Disclaimer and caution: This script/file was generated via generative AI

Thursday February 13, 2025, without prejudice

Prompt:

'For a given data set available in a CSV format, write a python code that does the following:

1. Loads the CSV data to a pandas data frame - the first row of the CSV contains headers.
2. Save images to disk of histograms of the data distribution for continuous 'Age' and discrete 'Gender' and 'Selector'.
3. Save a correlation heatmap of the data set.
4. Identify the top n features with highest absolute correlation with the attribute `Selector`.'

Response:

Here's a well-structured Python code that fulfills your requirements using the pandas, matplotlib, and seaborn libraries. Make sure you have these libraries installed in your environment. You can install them using pip if you haven't done so: bash

```
pip install pandas matplotlib seaborn
```

Now, here's the complete code:

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

def load_csv_to_dataframe(file_path):
    """
    Load CSV data into a pandas DataFrame.

    Parameters:
        file_path (str): The path to the CSV file.

    Returns:
        pd.DataFrame: DataFrame containing the loaded data.
    """
    try:
        df = pd.read_csv(file_path)
        return df
    except FileNotFoundError:
        print(f"Error: The file '{file_path}' was not found.")
        return None
    except pd.errors.EmptyDataError:
        print("Error: The file is empty.")
        return None
    except pd.errors.ParserError:
        print("Error: Could not parse the file.")
        return None
```

```
def plot_histograms(df):
    """
    Plot histograms for the 'Age' continuous attribute and 'Gender', 'Selector'
discrete attributes.

    Parameters:
        df (pd.DataFrame): The DataFrame containing the data.
    """
    plt.figure(figsize=(15, 5))

    # Histogram for 'Age'
    plt.subplot(1, 3, 1)
    sns.histplot(df['Age'], bins=30, kde=True, color='blue')
    plt.title('Age Distribution')
    plt.xlabel('Age')
    plt.ylabel('Frequency')

    # Histogram for 'Gender'
    plt.subplot(1, 3, 2)
    sns.countplot(data=df, x='Gender', palette='Set2')
    plt.title('Gender Distribution')
    plt.xlabel('Gender')
    plt.ylabel('Count')

    # Histogram for 'Selector'
    plt.subplot(1, 3, 3)
    sns.countplot(data=df, x='Selector', palette='Set3')
    plt.title('Selector Distribution')
    plt.xlabel('Selector')
    plt.ylabel('Count')

    # Save the histograms to a file
    plt.tight_layout()
    plt.savefig('histograms.png')
    plt.close()

def plot_correlation_heatmap(df):
    """
    Plot a heatmap of the correlation matrix of the DataFrame.

    Parameters:
        df (pd.DataFrame): The DataFrame containing the data.
    """
    plt.figure(figsize=(12, 8))
    correlation_matrix = df.corr()

    sns.heatmap(correlation_matrix, annot=True, fmt=".2f", cmap='coolwarm',
square=True)
    plt.title('Correlation Heatmap')

    # Save the heatmap to a file
    plt.savefig('correlation_heatmap.png')
    plt.close()
```

```
def get_top_features_with_selector(df, target='Selector', top_n=5):
    """
    Identify the top N features with the highest absolute correlation with the
specified target attribute.

    Parameters:
        df (pd.DataFrame): The DataFrame containing the data.
        target (str): The target attribute to check correlation against.
        top_n (int): The number of top features to return.

    Returns:
        list: List of top N feature names with the highest absolute correlation.
    """
    correlation_matrix = df.corr()

    # Get the absolute correlation values with the target attribute
    target_correlation = correlation_matrix[target].abs()

    # Sort the correlations and get the top N features
    top_features = target_correlation.sort_values(ascending=False).head(top_n +
1).index.tolist()

    # Exclude the target itself from the list
    top_features.remove(target)

    return top_features

# Main execution
if __name__ == "__main__":
    # Load the CSV file into a DataFrame
    file_path = 'data.csv'  # Replace with your CSV file path
    df = load_csv_to_dataframe(file_path)

    if df is not None:
        # Plot histograms
        plot_histograms(df)

        # Plot correlation heatmap
        plot_correlation_heatmap(df)

        # Get top features with respect to 'Selector'
        top_features = get_top_features_with_selector(df)
        print("Top 5 features with highest absolute correlation with 'Selector':",
top_features)
```

Here's a more concise answer:

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

```
# Load the CSV data into a pandas data frame
data = pd.read_csv('https://cf-courses-data.s3.us.cloud-object-
storage.appdomain.cloud/IBMSkillsNetwork-AI0273EN-
SkillsNetwork/labs/v1/m2/data/ILPD.csv')

# Save histograms of data distribution for 'Age', 'Gender', and 'Selector'
plt.figure()
data['Age'].hist()
plt.savefig('age_histogram.png')

plt.figure()
data['Gender'].hist()
plt.savefig('gender_histogram.png')

plt.figure()
data['Selector'].hist()
plt.savefig('selector_histogram.png')

# Save correlation heatmap of the data set
plt.figure(figsize=(12,8))
corr = data.corr()
sns.heatmap(abs(corr), annot=True)
plt.savefig('correlation_heatmap.png', bbox_inches='tight')

# Identify top 5 features with highest absolute correlation with 'Selector'
correlation_with_selector = corr['Selector'].abs().sort_values(ascending=False)
top_5_features = correlation_with_selector[1:6]
# Exclude 'Selector' itself
print(top_5_features)
```

We then want to run this in: https://colab.research.google.com

We now get nice pretty charts