is used to return the maximum value from each column. For example, area has 16200 as the maximum value.

**df.describe()**



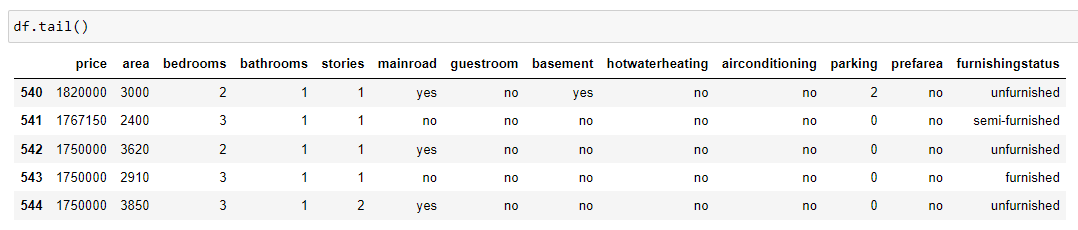
df. describe () function returns the Measure of Central Tendency and Five Point Summary.

**df.head()**

****

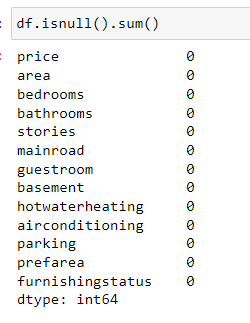
df. head () function returns the top 5 rows of the dataset.

**df. tail ()**



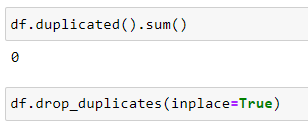
df. tail () function returns the bottom 5 rows of the dataset.

**df.isnull().sum()**



df.isnull().sum() is a chain function formed using isnull() and sum() functions it shows the sum of null values in a column.

**df. duplicated().sum()**



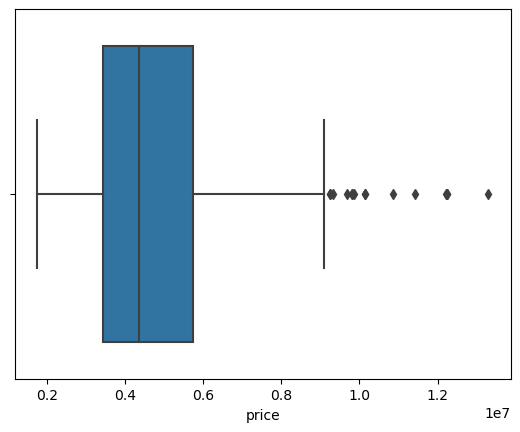
df. duplicated().sum() function returns the total duplicate values in the dataset. This dataset has 0 duplicate values.

df.drop\_duplicates(inplace=True) function drop all the duplicate values.

**Identification of Outliers using Box Plot and Outliers**

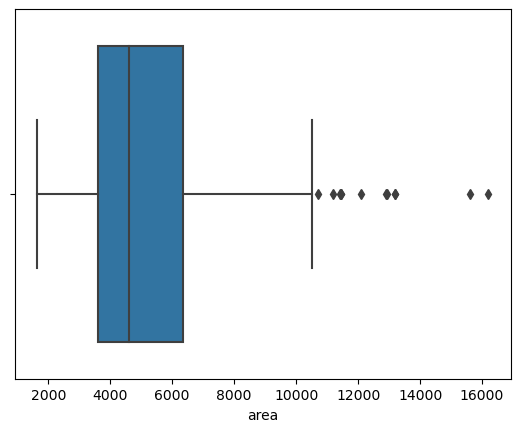
****

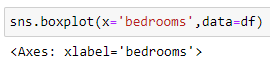
→Figure represents the Boxplot for price.



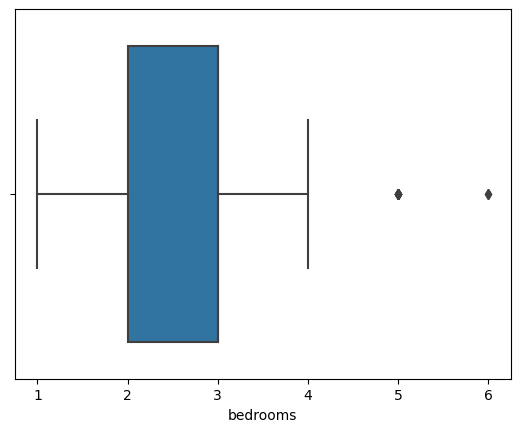


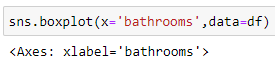
→ Figure represents the Boxplot for area.



