

Start coding or [generate](#) with AI.

Upload the Dataset

```
from google.colab import files
uploaded = files.upload()
```

 Choose Files Data set

- **Data set**(n/a) - 71251 bytes, last modified: 5/8/2025 - 100% done


Saving Data set to Data set

Load the Dataset

```
import pandas as pd
import io

# Display uploaded file names
print(uploaded.keys()) # This will help you find the actual filename

# Then load the dataset using the correct name
df = pd.read_csv(io.BytesIO(uploaded['Data set'])) # No .csv if the filename is "Data set"
df.head()
```

 dict\_keys(['Data set'])

	Company Name	Symbol	Industry	Series	Open	High	Low	Previous Close	Last Traded Price	Change	Percentage Change	Share Volume	Value (Indian Rupee)
0	3M India Ltd.	3MINDIA	Diversified	EQ	21950.00	21999.00	21126.05	21854.05	21575.00	-279.05	-1.28	4159	8.965357e+07
1	Aarti Drugs Ltd.	AARTIDRUGS	Healthcare	EQ	400.50	401.80	394.10	403.85	400.00	-3.85	-0.95	31782	1.262731e+07
2	Aavas Financiers Ltd.	AAVAS	Financial Services	EQ	1997.10	2004.05	1894.50	2015.45	1943.15	-72.30	-3.59	150704	2.907728e+08

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

Data Exploration

```
df.info()
df.describe(include='all')
```

```
>>> df
Out[1]: <class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 501 entries, 0 to 500

Data columns (total 17 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

	Column	Non-null Count	Type
0	Company Name	501 non-null	object
1	Symbol	501 non-null	object
2	Industry	501 non-null	object
3	Series	501 non-null	object
4	Open	501 non-null	float64
5	High	501 non-null	float64
6	Low	501 non-null	float64
7	Previous Close	501 non-null	float64
8	Last Traded Price	501 non-null	float64
9	Change	501 non-null	object
10	Percentage Change	501 non-null	object
11	Share Volume	501 non-null	int64
12	Value (Indian Rupee)	501 non-null	float64
13	52 Week High	501 non-null	float64
14	52 Week Low	501 non-null	float64
15	365 Day Percentage Change	501 non-null	object
16	30 Day Percentage Change	501 non-null	object


```
dtypes: float64(8), int64(1), object(8)
```

```
memory usage: 66.7+ KB
```

	Company Name	Symbol	Industry	Series	Open	High	Low	Previous Close	Last Traded Price	Change	Percentage Change
count	501	501	501	501	501.000000	501.000000	501.000000	501.000000	501.000000	501	501
unique	501	501	21	2	NaN	NaN	NaN	NaN	NaN	372	354
top	Zydu Wellness Ltd.	ZYDUSWELL	Financial Services	EQ	NaN	NaN	NaN	NaN	NaN	0.10	-0.33
freq	1	1	88	500	NaN	NaN	NaN	NaN	NaN	7	5
mean	NaN	NaN	NaN	NaN	1525.904491	1553.804990	1504.042415	1528.061277	1536.925449	NaN	NaN
std	NaN	NaN	NaN	NaN	4466.627117	4576.377692	4435.492332	4477.209376	4532.004734	NaN	NaN
min	NaN	NaN	NaN	NaN	6.750000	6.950000	6.700000	6.850000	6.800000	NaN	NaN
25%	NaN	NaN	NaN	NaN	215.300000	221.550000	210.600000	217.200000	214.650000	NaN	NaN
50%	NaN	NaN	NaN	NaN	551.100000	569.100000	547.000000	554.750000	563.000000	NaN	NaN
75%	NaN	NaN	NaN	NaN	1404.500000	1421.250000	1396.850000	1411.700000	1410.000000	NaN	NaN
max	NaN	NaN	NaN	NaN	70300.000000	72500.000000	70300.000000	70800.900000	71900.000000	NaN	NaN

## Check for Missing Values

