



```
import pandas as pd
import csv
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import MinMaxScaler,StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error,
    r2_score,mean_absolute_error
import numpy as np
from sklearn.model_selection import train_test_split
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
pd.read_csv('/content/drive/MyDrive/ML Lab/Housing.csv')
```

<div>

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basemen
0	13300000	7420	4	2	3	yes	no	no
1	12250000	8960	4	4	4	yes	no	no
2	12250000	9960	3	2	2	yes	no	yes
3	12215000	7500	4	2	2	yes	no	yes
4	11410000	7420	4	1	2	yes	yes	yes
...
540	1820000	3000	2	1	1	yes	no	yes
541	1767150	2400	3	1	1	no	no	no
542	1750000	3620	2	1	1	yes	no	no
543	1750000	2910	3	1	1	no	no	no
544	1750000	3850	3	1	2	yes	no	no

<div class="colab-df-buttons">

<button class="colab-df-convert" onclick="convertToInteractive('df-da81fda5-a48f-4577-934d-eeeba599479a')"

title="Convert this dataframe to an interactive table."

<script>

const buttonEl = document.querySelector('#df-da81fda5-eeeba599479a button.colab-df-convert');

buttonEl.style.display = google.colab.kernel.accessAllowed ? ''

```
style="display:none;">
```

```
    async function convertToInteractive(key) {
      const element = document.querySelector(
        `#df-e773778e-287b-45b9-973a-41757fba4bc2`);
      const dataTable =
        await
        google.colab.kernel.invokeFunction('convertToInteractive',
        [key], {});
      if (!dataTable) return;

      const docLinkHtml = 'Like what you see?<br>
        <a href="https://colab.research.google.com/notebooks/quickchart.ipynb" target="_blank">Click here</a>
        to learn more about interactive charts.';
      element.innerHTML = docLinkHtml;
      dataTable['output_type'] = 'display_data';
      await google.colab.output.renderOutput(dataTable, element);
      const docLink = document.createElement('a');
      docLink.innerHTML = docLinkHtml;
      element.appendChild(docLink);
    }
  }
</script>
```

```
<div id="df-e773778e-287b-45b9-973a-41757fba4bc2">
  <button class="colab-df-quickchart" onclick="quickchart('df-
e773778e-287b-45b9-973a-41757fba4bc2')"
    title="Suggest charts"
    style="display:none;">
```



```
</button>
```

```
<script>
  async function quickchart(key) {
    const quickchartButtonEl =
      document.querySelector('#' + key + ' button');
    quickchartButtonEl.disabled = true; // To prevent multiple
    clicks.
    quickchartButtonEl.classList.add('colab-df-spinner');
    try {
      const charts = await google.colab.kernel.invokeFunction(
        'suggestCharts', [key], {});
    } catch (error) {
```

```

        console.error('Error during call to suggestCharts:', error);
    }
    quickchartButtonEl.classList.remove('colab-df-spinner');
    quickchartButtonEl.classList.add('colab-df-quickchart-
complete');
}
(() => {
    let quickchartButtonEl =
        document.querySelector('#df-
e773778e-287b-45b9-973a-41757fba4bc2 button');
    quickchartButtonEl.style.display =
        google.colab.kernel.accessAllowed ? 'block' : 'none';
})();
</script>
</div>

</div>

```

```
housing_df=pd.read_csv('/content/drive/MyDrive/ML Lab/Housing.csv')
```

```
housing_df.shape
```

```
(545, 12)
```

```
housing_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 545 entries, 0 to 544
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   price                 545 non-null   int64
1   area                  545 non-null   int64
2   bedrooms              545 non-null   int64
3   bathrooms              545 non-null   int64
4   stories                545 non-null   int64
5   mainroad               545 non-null   object
6   guestroom              545 non-null   object
7   basement               545 non-null   object
8   airconditioning        545 non-null   object
9   parking                545 non-null   int64
10  prefarea               545 non-null   object
11  furnishingstatus       545 non-null   object
dtypes: int64(6), object(6)
memory usage: 51.2+ KB

```


```
housing_df.describe()
```

<div>

	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

<div class="colab-df-buttons">

<button class="colab-df-convert"
onclick="convertToInteractive('df-
f555fabe-00ab-40c1-be57-55486280c5ee')"
title="Convert this dataframe
to an interactive table."
style="display:none;">

 <script>
const buttonEl =
document.querySelector('#df-
f555fabe-00ab-40c1-be57-55486280c5ee
convert');
buttonEl.style.display =
google.colab.kernel.accessAllowed

async function convertToInteractive
const element = document.querySe
f555fabe-00ab-40c1-be57-55486280c5ee
const dataTable =
await
google.colab.kernel.invokeFunction(''
{});
if (!dataTable) return;

const docLinkHtml = 'Like what y
<a target="_blank"
href=https://colab.research.google.c
data_table.ipynb>data table notebook
+ ' to learn more about intera
element.innerHTML = '';
dataTable['output_type'] = 'disp
await google.colab.output.render
element);
const docLink = document.createE
docLink.innerHTML = docLinkHtml;
element.appendChild(docLink);
}
</script>

```

<div id="df-2b088d96-d904-4e7e-89e5-7ee4d7e65fcf">
  <button class="colab-df-quickchart"
onclick="quickchart('df-2b088d96-d904-4e7e-89e5-7ee4d7e65fcf') "
        title="Suggest charts"
        style="display:none;">

```



```

</button>

```

```

<script>
  async function quickchart(key) {
    const quickchartButtonEl =
      document.querySelector('#' + key + ' button');
    quickchartButtonEl.disabled = true;  // To prevent multiple
clicks.
    quickchartButtonEl.classList.add('colab-df-spinner');
    try {
      const charts = await google.colab.kernel.invokeFunction(
        'suggestCharts', [key], {});
    } catch (error) {
      console.error('Error during call to suggestCharts:', error);
    }
    quickchartButtonEl.classList.remove('colab-df-spinner');
    quickchartButtonEl.classList.add('colab-df-quickchart-
complete');
  }
  (() => {
    let quickchartButtonEl =

document.querySelector('#df-2b088d96-d904-4e7e-89e5-7ee4d7e65fcf
button');
    quickchartButtonEl.style.display =
      google.colab.kernel.accessAllowed ? 'block' : 'none';
  })();
</script>
</div>

```

```

</div>

```

```

housing_df.isnull().sum()

```

	0
price	0
area	0
bedrooms	0
bathrooms	0

	0
stories	0
mainroad	0
guestroom	0
basement	0
hotwaterheating	0
airconditioning	0
parking	0
prefarea	0
furnishingstatus	0

dtype: int64

housing_df.nunique()

	0
price	219
area	284
bedrooms	6
bathrooms	4
stories	4
mainroad	2
guestroom	2
basement	2
hotwaterheating	2
airconditioning	2
parking	4
prefarea	2
furnishingstatus	3

dtype: int64

```
!pip install ydata_profiling
```

Collecting ydata_profiling

Downloading ydata_profiling-4.16.1-py2.py3-none-any.whl.metadata
(22 kB)

Collecting scipy<1.16,>=1.4.1 (from ydata_profiling)

Downloading
scipy-1.15.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(61 kB)

◆[2K ◆[90m _____ ◆[0m
◆[32m62.0/62.0 kB◆[0m ◆[31m718.8 kB/s◆[0m eta ◆[36m0:00:00◆[0m
◆[?25hRequirement already satisfied: pandas!=1.4.0,<3.0,>1.1 in /usr/
local/lib/python3.11/dist-packages (from ydata_profiling) (2.2.2)
Requirement already satisfied: matplotlib<=3.10,>=3.5 in /usr/local/
lib/python3.11/dist-packages (from ydata_profiling) (3.10.0)
Requirement already satisfied: pydantic>=2 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (2.11.7)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (6.0.2)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (3.1.6)
Collecting visions<0.8.2,>=0.7.5 (from
visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling)
 Downloading visions-0.8.1-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.2,>=1.16.0 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (2.0.2)
Collecting htmlmin==0.1.12 (from ydata_profiling)
 Downloading htmlmin-0.1.12.tar.gz (19 kB)
 Preparing metadata (setup.py) ... ◆[?25l◆[?25hdone
Collecting phik<0.13,>=0.11.1 (from ydata_profiling)
 Downloading
phik-0.12.5-cp311-cp311-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl.metadata
(5.6 kB)
Requirement already satisfied: requests<3,>=2.24.0 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (2.32.3)
Requirement already satisfied: tqdm<5,>=4.48.2 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (4.67.1)
Requirement already satisfied: seaborn<0.14,>=0.10.1 in /usr/local/
lib/python3.11/dist-packages (from ydata_profiling) (0.13.2)
Collecting multimethod<2,>=1.4 (from ydata_profiling)
 Downloading multimethod-1.12-py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in /usr/local/
lib/python3.11/dist-packages (from ydata_profiling) (0.14.5)
Requirement already satisfied: typeguard<5,>=3 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (4.4.4)
Collecting imagehash==4.3.1 (from ydata_profiling)
 Downloading ImageHash-4.3.1-py2.py3-none-any.whl.metadata (8.0 kB)
Requirement already satisfied: wordcloud>=1.9.3 in /usr/local/lib/
python3.11/dist-packages (from ydata_profiling) (1.9.4)
Collecting dacite>=1.8 (from ydata_profiling)
 Downloading dacite-1.9.2-py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: numba<=0.61,>=0.56.0 in /usr/local/
lib/python3.11/dist-packages (from ydata_profiling) (0.60.0)
Requirement already satisfied: PyWavelets in /usr/local/lib/
python3.11/dist-packages (from imagehash==4.3.1->ydata_profiling)
(1.8.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/

dist-packages (from imagehash==4.3.1->ydata_profiling) (11.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/
python3.11/dist-packages (from jinja2<3.2,>=2.11.1->ydata_profiling)
(3.0.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/
python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/
python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/
python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (4.59.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/
python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/
python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/
python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/
lib/python3.11/dist-packages (from
matplotlib<=3.10,>=3.5->ydata_profiling) (2.9.0.post0)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/
local/lib/python3.11/dist-packages (from
numba<=0.61,>=0.56.0->ydata_profiling) (0.43.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/
python3.11/dist-packages (from
pandas!=1.4.0,<3.0,>1.1->ydata_profiling) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/
python3.11/dist-packages (from
pandas!=1.4.0,<3.0,>1.1->ydata_profiling) (2025.2)
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/
python3.11/dist-packages (from phik<0.13,>=0.11.1->ydata_profiling)
(1.5.1)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/
lib/python3.11/dist-packages (from pydantic>=2->ydata_profiling)
(0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/
lib/python3.11/dist-packages (from pydantic>=2->ydata_profiling)
(2.33.2)
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/
local/lib/python3.11/dist-packages (from
pydantic>=2->ydata_profiling) (4.14.1)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/
local/lib/python3.11/dist-packages (from


```
pydantic>=2->ydata_profiling) (0.4.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/
local/lib/python3.11/dist-packages (from
requests<3,>=2.24.0->ydata_profiling) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/
python3.11/dist-packages (from requests<3,>=2.24.0->ydata_profiling)
(3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/
python3.11/dist-packages (from requests<3,>=2.24.0->ydata_profiling)
(2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/
python3.11/dist-packages (from requests<3,>=2.24.0->ydata_profiling)
(2025.7.14)
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/
python3.11/dist-packages (from
statsmodels<1,>=0.13.2->ydata_profiling) (1.0.1)
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/
python3.11/dist-packages (from
visions<0.8.2,>=0.7.5->visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling)
(25.3.0)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/
python3.11/dist-packages (from
visions<0.8.2,>=0.7.5->visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling)
(3.5)
Collecting puremagic (from
visions<0.8.2,>=0.7.5->visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling)
  Downloading puremagic-1.30-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/
dist-packages (from python-
dateutil>=2.7->matplotlib<=3.10,>=3.5->ydata_profiling) (1.17.0)
Downloading ydata_profiling-4.16.1-py2.py3-none-any.whl (400 kB)
[2K] [90m-----[0m
[32m400.1/400.1 kB[0m [31m2.8 MB/s[0m eta [36m0:00:00[0m
[?25hDownloading ImageHash-4.3.1-py2.py3-none-any.whl (296 kB)
[2K] [90m-----[0m
[32m296.5/296.5 kB[0m [31m11.8 MB/s[0m eta [36m0:00:00[0m
[?25hDownloading dacite-1.9.2-py3-none-any.whl (16 kB)
Downloading multimethod-1.12-py3-none-any.whl (10 kB)
Downloading
phik-0.12.5-cp311-cp311-manylinux_2_24_x86_64.manylinux_2_28_x86_64.whl
(679 kB)
[2K] [90m-----[0m
[32m679.0/679.0 kB[0m [31m9.3 MB/s[0m eta [36m0:00:00[0m
[?25hDownloading
scipy-1.15.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(37.7 MB)
[2K] [90m-----[0m [32m37.7/
37.7 MB[0m [31m32.7 MB/s[0m eta [36m0:00:00[0m
```

```

[?25hDownloading visions-0.8.1-py3-none-any.whl (105 kB)
[2K  [90m-----[0m
[32m105.4/105.4 kB[0m [31m9.7 MB/s[0m eta [36m0:00:00[0m
[?25hDownloading puremagic-1.30-py3-none-any.whl (43 kB)
[2K  [90m-----[0m [32m43.3/
43.3 kB[0m [31m3.0 MB/s[0m eta [36m0:00:00[0m
[?25hBuilding wheels for collected packages: htmlmin
  Building wheel for htmlmin (setup.py) ... [?25l[?25hdone
  Created wheel for htmlmin: filename=htmlmin-0.1.12-py3-none-any.whl
  size=27081
  sha256=930d3aa70c99dfb0b30b3ec9902c822f8d1e0c75c3b4ef5abbe613f982d457b5
  Stored in directory: /root/.cache/pip/wheels/8d/55/1a/
19cd535375ed1ede0c996405ebffe34b196d78e2d9545723a2
Successfully built htmlmin
Installing collected packages: puremagic, htmlmin, scipy,
multimethod, dacite, imagehash, visions, phik, ydata_profiling
  Attempting uninstall: scipy
    Found existing installation: scipy 1.16.0
    Uninstalling scipy-1.16.0:
      Successfully uninstalled scipy-1.16.0
Successfully installed dacite-1.9.2 htmlmin-0.1.12 imagehash-4.3.1
multimethod-1.12 phik-0.12.5 puremagic-1.30 scipy-1.15.3
visions-0.8.1 ydata_profiling-4.16.1

