

```
In [1]: import gradio as gr
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import ollama

# Function to Perform EDA and Generate Visualizations
def eda_analysis(file_path):
    df = pd.read_csv(file_path)

    # Fill missing values with median for numeric columns
    for col in df.select_dtypes(include=['number']).columns:
        df[col].fillna(df[col].median(), inplace=True)

    # Fill missing values with mode for categorical columns
    for col in df.select_dtypes(include=['object']).columns:
        df[col].fillna(df[col].mode()[0], inplace=True)

    # Data Summary
    summary = df.describe(include='all').to_string()

    # Missing Values
    missing_values = df.isnull().sum().to_string()

    # Generate AI Insights
    insights = generate_ai_insights(summary)

    # Generate Data Visualizations
    plot_paths = generate_visualizations(df)

    return f"\n Data Loaded Successfully!\n\n Summary:\n{summary}\n\n Missing Values:\n{missing_values}\n\n AI Insights:\n{insights}"

# AI-Powered Insights using Mistral-7B (Ollama)
def generate_ai_insights(df_summary):
    prompt = f"Analyze the dataset summary and provide insights:\n\n{df_summary}"
    response = ollama.chat(model="deepseek-coder", messages=[{"role": "user", "content": prompt}])
    return response['message']['content']

# Function to Generate Data Visualizations
def generate_visualizations(df):
```

```

plot_paths = []

# Histograms for Numeric Columns
for col in df.select_dtypes(include=['number']).columns:
    plt.figure(figsize=(6,4))
    sns.histplot(df[col], bins=30, kde=True, color="blue")
    plt.title(f"Distribution of {col}")
    path = f"{col}_distribution.png"
    plt.savefig(path)
    plot_paths.append(path)
    plt.close()

# Correlation Heatmap (only numeric columns)
numeric_df = df.select_dtypes(include=['number'])
if not numeric_df.empty:
    plt.figure(figsize=(8,5))
    sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
    plt.title("Correlation Heatmap")
    path = "correlation_heatmap.png"
    plt.savefig(path)
    plot_paths.append(path)
    plt.close()

return plot_paths

# Gradio Interface
demo = gr.Interface(
    fn=eda_analysis,
    inputs=gr.File(type="filepath"),
    outputs=[gr.Textbox(label="EDA Report"), gr.Gallery(label="Data Visualizations")],
    title="📊 LLM-Powered Exploratory Data Analysis (EDA)",
    description="Upload any dataset CSV file and get automated EDA insights with AI-powered analysis and visualizations"
)

# Launch the Gradio App
demo.launch(share=True)

```

* Running on local URL: <http://127.0.0.1:7862>

* Running on public URL: <https://29f93bb8cddb31aa24.gradio.live>

This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from the terminal in the working directory to deploy to Hugging Face Spaces (<https://huggingface.co/spaces>)

Out[1]:

In []:

In []: