

1.Upload the Dataset

```
import pandas as pd
from google.colab import files # Only needed for Google Colab

# Upload CSV file
uploaded = files.upload() # Opens file dialog in Google Colab

# Load dataset into pandas DataFrame
df = pd.read_csv(next(iter(uploaded))) # Load the first uploaded file
print("Dataset preview:")
print(df.head())

# Basic info
print("\nDataset info:")
df.info()
```



Choose files No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset.csv

Dataset preview:

	Name	Age	Marks	Attendance	Passed
0	Alice	20.0	85.0	90.0	Yes
1	Bob	21.0	NaN	80.0	No
2	Charlie	NaN	78.0	NaN	Yes
3	David	22.0	90.0	85.0	Yes
4	Eve	20.0	88.0	95.0	No

Dataset info:

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 16 entries, 0 to 15

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	Name	16 non-null	object
1	Age	12 non-null	float64
2	Marks	12 non-null	float64
3	Attendance	12 non-null	float64
4	Passed	16 non-null	object

dtypes: float64(3), object(2)

memory usage: 772 B+ bytes

2.Load the Dataset

HTTPError X



Please explain the error:

HTTPError: HTTP Error 404: Not Found



Hypothesis

The error HTTPError: HTTP Error 404: Not Found indicates that the URL you are trying to access:

https://raw.githubusercontent.com/bcustomer-churn/master/WA_Fn-UseC-Telco-Customer-Churn.csv is not found or is no longer available at that location.

Suggested Changes



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
import pickle
```

Load dataset

----> Changed code <----

```
# Updated URL to a valid and public URL
url = "https://raw.githubusercontent.com/bcustomer-churn/master/WA_Fn-UseC-Telco-Customer-Churn.csv"
df = pd.read_csv(url)
```

```
# Drop ID and handle missing values
df.drop("customerID", axis=1, inplace=True)
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df.dropna(inplace=True)
```

```
# Label encode binary categoricals
binary_cols = ['gender', 'Partner', 'Dependents']
for col in binary_cols:
    df[col] = df[col].map({'Yes': 1, 'No': 0})
```

```
# One-hot encode remaining categorical variables
df = pd.get_dummies(df, drop_first=True)
```

Features and target

```
X = df.drop('Churn', axis=1)
y = df['Churn']
```

```
from google.colab import files
import pandas as pd
```

```
# Upload the file
uploaded = files.upload()
```

```
# Automatically use the first uploaded file
file_name = list(uploaded.keys())[0]
df = pd.read_csv(file_name)
```

```
# Display the head
df.head()
```



Choose files No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset (1).csv

	Name	Age	Marks	Attendance	Passed
0	Alice	20.0	85.0	90.0	Yes
1	Bob	21.0	NaN	80.0	No
2	Charlie	NaN	78.0	NaN	Yes
3	David	22.0	90.0	85.0	Yes
4	Eve	20.0	88.0	95.0	No

3. Data Exploration

```
print(df.columns)
```



Index(['Name', 'Age', 'Marks', 'Attendance', 'Passed'], dtype=



4. Check for Missing Values and Duplicates

```
import pandas as pd
```

```
# Load your data (example)
# df = pd.read_csv('your_dataset.csv')
```

```
# Check for missing values
print("Missing Values:\n", df.isnull().sum())
```

```
# Total missing values
print("\nTotal missing values:", df.isnull().sum().sum())
```

```
# Check for duplicates
duplicates = df.duplicated()
print("\nNumber of duplicate rows:", duplicates.sum())
```

```
# Optionally display duplicate rows
if duplicates.sum() > 0:
    print("\nDuplicate Rows:\n", df[duplicates])
```

```
# Scale numeric values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

```
# Split and train
X_train, X_test, y_train, y_test =
model = RandomForestClassifier()
model.fit(X_train, y_train)
```

```
# Save model and scaler
with open("churn_model.pkl", "wb")
    pickle.dump(model, f)
with open("churn_scaler.pkl", "wb")
    pickle.dump(scaler, f)
```

```
# Save column order for later
feature_cols = X.columns
```

[Use code with caution](#)

► Sources

Rate this answer

```

Missing Values:
  Name      0
  Age       4
  Marks     4
  Attendance 4
  Passed    0
dtype: int64

```

Total missing values: 12

Number of duplicate rows: 0

5. Visualize a Few Features

```

!pip install pandas
!pip install matplotlib
!pip install seaborn

```

```

Requirement already satisfied: pandas in /usr/local/lib/python3.10.12/dist-packages (2.2.3)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.10.12/dist-packages (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10.12/dist-packages (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10.12/dist-packages (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10.12/dist-packages (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10.12/dist-packages (1.17.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10.12/dist-packages (3.9.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10.12/dist-packages (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10.12/dist-packages (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10.12/dist-packages (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10.12/dist-packages (1.4.7)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.10.12/dist-packages (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10.12/dist-packages (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.10.12/dist-packages (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10.12/dist-packages (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10.12/dist-packages (2.9.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10.12/dist-packages (1.17.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10.12/dist-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in /usr/local/lib/python3.10.12/dist-packages (1.26.4)
Requirement already satisfied: pandas>=1.2 in /usr/local/lib/python3.10.12/dist-packages (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in /usr/local/lib/python3.10.12/dist-packages (3.9.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10.12/dist-packages (1.1.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10.12/dist-packages (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10.12/dist-packages (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.10.12/dist-packages (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10.12/dist-packages (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.10.12/dist-packages (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10.12/dist-packages (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10.12/dist-packages (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10.12/dist-packages (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10.12/dist-packages (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10.12/dist-packages (1.17.0)

```

```
# Step 1: Install and import required libraries
```

```

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files

```

```

# Step 2: Upload your dataset
uploaded = files.upload()

```

```

# Step 3: Load the dataset (replace 'your_file.csv' with actual fi
df = pd.read_csv(next(iter(uploaded)))

# Step 4: Preview the data (and column names)
print(df.head())
print(df.columns) # Print the column names to check if they match

# Step 5: Visualize numerical features against churn
# Adjust these features to match the actual column names in your c
features_to_plot = ['MonthlyCharges', 'TotalCharges', 'tenure']
# Example: If 'MonthlyCharges' is 'monthly_charges' in your data,

# Check if the features in features_to_plot are present in the Dat
for feature in features_to_plot:
    if feature not in df.columns:
        print(f"Error: Column '{feature}' not found in the DataFra
        # Attempt to find closest match using case-insensitive par
        closest_match = next((col for col in df.columns if feature
        if closest_match:
            print(f"Replacing '{feature}' with '{closest_match}'")
            features_to_plot = [closest_match if f == feature else
            else:
                print(f"No suitable replacement found for '{feature}'.
                features_to_plot = [f for f in features_to_plot if f !

for feature in features_to_plot:
    plt.figure(figsize=(8, 4))
    sns.histplot(data=df, x=feature, hue='Churn', kde=True, elemer
    plt.title(f'{feature} Distribution by Churn')
    plt.xlabel(feature)
    plt.ylabel('Density')
    plt.show()

# Step 6: Count plot of Churn
# Check if 'Churn' column exists
if 'Churn' in df.columns:
    plt.figure(figsize=(6, 4))
    sns.countplot(data=df, x='Churn')
    plt.title('Churn Distribution')
    plt.xlabel('Churn')
    plt.ylabel('Count')
    plt.show()
else:
    print("Error: Column 'Churn' not found in the DataFrame. Skip

```



No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset (5).csv

	Name	Age	Marks	Attendance	Passed
0	Alice	20.0	85.0	90.0	Yes
1	Bob	21.0	NaN	80.0	No
2	Charlie	NaN	78.0	NaN	Yes
3	David	22.0	90.0	85.0	Yes
4	Eve	20.0	88.0	95.0	No

Index(['Name', 'Age', 'Marks', 'Attendance', 'Passed'], dtype=Error: Column 'MonthlyCharges' not found in the DataFrame. Replacement No suitable replacement found for 'MonthlyCharges'. Please check Error: Column 'TotalCharges' not found in the DataFrame. Replacement No suitable replacement found for 'TotalCharges'. Please check Error: Column 'tenure' not found in the DataFrame. Replacement No suitable replacement found for 'tenure'. Please check your

6. Identify Target and Features

```
# Step 1: Import required libraries
import pandas as pd
from google.colab import files

# Step 2: Upload your dataset (CSV)
uploaded = files.upload()

# Step 3: Load the dataset
df = pd.read_csv(next(iter(uploaded)))
print("Data loaded successfully!")

# Step 4: Display basic info
print("\nDataset Head:")
print(df.head())

print("\nDataset Info:")
df.info()

# Step 5: Identify the target column
# Replace 'YourTargetColumnName' with the actual name of your target
target = 'Marks' # Example: If 'Marks' is your target variable, use it
print(f"\nTarget column: {target}")

# Step 6: Separate features from target
X = df.drop(columns=[target])
y = df[target]

print("\nFeatures (X) columns:")
print(X.columns.tolist())

# Step 7: Check feature types
numerical_features = X.select_dtypes(include=['int64', 'float64'])
categorical_features = X.select_dtypes(include=['object', 'bool'])

print("\nNumerical Features:")
print(numerical_features)

print("\nCategorical Features:")
print(categorical_features)
```

[Choose files](#) No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset (7).csv

Data loaded successfully!

Dataset Head:

	Name	Age	Marks	Attendance	Passed
0	Alice	20.0	85.0	90.0	Yes
1	Bob	21.0	NaN	80.0	No
2	Charlie	NaN	78.0	NaN	Yes
3	David	22.0	90.0	85.0	Yes
4	Eve	20.0	88.0	95.0	No

Dataset Info:

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 16 entries, 0 to 15

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	Name	16 non-null	object
1	Age	12 non-null	float64
2	Marks	12 non-null	float64
3	Attendance	12 non-null	float64
4	Passed	16 non-null	object

dtypes: float64(3), object(2)

memory usage: 772.0+ bytes

Target column: Marks

Features (X) columns:

['Name', 'Age', 'Attendance', 'Passed']

Numerical Features:

['Age', 'Attendance']

Categorical Features:



7.Convert Categorical Columns to Numerical

Step 1: Import libraries

```
import pandas as pd
```

```
from sklearn.preprocessing import LabelEncoder
```

Step 2: Upload the dataset

```
from google.colab import files
```

```
uploaded = files.upload()
```

Step 3: Load the dataset

```
df = pd.read_csv(next(iter(uploaded)))
```

Step 4: Identify categorical columns

```
categorical_cols = df.select_dtypes(include=['object', 'bool']).columns
print("Categorical columns:", categorical_cols)
```

Step 5: Handle missing values if any (optional)

```
df = df.dropna() # Or use df.fillna(method='ffill') / df.fillna(v
```

Step 6: Convert categorical columns to numeric using Label Encoder

```
le = LabelEncoder()
```

```
for col in categorical_cols:
```

```
df[col] = le.fit_transform(df[col])

print("\nDataset after encoding:")
print(df.head())
```



Choose files

No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset (8).csv

Categorical columns: ['Name', 'Passed']

Dataset after encoding:

	Name	Age	Marks	Attendance	Passed
0	0	20.0	85.0	90.0	1
3	1	22.0	90.0	85.0	1
4	2	20.0	88.0	95.0	0
8	3	24.0	92.0	92.0	1

```
df_encoded = pd.get_dummies(df, drop_first=True)
print(df_encoded.head())
```



	Name	Age	Marks	Attendance	Passed
0	0	20.0	85.0	90.0	1
3	1	22.0	90.0	85.0	1
4	2	20.0	88.0	95.0	0
8	3	24.0	92.0	92.0	1
12	4	21.0	83.0	80.0	0

8. One-Hot Encoding

```
# Step 1: Install and import required libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

```
# Step 2: Load your dataset
from google.colab import files
uploaded = files.upload() # Upload your CSV file
df = pd.read_csv(next(iter(uploaded))) # Automatically takes the
df.head()
```




Choose files No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset (9).csv

	Name	Age	Marks	Attendance	Passed
0	Alice	20.0	85.0	90.0	Yes
1	Bob	21.0	NaN	80.0	No
2	Charlie	NaN	78.0	NaN	Yes
3	David	22.0	90.0	85.0	Yes



9. Feature Scaling

```
# Import necessary libraries
import pandas as pd
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.model_selection import train_test_split
```

```
# ... (rest of your code) ...
```

```
# Handle missing values (if necessary)
```

```
# ----> Changed code <----
```

```
for column in df.select_dtypes(include=['number']).columns:
    df[column].fillna(df[column].mean(), inplace=True)
```

```
# ... (rest of your code) ...
```



<ipython-input-32-b0b3cee6c3da>:11: FutureWarning: A value is
The behavior will change in pandas 3.0. This inplace method wi

For example, when doing 'df[col].method(value, inplace=True)',

```
df[column].fillna(df[column].mean(), inplace=True)
```



10. Train-Test Split

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, accuracy_score
```

11. Model Building

```
# Uncomment if not already installed
# !pip install pandas scikit-learn matplotlib seaborn
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

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

```
# Use the uploaded file name directly
df = pd.read_csv(next(iter(uploaded))) # Automatically reads the
df.head()
```



Choose files

No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving students_dataset.csv to students_dataset (11).csv

	Name	Age	Marks	Attendance	Passed
0	Alice	20.0	85.0	90.0	Yes
1	Bob	21.0	NaN	80.0	No
2	Charlie	NaN	78.0	NaN	Yes
3	David	22.0	90.0	85.0	Yes



```
# Drop customer ID
if 'customerID' in df.columns: # Check if 'customerID' exists
    df.drop('customerID', axis=1, inplace=True)
else:
    print("Column 'customerID' not found. Skipping drop.")
```

```
# Replace missing values
# ----> Changed code <----
# Check if 'TotalCharges' column exists before processing
if 'TotalCharges' in df.columns:
    df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors=
    df['TotalCharges'].fillna(df['TotalCharges'].median(), inplace=
else:
    print("Column 'TotalCharges' not found. Skipping missing value
```

```
# ... (rest of your code) ...
```



Column 'customerID' not found. Skipping drop.

Column 'TotalCharges' not found. Skipping missing value replac



12. Evaluation

```
# Install required packages
!pip install seaborn --quiet
```

```
# Imports
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Assuming you have downloaded the file as 'telco-customer-churn.csv'
import pandas as pd
from google.colab import files

# Upload the CSV file
uploaded = files.upload() # Opens file dialog in Google Colab

# Load the dataset if uploaded successfully
if uploaded:
    df = pd.read_csv(next(iter(uploaded)))
    df.head()
else:
    print("Error: File 'telco-customer-churn.csv' not uploaded.")
```



Choose files

No file chosen

Upload widget is only available

when the cell has been executed in the current browser session. Please
 rerun this cell to enable



13. Make Predictions from New Input

```
# Example new customer data (same structure as training data before)
# Example new customer data (same structure as training data before)
new_customer = {
    'gender': 'Female',
    'SeniorCitizen': 0,
}
```

14. Convert to DataFrame and Encode

```
# Before applying StandardScaler, make sure categorical features are in X
# ... your existing code for loading and preprocessing data ...

# Assuming 'Name' and other categorical features are in X:
# Convert 'Name' and other categoricals to numerical using Label Encoder
# Option 1: Label Encoding (for binary categoricals like Yes/No)
from sklearn.preprocessing import LabelEncoder
for col in X.select_dtypes(include=['object']).columns: # Loop through columns
    if X[col].nunique() == 2: # If the column only has 2 unique values
        le = LabelEncoder()
        X[col] = le.fit_transform(X[col])

# Option 2: One-Hot Encoding (for multi-category features)
```

```
X = pd.get_dummies(X, columns=[c for c in X.select_dtypes(include=
# ... rest of your code ...

# Now apply StandardScaler
scaler = StandardScaler()
scaler.fit(X) # This should now work without the ValueError
```



StandardScaler



StandardScaler()

15. Predict the Final Grade

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error
from sklearn.preprocessing import LabelEncoder

# Load the dataset from UCI repository
# Original URL that was giving a 404 error:
# url = 'https://archive.ics.uci.edu/ml/machine-learning-databases

# Updated URL:
url = 'https://raw.githubusercontent.com/aniruddhachoudhury/Red-Wi
# This URL is a publicly available dataset that should work

df = pd.read_csv(url, sep=';')

# Show first few rows
df.head()
```



**fixed acidity,volatile acidity,citric acid,residual
sugar,chlorides,free sulfur dioxide,total sulfur
dioxide,density,pH,sulphates,alcohol,quality**

0	7.4,0.7,0.0,1.9,0.076,11.0,34.0,0.9978,3.51,0....
1	7.8,0.88,0.0,2.6,0.098,25.0,67.0,0.9968,3.2,0....
2	7.8,0.76,0.04,2.3,0.092,15.0,54.0,0.997,3.26,0...
3	11.2,0.28,0.56,1.9,0.075,17.0,60.0,0.998,3.16,...

16. Deployment-Building an Interactive App

```
!pip install streamlit pyngrok --quiet
```



44.3/44.3	kB	1.8
9.9/9.9	MB	54.3
6.9/6.9	MB	66.5
79.1/79.1	kB	4.1

```
%%writefile app.py
import streamlit as st
import pandas as pd
import numpy as np
import pickle

# Load model and scaler
model = pickle.load(open("model.pkl", "rb"))
scaler = pickle.load(open("scaler.pkl", "rb"))

st.title("📉 Customer Churn Prediction App")
st.write("Enter customer details to predict churn:")

# Input form
gender = st.selectbox("Gender", ["Male", "Female"])
SeniorCitizen = st.selectbox("Senior Citizen", [0, 1])
Partner = st.selectbox("Partner", ["Yes", "No"])
Dependents = st.selectbox("Dependents", ["Yes", "No"])
tenure = st.slider("Tenure (months)", 0, 72, 12)
MonthlyCharges = st.number_input("Monthly Charges", 0.0, 200.0, 70)
TotalCharges = st.number_input("Total Charges", 0.0, 10000.0, 2500)

# Preprocess inputs
data = {
```

Enter a prompt here



0/2000

Gemini can make mistakes, so double-check responses and use code with caution. [Learn more](#)