```

```

from ydata_profiling import ProfileReport
profile=ProfileReport(housing_df,title='Housing Pandas Profiling
Report')
profile.to_file('housing_pandas_profiling.html')

```

```
Summarize dataset: 0%|          | 0/5 [00:00<?, ?it/s]
```

```

0%|          | 0/13 [00:00<?, ?it/s][A
23%|██        | 3/13 [00:00<00:00, 21.26it/s][A
46%|██████    | 6/13 [00:00<00:00, 16.13it/s][A
62%|████████  | 8/13 [00:00<00:00, 15.85it/s][A
77%|█████████ |10/13 [00:00<00:00, 15.33it/s][A
100%|██████████|13/13 [00:00<00:00, 13.83it/s]

```

```
Generate report structure: 0%|          | 0/1 [00:00<?, ?it/s]
```

```
Render HTML: 0%|          | 0/1 [00:00<?, ?it/s]
```

Export report to file: 0%| | 0/1 [00:00<?, ?it/s]

```
advertising_df=pd.read_csv('/content/drive/MyDrive/ML Lab/
advertising.csv')
```

```
profile=ProfileReport(advertising_df,title='advertising Pandas
Profiling Report')
```

```
profile.to_file('advertising_pandas_profiling.html')
```

Summarize dataset: 0%| | 0/5 [00:00<?, ?it/s]

100%|██████████| 4/4 [00:00<00:00, 23663.21it/s]

Generate report structure: 0%| | 0/1 [00:00<?, ?it/s]

Render HTML: 0%| | 0/1 [00:00<?, ?it/s]

Export report to file: 0%| | 0/1 [00:00<?, ?it/s]

```
# how water heating is highly imabalanced so it is removed
housing_df.drop('hotwaterheating',axis=1,inplace=True)
```

```
housing_df.shape
```

```
(545, 12)
```

```
numerical_data=housing_df.select_dtypes(include='number')
```

```
categorical_data=housing_df.select_dtypes(exclude='number')
```

```
print(categorical_data)
```

	mainroad	guestroom	basement	airconditioning	prefarea
furnishingstatus					
0	yes	no	no	yes	yes
furnished					
1	yes	no	no	yes	no
furnished					
2	yes	no	yes	no	yes
					semi-

```

furnished
3      yes      no      yes      yes      yes
furnished
4      yes      yes      yes      yes      no
furnished
..      ...      ...      ...      ...      ...
540     yes      no      yes      no      no
unfurnished
541     no      no      no      no      no      semi-
furnished
542     yes      no      no      no      no
unfurnished
543     no      no      no      no      no
furnished
544     yes      no      no      no      no
unfurnished

```

```
[545 rows x 6 columns]
```

```

categorical_columns =
    housing_df.select_dtypes(include='object').columns.tolist()
print(categorical_columns)

```

```

['mainroad', 'guestroom', 'basement', 'airconditioning', 'prefarea',
'furnishingstatus']

```

```
encoded_df = pd.DataFrame(index=housing_df.index)
```

```

for col in categorical_data.columns:
    unique_values = categorical_data[col].unique()
    for val in unique_values:
        encoded_df[f"{col}_{val}"] = (categorical_data[col] ==
        val).astype(int)

```

```
encoded_df = pd.concat([encoded_df, numerical_data], axis=1)
```

```
encoded_df
```

<div>

	mainroad_yes	mainroad_no	guestroom_no	guestroom_yes	basement_no	ba
0	1	0	1	0	1	0
1	1	0	1	0	1	0
2	1	0	1	0	0	1
3	1	0	1	0	0	1
4	1	0	0	1	0	1

	mainroad_yes	mainroad_no	guestroom_no	guestroom_yes	basement_no	ba
...
540	1	0	1	0	0	1
541	0	1	1	0	1	0
542	1	0	1	0	1	0
543	0	1	1	0	1	0
544	1	0	1	0	1	0

```
<div class="colab-df-buttons">
```

```
<button class="colab-df-convert"
onclick="convertToInteractive('df-
ad4ea839-ae21-4455-8e0c-165986047ffd') "
  title="Convert this dataframe
to an interactive table."
  style="display:none;">
```

```
<script>
  const buttonEl =
    document.querySelector('#df-
ad4ea839-ae21-4455-8e0c-165986047ffd
convert');
  buttonEl.style.display =
    google.colab.kernel.accessAllowed

  async function convertToInteractive() {
    const element = document.querySelector(
      '#df-ad4ea839-ae21-4455-8e0c-165986047ffd
      convert');
    const dataTable =
      await
      google.colab.kernel.invokeFunction('
      convertToInteractive', [element], {});
    if (!dataTable) return;

    const docLinkHtml = 'Like what you saw?
    <a target="_blank"
      href="https://colab.research.google.com/
      ' + data_table.ipynb + '>data table notebook
      ' + ' to learn more about interactive
      data tables.';
    element.innerHTML = '';
    dataTable['output_type'] = 'display_data';
    await google.colab.output.renderOutput(
      dataTable, element);
    const docLink = document.createElement('a');
    docLink.innerHTML = docLinkHtml;
    element.appendChild(docLink);
  }
</script>
```

```
<div id="df-5338734f-1c54-44b4-8366-57297aa46ac4">
```

```

<button class="colab-df-quickchart"
onclick="quickchart('df-5338734f-1c54-44b4-8366-57297aa46ac4') "
        title="Suggest charts"
        style="display:none;">