```
df.isnull().sum()
```



	0
Company Name	0
Symbol	0
Industry	0
Series	0
Open	0
High	0
Low	0
Previous Close	0
Last Traded Price	0
Change	0
Percentage Change	0
Share Volume	0
Value (Indian Rupee)	0
52 Week High	0
52 Week Low	0
365 Day Percentage Change	0
30 Day Percentage Change	0

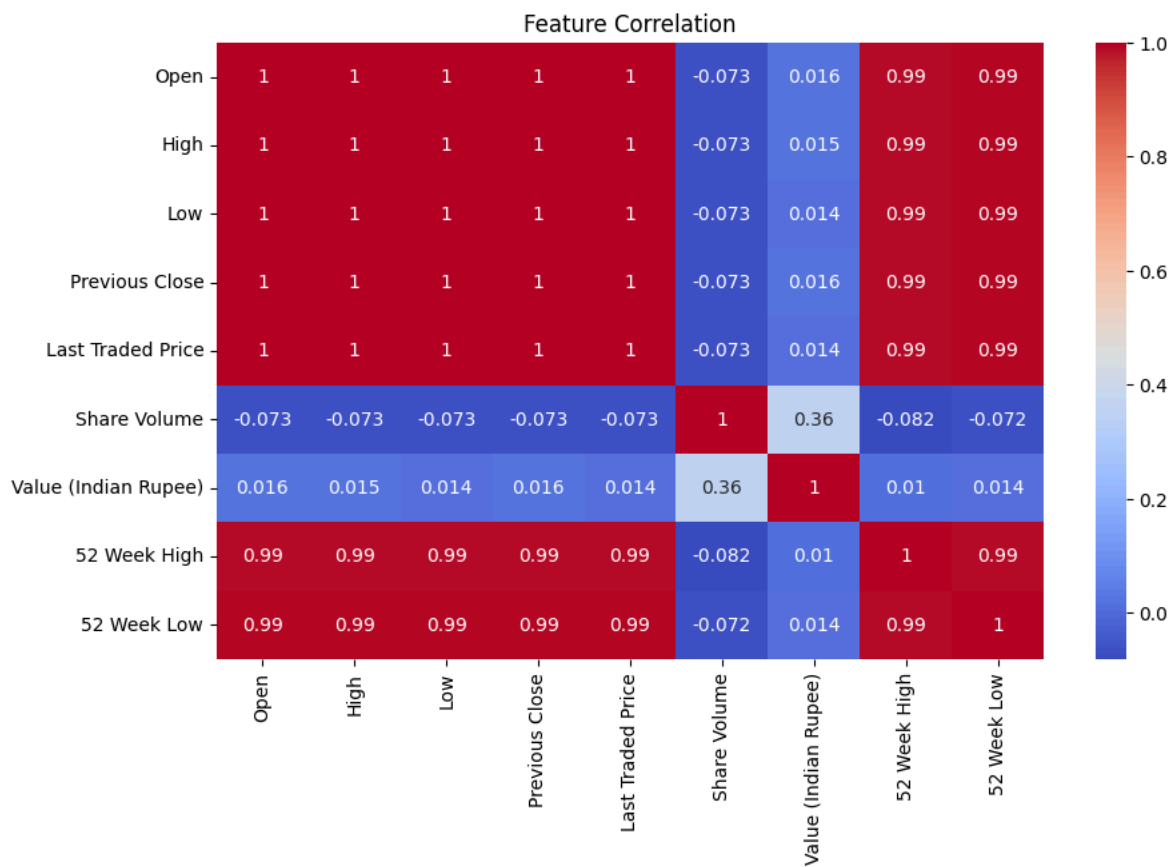
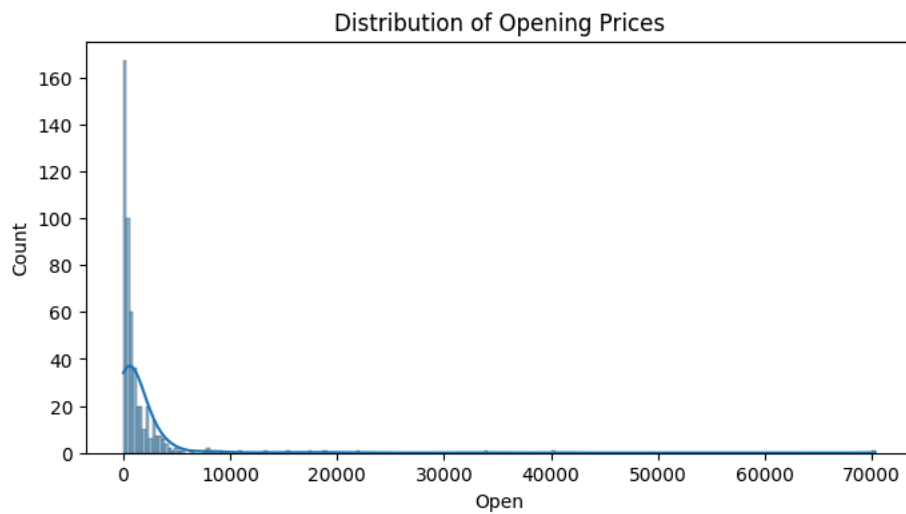
dtype: int64

### Visualize a Few Features

```
import matplotlib.pyplot as plt
import seaborn as sns

# Distribution of opening prices
plt.figure(figsize=(8, 4))
sns.histplot(df['Open'], kde=True)
plt.title("Distribution of Opening Prices")
plt.show()

# Correlation heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.select_dtypes(include='number').corr(), annot=True, cmap="coolwarm")
plt.title("Feature Correlation")
plt.show()
```



### Identify Target and Features

```
# Predicting 'Last Traded Price'
target = 'Last Traded Price'
features = df.select_dtypes(include=['number']).drop(columns=[target]).columns.tolist()
X = df[features]
y = df[target]
```

### Convert Categorical Data

```
categorical = df.select_dtypes(include='object').drop(columns=['Change', 'Percentage Change']).columns
df[categorical].head()
```



	Company Name	Symbol	Industry	Series	365 Day Percentage Change	30 Day Percentage Change
0	3M India Ltd.	3MINDIA	Diversified	EQ	-10.18	8.22
1	Aarti Drugs Ltd.	AARTIDRUGS	Healthcare	EQ	-42.92	-5.10
2	Aavas Financiers Ltd.	AAVAS	Financial Services	EQ	-25.69	-9.39
3	ABB India Ltd.	ABB	Capital Goods	EQ	27.25	-5.21
4	Abbott India Ltd.	ABBOTINDIA	Healthcare	EQ	11.61	5.84




## One Hot Encoding

```
df_encoded = pd.get_dummies(df, columns=categorical, drop_first=True)
```

## Feature Scaling

```
target = 'Last Traded Price'
```

## Train Test Split python Copy Edit

```
y = y.reset_index(drop=True)
```

## Model Building

```
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor()
```

## Evaluation python Copy

```
import numpy as np

# Fix percentage/numeric columns: remove commas, %, and replace '-' with NaN
for col in ['Change', 'Percentage Change', '365 Day Percentage Change', '30 Day Percentage Change', 'Last Traded Price']:
    df[col] = df[col].astype(str).str.replace(',', '', regex=False) # remove commas
    df[col] = df[col].str.replace('%', '', regex=False) # remove %
    df[col] = df[col].replace('-', np.nan) # replace '-' with NaN
    df[col] = df[col].astype(float) # convert to float
```

## Make Prediction from New Input


```
# Example: using a random row
sample_input = 3153.61
```

## Convert to DataFrame and Encode

```
# Already done in Step 8 using pd.get_dummies
# df_encoded is your working DataFrame
```

Predict the Final Grade (in this context, the price)

[3153.61, 440.90, 635.23, 396.65, 596.84]

 [3153.61, 440.9, 635.23, 396.65, 596.84]

Build an Interactive App (Streamlit alternative using Gradio)

!pip install gradio

```

Requirement already satisfied: pillow<12.0,>=8.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (11.2.1)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (2.11.4)
Collecting pydub (from gradio)
  Downloading pydub-0.25.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting python-multipart>=0.0.18 (from gradio)
  Downloading python_multipart-0.0.20-py3-none-any.whl.metadata (1.8 kB)
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (6.0.2)
Collecting ruff>=0.9.3 (from gradio)
  Downloading ruff-0.11.8-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (25 kB)
Collecting safehttpx<0.2.0,>=0.1.6 (from gradio)
  Downloading safehttpx-0.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting semantic-version~2.0 (from gradio)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
Collecting starlette<1.0,>=0.40.0 (from gradio)
  Downloading starlette-0.46.2-py3-none-any.whl.metadata (6.2 kB)
Collecting tomlkit<0.14.0,>=0.12.0 (from gradio)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-packages (from gradio) (0.15.3)
Requirement already satisfied: typing-extensions~4.0 in /usr/local/lib/python3.11/dist-packages (from gradio) (4.13.2)
Collecting uvicorn>=0.14.0 (from gradio)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.11/dist-packages (from gradio-client==1.10.0->gradio)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (2025.4.26)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx>=0.24.1->gradio) (0.16)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (3.18.0)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.28.1->gradio) (4.67.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio) (2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<2.12,>=2.0->gradio)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in /usr/local/lib/python3.11/dist-packages (from typer<1.0,>=0.12->gradio) (13.9.4)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas<3.0,>=1.0->gradio)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich>=10.11.0->typer<1.0,>=0.12->gradio)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.28.1->gradio)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub>=0.28.1->gradio)
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->rich>=10.11.0->typer<1.0,>=0.12->gradio)
Downloading gradio-5.29.0-py3-none-any.whl (54.1 MB)
54.1/54.1 MB 21.7 MB/s eta 0:00:00
Downloading gradio_client-1.10.0-py3-none-any.whl (322 kB)
322.9/322.9 kB 23.9 MB/s eta 0:00:00
Downloading aiofiles-24.1.0-py3-none-any.whl (15 kB)
Downloading fastapi-0.115.12-py3-none-any.whl (95 kB)
95.2/95.2 kB 7.8 MB/s eta 0:00:00
Downloading groovy-0.1.2-py3-none-any.whl (14 kB)
Downloading python_multipart-0.0.20-py3-none-any.whl (24 kB)
Downloading ruff-0.11.8-py3-none-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
11.5/11.5 MB 108.8 MB/s eta 0:00:00
Downloading safehttpx-0.1.6-py3-none-any.whl (8.7 kB)

```

Create a Prediction Function

```

def predict_price(Open, High, Low, Previous_Close, Volume):
    input_data = pd.DataFrame([[Open, High, Low, Previous_Close, Volume]],
                               columns=['Open', 'High', 'Low', 'Previous Close', 'Share Volume'])
    input_scaled = scaler.transform(input_data)

```

```
prediction = model.predict(input_scaled)
return f"Predicted Last Traded Price: ₹{prediction[0]:.2f}"
```