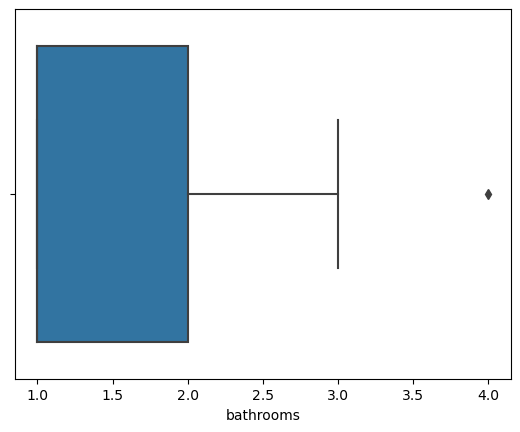


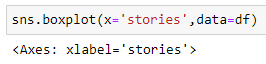
→Figure represents the Boxplot for bedrooms



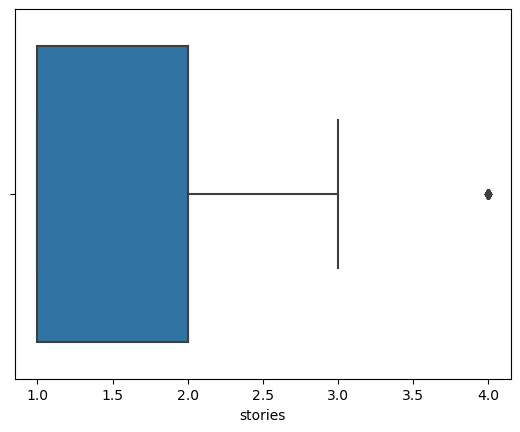


→Figure represents the Boxplot for bathrooms

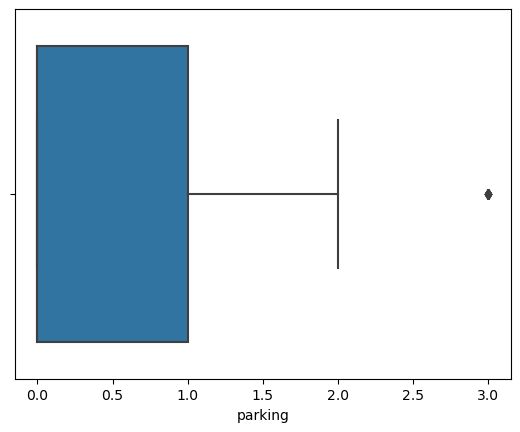




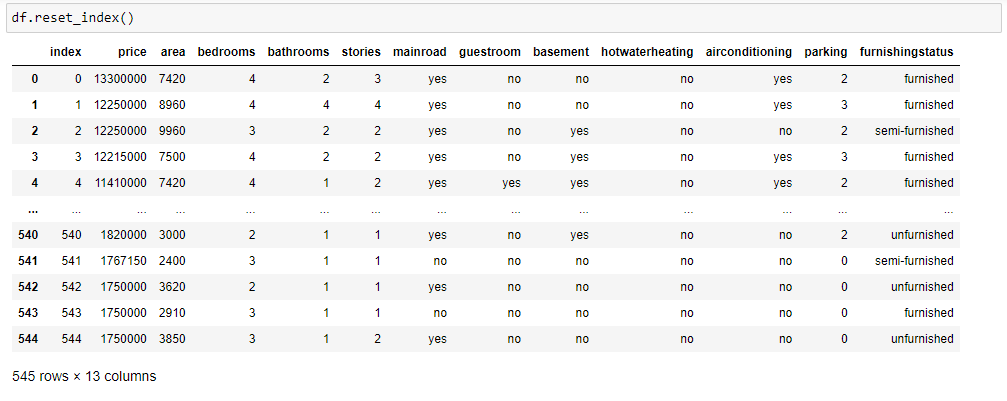
→ Figure represents the Boxplot for stories.





→ Figure represents the Boxplot for parking.

**RESETING INDEX AFTER REMOVING OUTLIERS**

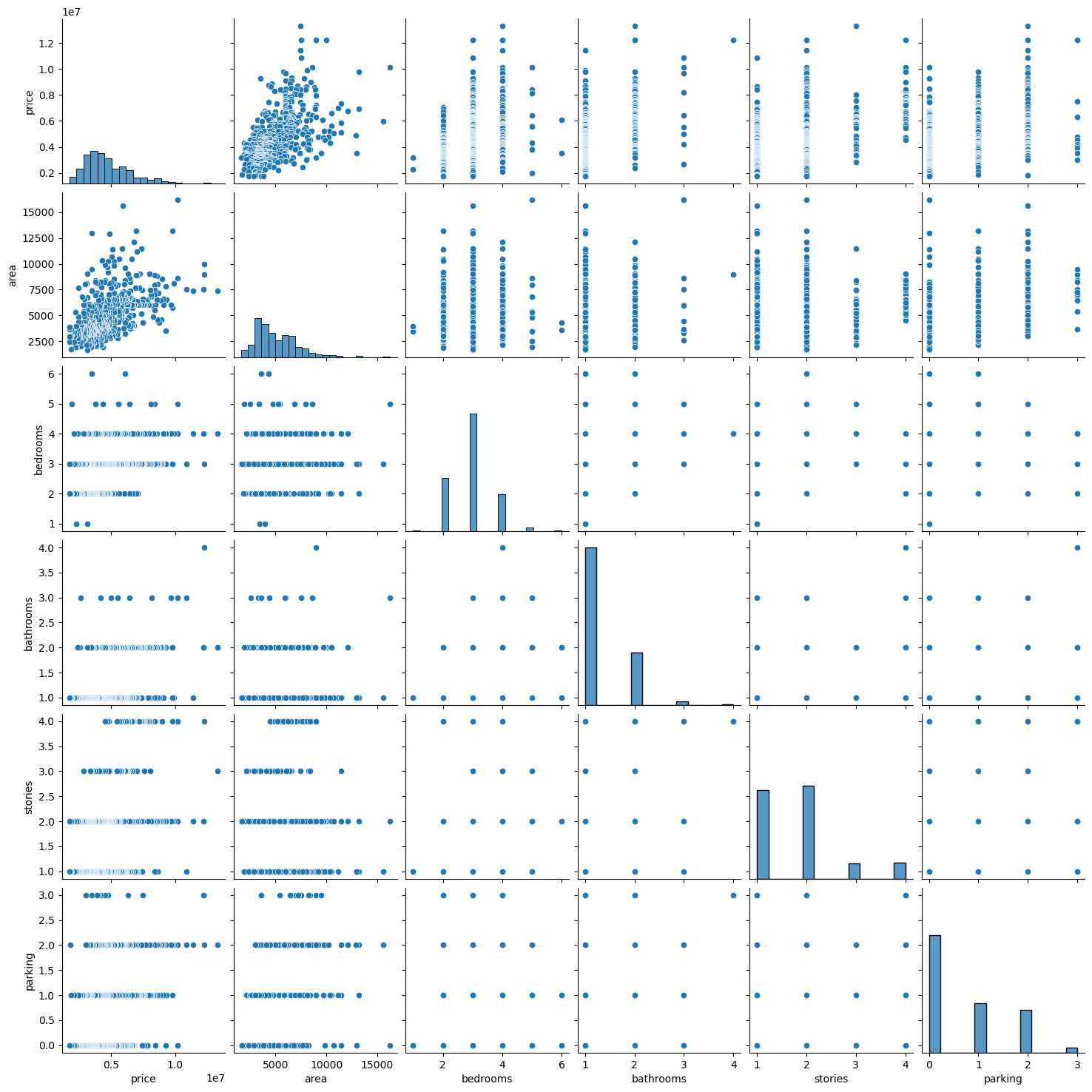
****

Outliers are data points that deviate significantly from the rest of the dataset. They can arise due to measurement errors, data entry errors, or inherent variability in the data. Outliers can skew the analysis results, leading to inaccurate conclusions. Therefore, it is essential to identify and handle outliers before performing statistical analysis.

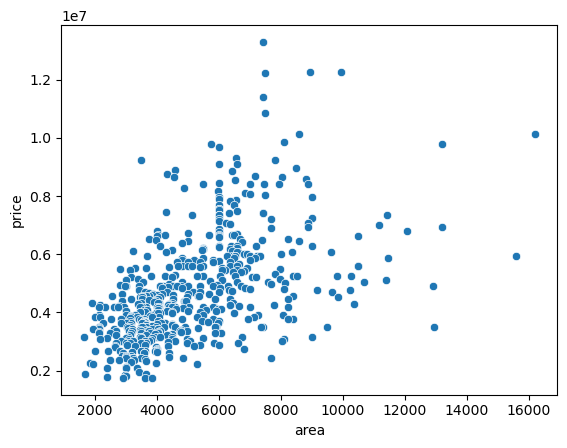
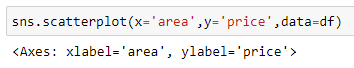
**Dataset Trend Visualization**