```



```

</button>

```

```

<script>
  async function quickchart(key) {
    const quickchartButtonEl =
      document.querySelector('#' + key + ' button');
    quickchartButtonEl.disabled = true; // To prevent multiple
clicks.
    quickchartButtonEl.classList.add('colab-df-spinner');
    try {
      const charts = await google.colab.kernel.invokeFunction(
        'suggestCharts', [key], {});
    } catch (error) {
      console.error('Error during call to suggestCharts:', error);
    }
    quickchartButtonEl.classList.remove('colab-df-spinner');
    quickchartButtonEl.classList.add('colab-df-quickchart-
complete');
  }
  (() => {
    let quickchartButtonEl =

document.querySelector('#df-5338734f-1c54-44b4-8366-57297aa46ac4
button');
    quickchartButtonEl.style.display =
      google.colab.kernel.accessAllowed ? 'block' : 'none';
  })();
</script>
</div>

```

```

<style>
.colab-df-generate {
  background-color: #E8F0FE;
  border: none;
  border-radius: 50%;
  cursor: pointer;
  display: none;
  fill: #1967D2;
  height: 32px;
  padding: 0 0 0 0;
  width: 32px;

```

```

}

.colab-df-generate:hover {
  background-color: #E2EBFA;
  box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
rgba(60, 64, 67, 0.15);
  fill: #174EA6;
}

[theme=dark] .colab-df-generate {
  background-color: #3B4455;
  fill: #D2E3FC;
}

[theme=dark] .colab-df-generate:hover {
  background-color: #434B5C;
  box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);
  filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));
  fill: #FFFFFF;
}
</style>
<button class="colab-df-generate"
onclick="generateWithVariable('encoded_df')"
title="Generate code using this dataframe."
style="display:none;">

```



</div>

encoded_df.shape

(545, 19)

X=encoded_df.drop('price',axis=1)

y=encoded_df['price']

y.shape

(545,)

X.shape

(545, 18)

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=

print(f"Shape of X_train: {X_train.shape}")

```

print(f"Shape of y_train: {y_train.shape}")
print(f"Shape of X_test: {X_test.shape}")
print(f"Shape of y_test: {y_test.shape}")

Shape of X_train: (436, 18)
Shape of y_train: (436,)
Shape of X_test: (109, 18)
Shape of y_test: (109,)

X_mean = X_train.mean()
X_std = X_train.std(ddof=0)
X_train_scaled = (X_train - X_mean) / X_std
X_test_scaled = (X_test - X_mean) / X_std
y_mean = y_train.mean()
y_std = y_train.std(ddof=0)
y_train_scaled = ((y_train - y_mean) / y_std).values.reshape(-1, 1)
y_test_scaled = ((y_test - y_mean) / y_std).values.reshape(-1, 1)

print("X_train_scaled:\n", X_train_scaled)
print("X_test_scaled:\n", X_test_scaled)
print("y_train_scaled:\n", y_train_scaled)
print("y_test_scaled:\n", y_test_scaled)

```

```

X_train_scaled:
      mainroad_yes  mainroad_no  guestroom_no  guestroom_yes
basement_no \
46      0.407155    -0.407155      0.466773    -0.466773
0.746420
93      0.407155    -0.407155      0.466773    -0.466773
-1.339728
335     0.407155    -0.407155      0.466773    -0.466773
-1.339728
412     0.407155    -0.407155      0.466773    -0.466773
-1.339728
471     0.407155    -0.407155      0.466773    -0.466773
0.746420
..          ...          ...          ...          ...
...
71      0.407155    -0.407155      0.466773    -0.466773
0.746420
106     0.407155    -0.407155      0.466773    -0.466773
-1.339728
270     0.407155    -0.407155      0.466773    -0.466773
0.746420
435     0.407155    -0.407155      0.466773    -0.466773
0.746420
102     0.407155    -0.407155     -2.142369      2.142369

```


0.746420

	basement_yes	airconditioning_yes	airconditioning_no
prefarea_yes \			
46	-0.746420	1.501243	-1.501243
-0.552620			
93	1.339728	1.501243	-1.501243
-0.552620			
335	1.339728	1.501243	-1.501243
-0.552620			
412	1.339728	-0.666115	0.666115
1.809561			
471	-0.746420	-0.666115	0.666115
-0.552620			
..
...			
71	-0.746420	1.501243	-1.501243
-0.552620			
106	1.339728	1.501243	-1.501243
1.809561			
270	-0.746420	-0.666115	0.666115
-0.552620			
435	-0.746420	-0.666115	0.666115
-0.552620			
102	-0.746420	1.501243	-1.501243
-0.552620			

	prefarea_no	furnishingstatus_furnished	furnishingstatus_semi-
furnished \			
46	0.552620	1.711119	
-0.870669			
93	0.552620	-0.584413	
1.148542			
335	0.552620	1.711119	
-0.870669			
412	-1.809561	-0.584413	
-0.870669			
471	0.552620	-0.584413	
-0.870669			
..
...			
71	0.552620	-0.584413	
-0.870669			
106	-1.809561	-0.584413	
1.148542			
270	0.552620	1.711119	
-0.870669			
435	0.552620	-0.584413	

