```
# Stock Data Science Project Pipeline in Google Colab Format

# 1. Upload the Dataset
from google.colab import files
uploaded = files.upload()

# 2. Load the Dataset
import pandas as pd

df = pd.read_csv('Dataset.csv')
df.head()

# 3. Data Exploration
df.info()
df.describe()
df.columns

# 4. Check for Missing Values and Duplicates
print("Missing Values:\n", df.isnull().sum())
print("\nDuplicates:", df.duplicated().sum())
df.drop_duplicates(inplace=True)

# 5. Visualize a Few Features
import matplotlib.pyplot as plt
import seaborn as sns

sns.histplot(df['Open'], kde=True)
plt.title('Distribution of Opening Prices')
plt.show()

# Correlation heatmap
plt.figure(figsize=(12, 6))
sns.heatmap(df.select_dtypes(include='number').corr(), annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

# 6. Identify Target and Features
# Let's predict 'Last Traded Price' as target
target = 'Last Traded Price'

# Drop unnecessary columns and target from features
features = df.drop(columns=[target, 'Company Name', 'Symbol'])
y = df[target]

# 7. Convert Categorical Columns to Numerical
cat_cols = features.select_dtypes(include='object').columns
features[cat_cols] = features[cat_cols].replace({'-': '0'}, regex=True)
# Define columns that contain percentage or currency strings
string_to_float_cols = [
    'Change', 'Percentage Change',
    '365 Day Percentage Change', '30 Day Percentage Change',
    'Value (Indian Rupee)', 'Share Volume',
    '52 Week High', '52 Week Low'
]

# Clean each column: remove %, commas, replace '-' with 0, and convert to float
for col in string_to_float_cols:
    features[col] = (
        features[col]
        .astype(str)
        .str.replace('%', '', regex=False)
        .str.replace(',', '', regex=False)
        .str.replace('-', '0', regex=False)
        .astype(float)
    )



# 8. One-Hot Encoding
features = pd.get_dummies(features, drop_first=True)

# 9. Feature Scaling
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(features)

# 10. Train-Test Split
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

```
# 11. Model Building
from sklearn.ensemble import RandomForestRegressor

model = RandomForestRegressor()
model.fit(X_train, y_train)

# 12. Evaluation
from sklearn.metrics import mean_squared_error, r2_score

preds = model.predict(X_test)
print("R^2 Score:", r2_score(y_test, preds))
import numpy as np

mse = mean_squared_error(y_test, preds)
rmse = np.sqrt(mse)
print("RMSE:", rmse)


# 13. Make Predictions from New Input
# Example input (manually define a row with the same features)
example = pd.DataFrame([features.iloc[0]])  # Using first row for demonstration

# 14. Convert to DataFrame and Encode
example_scaled = scaler.transform(example)

# 15. Predict the Final Grade
prediction = model.predict(example_scaled)
print("Predicted Last Traded Price:", prediction[0])

# 16. Deployment - Building an Interactive App
# 17. Create a Prediction Function
def predict_price(Open, High, Low, Previous_Close, Change, Percentage_Change,
                  Share_Volume, Value, Week_High, Week_Low, Year_Change,
                  Month_Change, Industry, Series):
    df_input = pd.DataFrame.from_dict({
        'Open': [float(Open)],
        'High': [float(High)],
        'Low': [float(Low)],
        'Previous Close': [float(Previous_Close)],
        'Change': [float(Change)],
        'Percentage Change': [float(Percentage_Change)],
        'Share Volume': [int(Share_Volume)],
        'Value (Indian Rupee)': [float(Value)],
        '52 Week High': [float(Week_High)],
        '52 Week Low': [float(Week_Low)],
        '365 Day Percentage Change': [float(Year_Change)],
        '30 Day Percentage Change': [float(Month_Change)],
        'Industry': [Industry],
        'Series': [Series]
    })
    df_input = pd.get_dummies(df_input, drop_first=True)
    df_input = df_input.reindex(columns=features.columns, fill_value=0)
    df_scaled = scaler.transform(df_input)
    pred = model.predict(df_scaled)
    return round(pred[0], 2)

# 18. Create the Gradio Interface
!pip install gradio --quiet
import gradio as gr

interface = gr.Interface(

    fn=predict_price,
    inputs=[
        gr.Number(label='Open'), gr.Number(label='High'), gr.Number(label='Low'),
        gr.Number(label='Previous Close'), gr.Number(label='Change'),
        gr.Number(label='Percentage Change'), gr.Number(label='Share Volume'),
        gr.Number(label='Value (Indian Rupee)'), gr.Number(label='52 Week High'),
        gr.Number(label='52 Week Low'), gr.Number(label='365 Day Percentage Change'),
        gr.Number(label='30 Day Percentage Change'), gr.Textbox(label='Industry'),
        gr.Textbox(label='Series')
    ],
    outputs='text',
    title="Stock Predictor "
)

interface.launch()
```

Choose files No file chosen

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 501 entries, 0 to 500
Data columns (total 17 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 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
Missing Values:
 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

Duplicates: 0
```