**PAIRPLOTS**

****

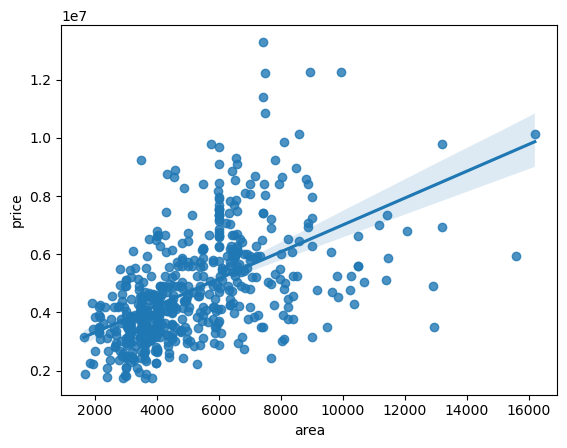
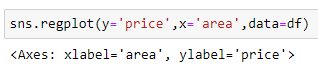
Pair plots are particularly useful in the context of outlier detection and data preprocessing. They provide a clear visual representation of how each variable interacts with others, making it easier to spot anomalies that do not follow the general pattern of the data.

**SCATTER PLOT**

****

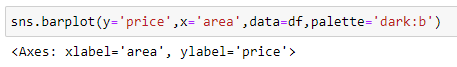
The image shows a scatter plot generated using the sns.scatterplot() function from the Seaborn library. Scatter plots are useful for visualizing the relationship between two continuous variables. They help identify trends, patterns, and potential outliers in the data.

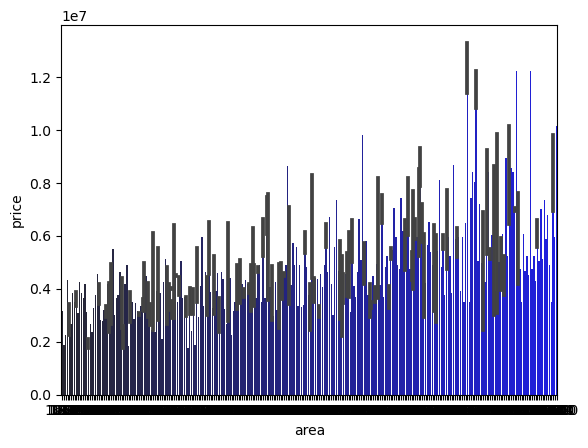
**REGRESSION PLOT**

****

A regression plot is a graphical representation of the relationship between two or more variables, typically used to show how a dependent variable changes as an independent variable changes. Here’s a step-by-step explanation of how to create a regression plot.

**BAR PLOT**

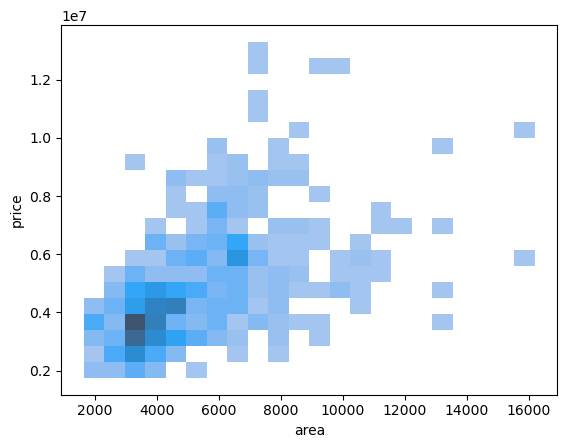
****

****

A bar plot (or bar chart) is a graphical display of data using bars of different heights. It is commonly used to compare quantities across different categories. Here’s an explanation of how to create and interpret a bar plot.

**HISTOGRAM**

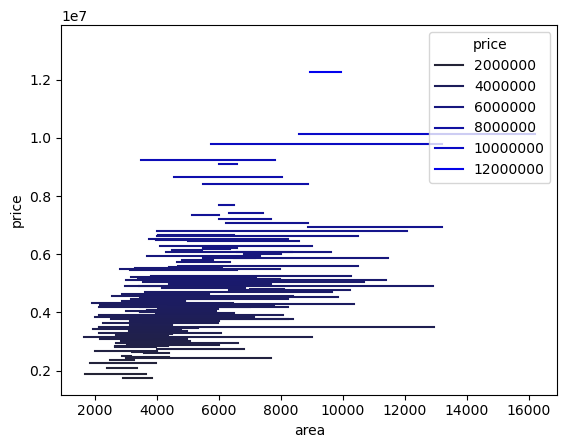
****

****

A histogram is a type of bar chart that represents the distribution of a dataset. It is used to show the frequency (or count) of data points that fall within specified ranges (bins). Here’s a step-by-step explanation of how to create and interpret a histogram.