-0.870669
102 0.552620 -0.584413
1.148542

	furnishingstatus_unfurnished	area	bedrooms	bathrooms
stories \				
46	-0.676900	0.384168	0.055271	1.539173
2.587644				
93	-0.676900	0.929181	0.055271	1.539173
-0.912499				
335	-0.676900	-0.607755	-1.283514	-0.557950
-0.912499				
412	1.477322	-1.155492	0.055271	-0.557950
0.254215				
471	1.477322	-0.637730	0.055271	-0.557950
0.254215				
..
...				
71	1.477322	0.384168	1.394055	1.539173
2.587644				
106	-0.676900	0.134371	1.394055	1.539173
-0.912499				
270	-0.676900	-0.297097	0.055271	1.539173
1.420929				
435	1.477322	-0.506019	-1.283514	-0.557950
-0.912499				
102	-0.676900	0.157080	0.055271	1.539173
2.587644				

	parking
46	0.367957
93	2.709987
335	1.538972
412	-0.803059
471	-0.803059
..	...
71	-0.803059
106	-0.803059
270	0.367957
435	-0.803059
102	0.367957

[436 rows x 18 columns]

X_test_scaled:

	mainroad_yes	mainroad_no	guestroom_no	guestroom_yes
basement_no \				
316	-2.456066	2.456066	0.466773	-0.466773
-1.339728				

77	0.407155	-0.407155	0.466773	-0.466773
0.746420				
360	0.407155	-0.407155	0.466773	-0.466773
0.746420				
90	0.407155	-0.407155	0.466773	-0.466773
0.746420				
493	0.407155	-0.407155	0.466773	-0.466773
0.746420				
..
...				
15	0.407155	-0.407155	0.466773	-0.466773
-1.339728				
357	-2.456066	2.456066	0.466773	-0.466773
0.746420				
39	0.407155	-0.407155	0.466773	-0.466773
0.746420				
54	0.407155	-0.407155	-2.142369	2.142369
0.746420				
155	0.407155	-0.407155	0.466773	-0.466773
-1.339728				

	basement_yes	airconditioning_yes	airconditioning_no
prefarea_yes \			
316	1.339728	-0.666115	0.666115
-0.552620			
77	-0.746420	1.501243	-1.501243
1.809561			
360	-0.746420	-0.666115	0.666115
-0.552620			
90	-0.746420	1.501243	-1.501243
-0.552620			
493	-0.746420	-0.666115	0.666115
-0.552620			
..
...			
15	1.339728	-0.666115	0.666115
-0.552620			
357	-0.746420	-0.666115	0.666115
-0.552620			
39	-0.746420	1.501243	-1.501243
-0.552620			
54	-0.746420	1.501243	-1.501243
-0.552620			
155	1.339728	-0.666115	0.666115
1.809561			

	prefarea_no	furnishingstatus_furnished	furnishingstatus_semi-furnished \
--	-------------	----------------------------	-----------------------------------

316	0.552620	-0.584413
-0.870669		
77	-1.809561	1.711119
-0.870669		
360	0.552620	-0.584413
1.148542		
90	0.552620	-0.584413
1.148542		
493	0.552620	1.711119
-0.870669		
..
15	0.552620	-0.584413
1.148542		
357	0.552620	1.711119
-0.870669		
39	0.552620	-0.584413
1.148542		
54	0.552620	-0.584413
1.148542		
155	-1.809561	1.711119
-0.870669		

	furnishingstatus_unfurnished	area	bedrooms	bathrooms
stories \				
316	1.477322	0.338750	1.394055	1.539173
0.254215				
77	-0.676900	0.611257	0.055271	1.539173
1.420929				
360	-0.676900	-0.506019	-1.283514	-0.557950
-0.912499				
90	-0.676900	-0.070009	0.055271	-0.557950
0.254215				
493	-0.676900	-0.542353	0.055271	-0.557950
-0.912499				
..
15	-0.676900	0.384168	1.394055	-0.557950
0.254215				
357	-0.676900	0.806553	1.394055	-0.557950
0.254215				
39	-0.676900	0.384168	1.394055	1.539173
2.587644				
54	-0.676900	0.384168	0.055271	1.539173
0.254215				
155	-0.676900	0.429586	0.055271	1.539173
-0.912499				

```
parking
316  0.367957
77   -0.803059
360  -0.803059
90   -0.803059
493  -0.803059

..      ...
15    1.538972
357   0.367957
39    0.367957
54    0.367957
155   1.538972
```

```
[109 rows x 18 columns]
```

```
y_train_scaled:
```

```
[[ 1.60509012]
 [ 0.90746568]
 [-0.44791897]
 [-0.72696874]
 [-0.96615427]
 [-0.04927642]
 [-1.68371084]
 [-0.96615427]
 [-1.24520405]
 [-0.56751173]
 [-0.57149815]
 [-1.2850683 ]
 [-0.28846195]
 [-0.60737598]
 [-0.60737598]
 [ 0.38524395]
 [-1.6739441 ]
 [ 0.30950186]
 [-0.68710449]
 [ 1.18651545]
 [-0.44791897]
 [-0.44791897]
 [ 0.6284159 ]
 [ 0.30950186]
 [ 0.7081444 ]
 [-0.68710449]
 [-1.30898685]
 [-1.16547554]
 [ 0.32943399]
 [-0.58744385]
 [-0.28846195]
 [-1.38472894]
 [-0.64724024]
```

[-0.04927642]
[-0.28846195]
[-0.92230359]
[-0.82662938]
[2.36251095]
[-0.60737598]
[1.50144306]
[-0.00941217]
[-0.22866557]
[-0.60737598]
[-0.17285561]
[-0.24859769]
[-1.00601852]
[0.05038421]
[2.18312181]
[-0.26852982]
[1.70475076]
[-0.68710449]
[-0.766833]
[0.15004485]
[-0.88642576]
[0.46895888]
[0.66828015]
[-0.56751173]
[-0.60737598]
[2.1033933]
[0.78388649]
[0.03045208]
[-0.92629001]
[0.30551544]
[1.44563311]
[-0.08914068]
[2.20305393]
[2.42230733]
[-0.68710449]
[1.26624396]
[-0.05326285]
[-0.20873344]
[-0.13299136]
[-0.52764747]
[0.90746568]
[0.28956974]
[0.1899091]
[-0.10907281]
[2.24291819]
[-1.16547554]
[-0.90416535]
[-1.2850683]

