GitHub Project: Hair Fall Auto Analysis

Project Title: Hair Fall Auto Analysis

This project automates the process of analyzing sensor-based data related to hair fall. It uses Python to dynamically generate a Jupyter Notebook which reads a CSV file, performs statistical analysis, visualizes the data, and optionally runs the notebook cells sequentially in Visual Studio Code (VS Code). The tool is aimed at automating repetitive data analysis tasks with minimal human intervention.

1. Auto-Configuration

In this section, file paths are set up for the script directory, the dataset (CSV), and the output notebook. It uses Python's `os` module to get absolute paths so the script can be executed from any location:

```python
SCRIPT_DIR = os.path.dirname(os.path.abspath(__file__))
CSV_FILE = os.path.join(SCRIPT_DIR, "Hair_Fall_Sensor_Dataset.csv")
NOTEBOOK_NAME = os.path.join(SCRIPT_DIR, "Hair_Fall_Auto_Analysis.ipynb")
```

2. Notebook Creation

The function `create_self_running_notebook()` builds a new Jupyter Notebook programmatically using `nbformat`. It inserts:

- A Markdown cell with the title and timestamp
- A Python code cell for loading the CSV file
- A code cell for statistical summaries, value counts, and visualization

```python
df = pd.read_csv('Hair_Fall_Sensor_Dataset.csv')
display(df.describe())
display(df['label'].value_counts())
sns.histplot(df['motion'], kde=True)
```

GitHub Project: Hair Fall Auto Analysis

```

## 3. Sequential Execution in VS Code

To automate execution of notebook cells, the script:

- Detects the VS Code executable path

- Opens the notebook file using subprocess

- Simulates keyboard commands to run each cell sequentially using VS Code's CLI options

This is an advanced technique to ensure fully automated execution without manual interaction. Error handling is in place if VS Code or the Jupyter extension is not installed.

## 4. Main Execution Logic

The `__main__` section orchestrates the complete workflow:

1. Checks for the CSV file

2. Calls the notebook creation function

3. Executes the notebook cells if possible

It provides user feedback via printed messages and guides the user through troubleshooting in case of failure:

- Ensure VS Code is installed

- Install Jupyter extension in VS Code

- Manually run cells with Shift+Enter if needed
```