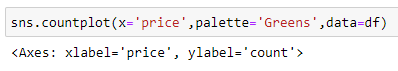
**LINE PLOT**

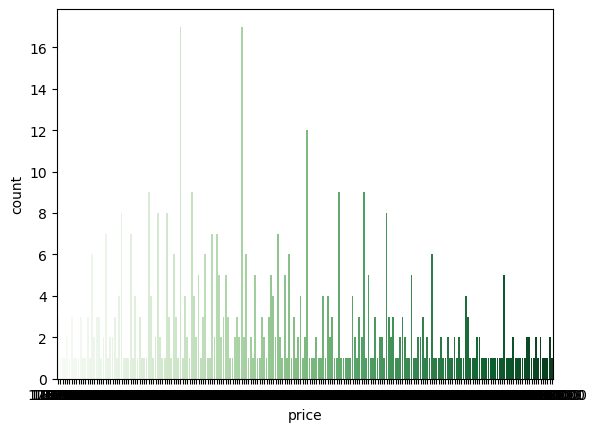
****

****

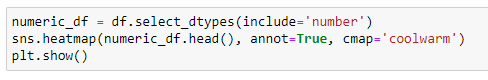
A line plot (or line chart) is a type of chart used to display information as a series of data points called 'markers' connected by straight line segments. It is commonly used to visualize trends over time. Here’s a step-by-step explanation of how to create and interpret a line plot.

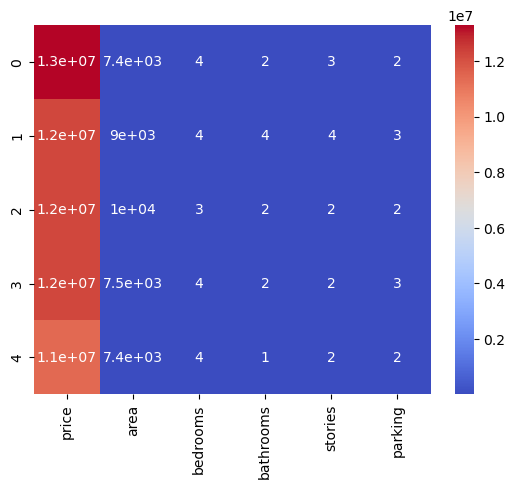
**COUNTER PLOT**

****

****

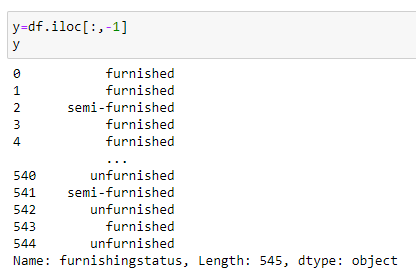
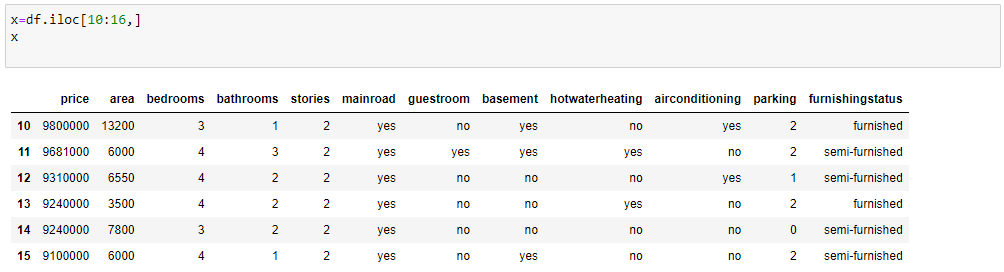
A counter plot, often referred to as a count plot, is used to visualize the count of observations in each category of a categorical variable. It is particularly useful for understanding the distribution of categorical data and comparing the frequencies of different categories.

**CORRELATION HEAMAP**

****

The image shows a correlation heatmap generated using the sns.heatmap() function from the Seaborn library. Correlation heatmaps are useful for visualizing the strength and direction of relationships between pairs of variables in a dataset.