[3.49864219]
[0.74800866]
[1.62502225]
[-0.33231263]
[-0.34626512]
[2.50203584]
[1.56522587]
[0.11018059]
[-0.73095517]
[4.27599515]
[-0.18481489]
[-0.48778322]
[1.38583672]
[0.15004485]
[-1.32493256]
[0.78787291]
[-0.56751173]
[-0.28846195]
[2.1033933]
[-0.04927642]
[0.11018059]
[2.13927113]
[3.09999965]
[0.28956974]
[1.54529374]
[0.79783898]
[1.86420777]
[0.44902675]
[-1.2850683]
[0.7081444]
[-0.70703662]
[0.74800866]
[-1.2850683]
[0.11018059]
[0.30950186]
[-0.62730811]
[-0.00941217]
[0.50882313]
[0.16997697]
[-0.766833]
[-0.80669725]
[-0.0931271]
[-0.28846195]
[-0.74690087]
[-1.16946197]
[0.53872132]
[-0.74690087]
[1.24631184]

[-0.24859769]
[-0.00941217]
[0.30950186]
[0.20984123]
[-0.40805471]
[-0.78676513]
[-1.14554341]
[0.18592268]
[1.08685482]
[0.78787291]
[-0.28846195]
[0.50882313]
[0.96726206]
[0.48889101]
[-0.92629001]
[-0.88642576]
[-0.12900493]
[-0.24461127]
[-0.70703662]
[-0.22069272]
[0.84766929]
[-0.38812258]
[1.10678695]
[-0.52764747]
[-1.32493256]
[0.7081444]
[-0.36819046]
[-0.46785109]
[-0.68710449]
[-1.05584884]
[-0.28846195]
[-1.00601852]
[-0.44791897]
[-0.80669725]
[-0.3283262]
[1.9000856]
[0.58855164]
[-0.10907281]
[-0.38812258]
[1.42570098]
[0.78787291]
[-0.28846195]
[-0.28846195]
[-0.14893706]
[-0.72696874]
[0.50882313]
[-0.17285561]
[-0.80669725]

[1.85424171]
[-0.5316339]
[0.13011272]
[-0.72696874]
[0.50882313]
[-0.88642576]
[0.50882313]
[-0.5316339]
[-0.38812258]
[0.7081444]
[-1.00601852]
[0.90746568]
[0.50882313]
[-0.06920855]
[-0.96615427]
[0.90746568]
[-0.88642576]
[-0.28846195]
[-0.68710449]
[0.7081444]
[0.24970548]
[0.11018059]
[-0.08914068]
[-0.48778322]
[-0.44791897]
[-0.68710449]
[-0.72696874]
[-0.88642576]
[-0.36819046]
[0.70415798]
[-0.48778322]
[0.03045208]
[-1.26513618]
[0.05038421]
[0.66429373]
[-1.00601852]
[-0.39808865]
[-1.46445745]
[-1.04588278]
[-1.04588278]
[-0.69109092]
[-0.44791897]
[-0.28846195]
[-0.74690087]
[-0.64724024]
[-0.82662938]
[0.78388649]
[0.28956974]

[-0.50771535]
[-0.64724024]
[-0.60737598]
[-0.34825833]
[0.48889101]
[-0.88642576]
[-0.04927642]
[-1.20533979]
[-0.52764747]
[0.38923037]
[-0.36819046]
[0.58855164]
[0.142072]
[-0.12900493]
[-0.0293443]
[0.74800866]
[0.7081444]
[-0.68710449]
[0.03045208]
[-0.90635789]
[-0.62730811]
[-0.49575607]
[0.66828015]
[-1.48438957]
[-0.26852982]
[-1.08574703]
[-0.96615427]
[0.80780504]
[-0.766833]
[-0.58744385]
[0.6284159]
[-1.36479681]
[1.62502225]
[0.30950186]
[1.26624396]
[-0.24859769]
[1.02705844]
[-0.08914068]
[0.22977335]
[-0.80669725]
[0.42909463]
[2.6216286]
[1.92169203]
[2.1033933]
[0.30950186]
[0.50882313]
[-0.88642576]
[-0.66717236]

[0.03443851]
[0.90746568]
[1.54529374]
[-0.06920855]
[-0.766833]
[-0.52764747]
[-0.80669725]
[0.29754259]
[-0.84656151]
[1.78447927]
[-1.12561129]
[-0.84656151]
[0.11018059]
[-0.68710449]
[-0.24859769]
[-0.5475796]
[-0.68710449]
[-0.12900493]
[-0.96615427]
[3.81755623]
[-0.12900493]
[-1.20533979]
[0.86760142]
[-0.00941217]
[-1.20533979]
[0.10619417]
[-1.00601852]
[0.30950186]
[0.90347925]
[-0.28846195]
[0.07031634]
[1.34597247]
[-0.70703662]
[1.585158]
[2.04359692]
[-0.17285561]
[-0.40805471]
[0.11018059]
[-1.19736694]
[2.1033933]
[-1.0658149]
[-0.00941217]
[-0.13299136]
[-0.60737598]
[-0.28846195]
[-0.766833]
[0.42909463]
[-0.68710449]

[-1.00601852]
[0.58855164]
[1.34597247]
[0.60848377]
[0.66429373]
[-1.38472894]
[2.94054264]
[-0.40805471]
[-0.48778322]
[1.30610822]
[-0.28846195]
[2.58176435]
[0.46497245]
[1.80441139]
[-0.61136241]
[0.22977335]
[-1.48438957]
[0.07031634]
[-0.72696874]
[-0.24859769]
[-1.12561129]
[-1.0658149]
[-0.44791897]
[-0.04927642]
[0.11018059]
[0.10619417]
[-0.28846195]
[-1.26513618]
[-0.10907281]
[-1.56411808]
[-0.89838504]
[-1.68371084]
[-0.28846195]
[-0.52764747]
[-0.18880131]
[-0.88642576]
[0.11018059]
[0.42909463]
[1.70475076]
[0.07031634]
[-0.68710449]
[0.1899091]
[-0.3283262]
[-0.13299136]
[0.82773717]
[1.50542949]
[-0.68710449]
[4.29592728]

[-1.16547554]
[-0.56751173]
[1.54529374]
[-0.766833]
[1.09482767]
[0.11018059]
[1.94393628]
[-0.80669725]
[-0.16886919]
[-0.7827787]
[-0.89041219]
[-0.84656151]
[-0.58744385]
[0.97523491]
[-0.46785109]
[-1.34486468]
[1.50542949]
[-0.48778322]
[-0.16886919]
[-0.08914068]
[2.58176435]
[-0.34825833]
[1.00712631]
[-0.68710449]
[-0.20873344]
[0.34936612]
[-0.40805471]
[-1.62391446]
[0.62442947]
[-0.30839408]
[-1.04588278]
[-1.40466106]
[-1.16946197]
[-0.36819046]
[-0.08914068]
[0.1899091]
[-0.99405925]
[-1.24520405]
[1.42570098]
[-0.9860864]
[-0.10907281]
[2.26285032]
[-0.34825833]
[-0.90635789]
[0.42909463]
[-0.22866557]
[0.1899091]
[-0.64724024]

```
[-0.72696874]
[-1.0658149 ]
[-0.48778322]
[-0.30839408]
[ 0.6603073 ]
[ 0.88753355]
[-0.60737598]
[ 1.00712631]
[-0.90635789]
[-0.40805471]
[ 0.09024847]
[-0.96615427]
[ 0.7081444 ]
[ 2.30271457]
[ 1.16658333]
[ 0.82773717]
[-0.20873344]
[-0.80669725]
[ 0.84766929]]
y_test_scaled:
[[-0.36819046]
[ 1.10678695]
[-0.56751173]
[ 0.98719418]
[-1.08574703]
[ 0.11018059]
[ 0.30950186]
[-0.0931271 ]
[-1.2850683 ]
[-0.77081942]
[ 3.09999965]
[-1.16547554]
[-0.766833 ]
[-0.766833 ]
[-1.38472894]
[-1.16547554]
[-1.16547554]
[ 1.50542949]
[-1.00601852]
[-1.04588278]
[ 1.1466512 ]
[ 0.4091625 ]
[-1.60398234]
[ 0.30950186]
[-0.29244837]
[ 4.29592728]
[-0.92629001]
[ 0.22977335]
```

[2.90067838]
[-1.24520405]
[1.18651545]
[-0.68710449]
[1.10678695]
[-1.00601852]
[-0.78676513]
[-0.28846195]
[0.11018059]
[-0.80669725]
[-0.68710449]
[-1.32493256]
[0.44902675]
[-0.58744385]
[1.10678695]
[0.11416702]
[-0.88642576]
[-0.12900493]
[1.06692269]
[0.58855164]
[-0.97014069]
[-1.64384659]
[2.1033933]
[-1.2850683]
[-0.24859769]
[-0.39808865]
[-0.83858866]
[-1.68371084]
[2.90067838]
[-1.48438957]
[-0.20873344]
[-0.94622214]
[-0.48778322]
[-0.68710449]
[0.02646566]
[-0.92629001]
[-0.08914068]
[1.02705844]
[1.12671907]
[0.22977335]
[-0.08914068]
[1.10678695]
[-0.60737598]
[0.50882313]
[-0.52764747]
[1.08685482]
[-0.84656151]
[1.10678695]

```
[-0.00941217]
[ 0.07031634]
[ 1.22637971]
[-1.40864749]
[ 1.32604034]
[-0.30839408]
[ 0.84766929]
[ 1.02705844]
[-1.60398234]
[ 1.98380054]
[-1.16547554]
[-0.29244837]
[ 4.89389109]
[ 2.83290915]
[-0.20873344]
[ 2.24291819]
[-0.57149815]
[ 0.24970548]
[ 0.98719418]
[ 0.7081444 ]
[ 0.6284159 ]
[ 0.58855164]
[ 0.86760142]
[ 0.50882313]
[-0.96615427]
[ 2.38244308]
[ 0.11018059]
[ 0.46895888]
[ 2.50203584]
[-0.5316339 ]
[ 1.82434352]
[ 1.50542949]
[ 0.46895888]]
```

```
class SimpleLinearRegression:
    def __init__(self, epochs, learning_rate):
        self.epochs = epochs
        self.learning_rate = learning_rate
        self.weights = None
        self.bias = 0

    def fit(self, X, y):
        X = np.array(X)
        y = np.array(y).flatten()
        m, n_features = X.shape
        self.weights = np.zeros(n_features)
        for epoch in range(self.epochs):
            for i in range(m):
```



```

        xi = X[i, :]
        yi = y[i]
        y_pred = np.dot(xi, self.weights) + self.bias
        error = y_pred - yi
        gradient_w = xi * error
        gradient_b = error
        self.weights -= self.learning_rate * gradient_w
        self.bias -= self.learning_rate * gradient_b
    def predict(self, X):

        return np.dot(X, self.weights) + self.bias
sgd_model = SimpleLinearRegression(learning_rate=0.0001, epochs=1000)
sgd_model.fit(X_train_scaled, y_train_scaled)
predictions = sgd_model.predict(X_test_scaled)

mse_error=mean_squared_error(y_test_scaled,predictions)
mae_error=mean_absolute_error(y_test_scaled,predictions)
r2_error=r2_score(y_test_scaled,predictions)
print(f"Mean Squared Error: {mse_error}")
print(f"Mean Absolute Error: {mae_error}")
print(f"r2-score: {r2_error}")

```

Mean Squared Error: 0.5891193268751346
Mean Absolute Error: 0.5666963223683446
r2-score: 0.6406250277445966

```

class BatchLinearRegression:
    def __init__(self, epochs=1000, learning_rate=0.0001):
        self.epochs = epochs
        self.learning_rate = learning_rate
        self.weights = None
        self.bias = 0

    def fit(self, X, y):
        X = np.array(X)
        y = np.array(y).flatten()
        m, n_features = X.shape
        self.weights = np.zeros(n_features)
        self.bias = 0

        for epoch in range(self.epochs):
            y_pred = np.dot(X, self.weights) + self.bias
            error = y_pred - y
            gradient_w = (1/m) * np.dot(X.T, error)
            gradient_b = (1/m) * np.sum(error)
            self.weights -= self.learning_rate * gradient_w

```

```

        self.bias -= self.learning_rate * gradient_b

def predict(self, X):
    X = np.array(X)
    return np.dot(X, self.weights) + self.bias

```

```

model = BatchLinearRegression(learning_rate=0.0001, epochs=1000)
model.fit(X_train_scaled, y_train_scaled)
predictions=model.predict(X_test_scaled)
mse_error=mean_squared_error(y_test_scaled,predictions)
mae_error=mean_absolute_error(y_test_scaled,predictions)
r2_error=r2_score(y_test_scaled,predictions)
print(f"Mean Squared Error: {mse_error}")
print(f"Mean Absolute Error: {mae_error}")
print(f"r2-score: {r2_error}")