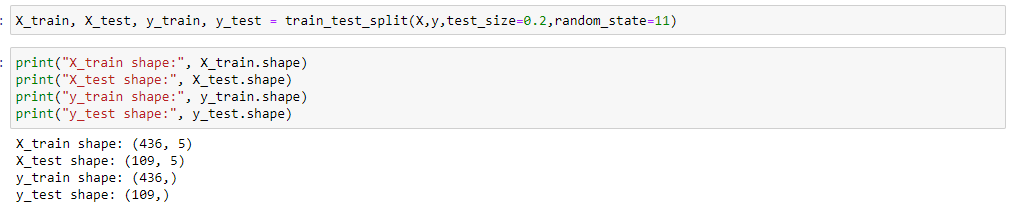
**Test and Train Splitting**

****

The x and y variables are separated into independent and dependent values using `iloc`. Columns from index 10 to 16 (excluding 16) are assigned to x, while the last column, which represents the weather type, is assigned to y as the dependent variable.

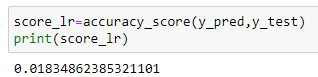
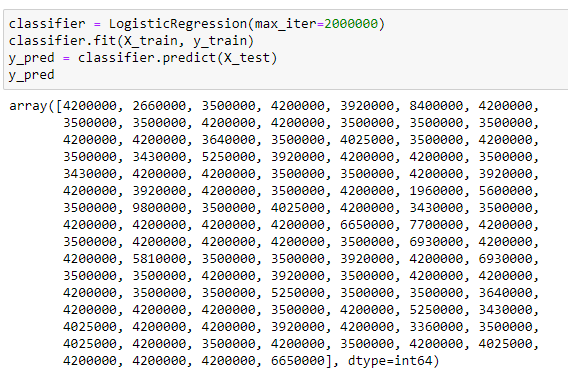
**LOGISTIC REGRESSOR**

****

****

The x and y variables are further divided into `x\_train`, `y\_train`, `x\_test`, and `y\_test`. The `x\_train` and `y\_train` subsets are used for training the model, while `x\_test` and `y\_test` are used for evaluating the model. The test size is set to 0.2, meaning 20% of the dataset will be used for testing. The random state is set to 11 to ensure reproducibility by controlling the selection of training rows.

**ACCURACY AND PREDICION**

****

**Prediction:** The act of using a model to forecast the outcomes based on input data.

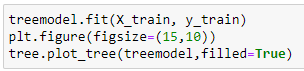
**Accuracy:** A measure of how many of those predictions were correct, often used as a performance metric for classification models.