```

Mean Squared Error: 1.2345718705658937
 Mean Absolute Error: 0.8364027589907924
 r2-score: 0.24688562827279736


advertising_df

	TV	Radio	Newspaper	Sales
0	230.1	37.8	69.2	22.1
1	44.5	39.3	45.1	10.4
2	17.2	45.9	69.3	12.0
3	151.5	41.3	58.5	16.5
4	180.8	10.8	58.4	17.9
...
195	38.2	3.7	13.8	7.6
196	94.2	4.9	8.1	14.0
197	177.0	9.3	6.4	14.8
198	283.6	42.0	66.2	25.5
199	232.1	8.6	8.7	18.4

200 rows × 4 columns

<div class="colab-df-buttons">

<button class="colab-df-convert"
 onclick="convertToInteractive('df-
 d30e005d-6e8c-4948-8a47-7f1069d26c90')"
 title="Convert this dataframe
 to an interactive table."
 style="display:none;">

 <script>
 const buttonEl =
 document.querySelector('#df-
 d30e005d-6e8c-4948-8a47-7f1069d26c90
 convert');
 buttonEl.style.display =

```

        google.colab.kernel.accessAllowed

        async function convertToInteractiveTable(element) {
            const element = document.querySelector('#' + key + ' table');
            const dataTable = await google.colab.kernel.invokeFunction('convertToInteractiveTable', [element], {});
            if (!dataTable) return;

            const docLinkHtml = 'Like what you saw?<br>';
            const docLink = document.createElement('a');
            docLink.innerHTML = docLinkHtml;
            element.appendChild(docLink);
        }
    </script>

```

```

<div id="df-7c049a99-7fe3-4911-8279-493e88cb7cff">
  <button class="colab-df-quickchart"
    onclick="quickchart('df-7c049a99-7fe3-4911-8279-493e88cb7cff')"
    title="Suggest charts"
    style="display:none;">

```



```

</button>

```

```

<script>
  async function quickchart(key) {
    const quickchartButtonEl =
      document.querySelector('#' + key + ' button');
    quickchartButtonEl.disabled = true; // To prevent multiple
    clicks.
    quickchartButtonEl.classList.add('colab-df-spinner');
    try {
      const charts = await google.colab.kernel.invokeFunction('convertToInteractiveTable', [quickchartButtonEl], {});
    } catch {
      // Ignore error
    }
  }

```

```

        'suggestCharts', [key], {}));
    } catch (error) {
        console.error('Error during call to suggestCharts:', error);
    }
    quickchartButtonEl.classList.remove('colab-df-spinner');
    quickchartButtonEl.classList.add('colab-df-quickchart-
complete');
}
(() => {
    let quickchartButtonEl =

```

```

document.querySelector('#df-7c049a99-7fe3-4911-8279-493e88cb7cff
button');
    quickchartButtonEl.style.display =
        google.colab.kernel.accessAllowed ? 'block' : 'none';
    })();
</script>
</div>

```

```

<style>
.colab-df-generate {
    background-color: #E8F0FE;
    border: none;
    border-radius: 50%;
    cursor: pointer;
    display: none;
    fill: #1967D2;
    height: 32px;
    padding: 0 0 0 0;
    width: 32px;
}

.colab-df-generate:hover {
    background-color: #E2EBFA;
    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
rgba(60, 64, 67, 0.15);
    fill: #174EA6;
}

[theme=dark] .colab-df-generate {
    background-color: #3B4455;
    fill: #D2E3FC;
}

[theme=dark] .colab-df-generate:hover {
    background-color: #434B5C;
    box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);
    filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));
    fill: #FFFFFF;
}

```

```

    }
</style>
<button class="colab-df-generate"
onclick="generateWithVariable('advertising_df')"
        title="Generate code using this dataframe."
        style="display:none;">

```



```
</div>
```

```

numerical_data = advertising_df.select_dtypes(include='number')
categorical_data = advertising_df.select_dtypes(exclude='number')

```

```

encoded_df = pd.DataFrame(index=advertising_df.index)
for col in categorical_data.columns:
    for val in categorical_data[col].unique():
        encoded_df[f"{col}_{val}"] = (categorical_data[col] ==
        val).astype(int)

```

```

encoded_df = pd.concat([encoded_df, numerical_data], axis=1)
X = encoded_df.drop("Sales", axis=1)
y = encoded_df["Sales"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
        test_size=0.2, random_state=42)
num_cols = [col for col in numerical_data.columns if col != 'Sales']
X_train_scaled = X_train.copy()
X_test_scaled = X_test.copy()
X_mean = X_train[num_cols].mean()
X_std = X_train[num_cols].std(ddof=0)
X_train_scaled[num_cols] = (X_train[num_cols] - X_mean) / X_std
X_test_scaled[num_cols] = (X_test[num_cols] - X_mean) / X_std
X_train_scaled = X_train_scaled.values
X_test_scaled = X_test_scaled.values
y_mean = y_train.mean()
y_std = y_train.std(ddof=0)
y_train_scaled = ((y_train - y_mean) / y_std).values.flatten()
y_test_scaled = ((y_test - y_mean) / y_std).values.flatten()
sgd_model = SimpleLinearRegression(learning_rate=0.0001, epochs=1000)
sgd_model.fit(X_train_scaled, y_train_scaled)
predictions = sgd_model.predict(X_test_scaled)

```

```

mse_error = mean_squared_error(y_test_scaled, predictions)
mae_error = mean_absolute_error(y_test_scaled, predictions)
r2_error = r2_score(y_test_scaled, predictions)

```

```
print(f"Mean Squared Error: {mse_error}")
print(f"Mean Absolute Error: {mae_error}")
print(f"R2-score: {r2_error}")
```

Mean Squared Error: 0.10851288190773564
Mean Absolute Error: 0.24629249736207912
R2-score: 0.9058908138484301

```
model = BatchLinearRegression(learning_rate=0.0001, epochs=1000)
model.fit(X_train_scaled, y_train_scaled)
predictions=model.predict(X_test_scaled)
mse_error=mean_squared_error(y_test_scaled,predictions)
mae_error=mean_absolute_error(y_test_scaled,predictions)
r2_error=r2_score(y_test_scaled,predictions)
print(f"Mean Squared Error: {mse_error}")
print(f"Mean Absolute Error: {mae_error}")
print(f"r2-score: {r2_error}")
```

Mean Squared Error: 0.9848789374046113
Mean Absolute Error: 0.8587336612361215
r2-score: 0.14585113188885312