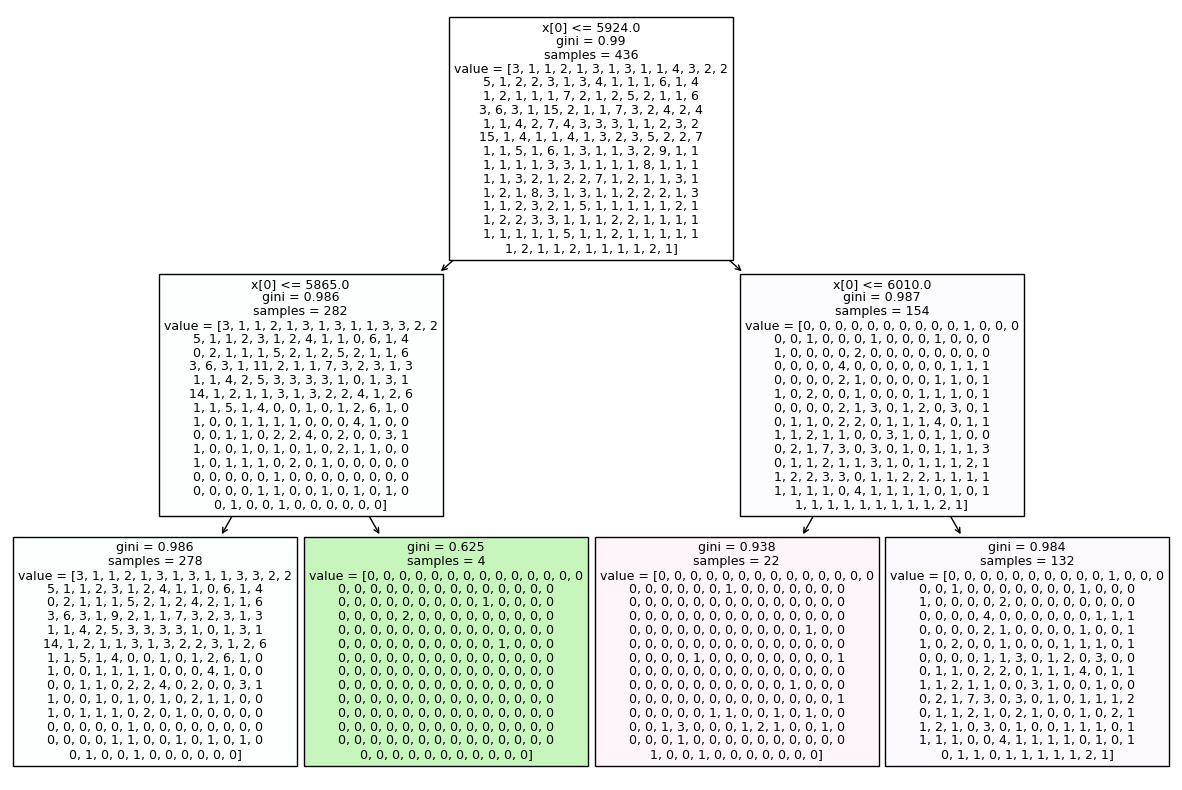
**DECISION TREE**

****

****

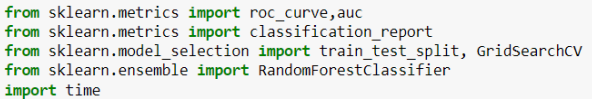
`DecisionTreeClassifier` is imported from `sklearn` and assigned to the variable `treemodel`, with a maximum depth of 2. The model is then trained using `x\_train` and `y\_train`.



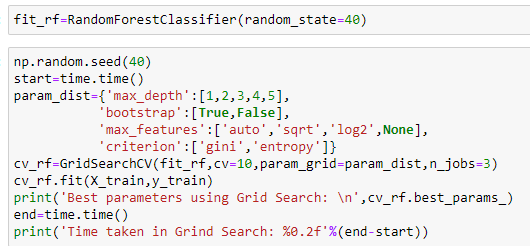
****

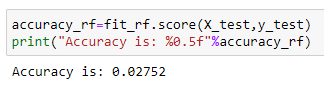
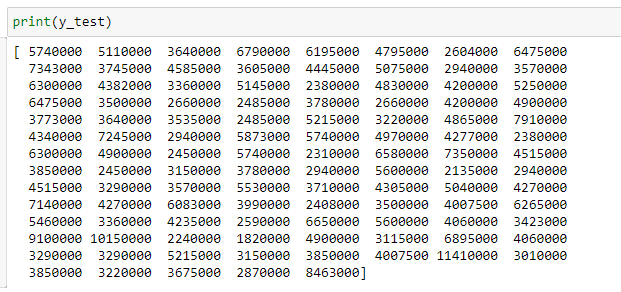
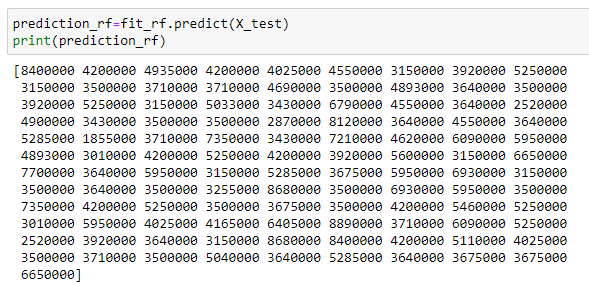
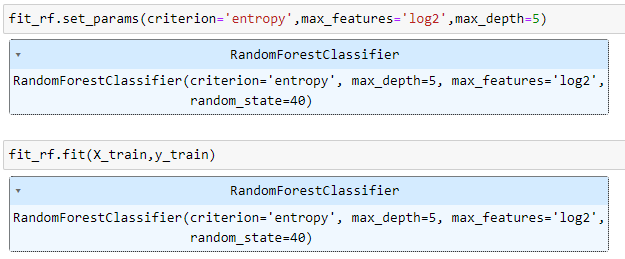
The max depth with value 2 graph is plotted using decision tree model. A tree plot visually represents the structure of a decision tree, illustrating how decisions are made based on feature values. It shows the tree's nodes, branches, and leaves, detailing the splits and outcomes at each node.

**RANDOM FOREST**

****

The following libraries are imported: `roc\_curve`, `auc`, `classification\_report`, `GridSearchCV`, and `RandomForestClassifier`. The `time` library is also imported. A `RandomForestClassifier` model.



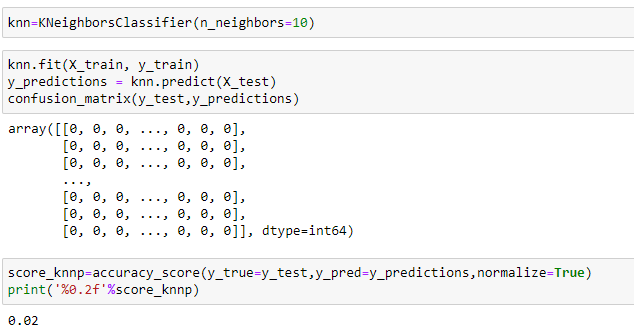
Parameters like `max\_depth`, `bootstrap`, `max\_features`, and `criterion` are set to optimize the accuracy of the dataset. The best parameters are determined using `GridSearchCV`. The `cv\_rf` (a `GridSearchCV` instance) is used to fit the model on `x\_train` and `y\_train`.

The values of `x\_test` are used to generate predictions, and these predictions are compared with `y\_test` to calculate the accuracy.

**K - Nearest Neighbor**

****

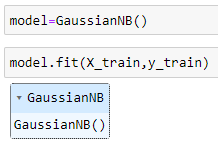
The `KNeighborsClassifier` is imported and instantiated with `n\_neighbors` set to 10, assigning it to the variable `knn`.

****

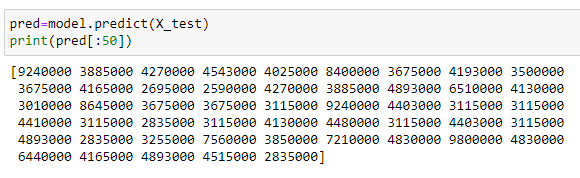
The `confusion\_matrix` library is imported, and a confusion matrix is generated using `y\_test` and `y\_prediction` to identify and analyse the errors in the model's predictions. The values of `y\_test` and `y\_prediction` are compared to determine the model's accuracy.

**Gaussian Naive Bayes**

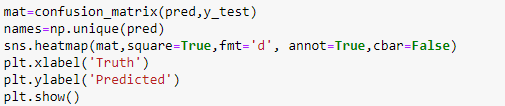
****

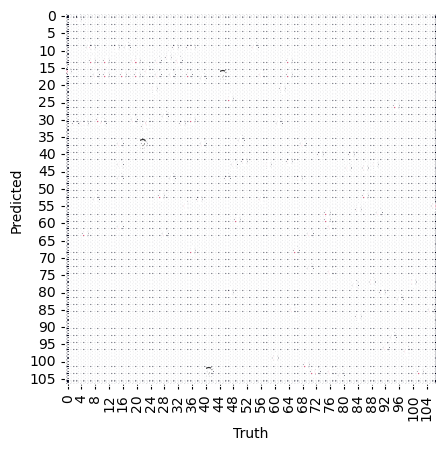
****

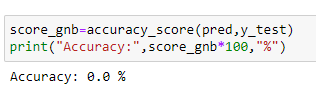
The `GaussianNB` library is imported from `sklearn`, and the model is fitted using `x\_train` and `y\_train`.



The predicted value of x\_test is assigned to a variable named as pred.

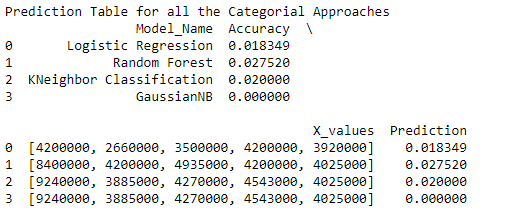
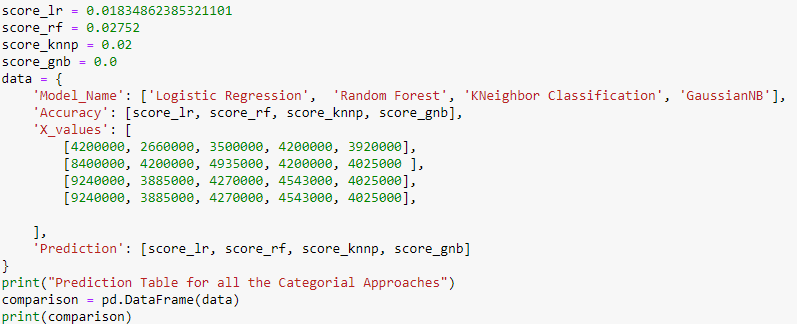


A heatmap for true and predicted values visualizes the performance of a classification model by showing how often each combination of actual and predicted classes occurs. It helps in understanding the distribution of prediction errors and correct classifications.



Using the `GaussianNB` model, the predicted values (`y\_pred`) are compared with the actual values (`y\_test`).

**Data Comparison**

****

A table summarizing the different algorithms and their corresponding accuracies.

**END OF REPORT**