

PHP and Python Integration

24– October - 2024

*A. Manikanta
Iot Team*



Copy_of_Leave_policies.ipynb

The given file is a Jupyter notebook (IPython Notebook) that involves data analysis and machine learning using Python.

Here is a detailed breakdown of the code from the notebook:

1. Imported Libraries:

- **pandas** and **numpy** are for data manipulation and numerical calculations.
- **train_test_split**: Splits the data into training and test sets.
- **RandomForestClassifier** and **DecisionTreeClassifier**: Used for classification tasks.
- **CountVectorizer**: Converts text into a matrix of token counts for text processing.
- **make_pipeline**: Chains steps together in a machine learning pipeline.

2. Loading Data:

- Paths to the Excel files containing leave balance data and leave records for Vizag employees are defined.
- The data is loaded into pandas DataFrames.

3. Handling Missing Values:

ffill() is used to propagate the last valid observation forward to fill missing values in both datasets.

4. Converting Date Columns:

- Date columns in the `leave_vizag` dataset are converted to datetime format for easier manipulation and analysis.

5. Leave Frequency Calculation:

- A new column, `leave_frequency`, is created. It checks if the number of leaves is greater than 2 and if the leave was taken in September or October.

6. Creating a Reason Factor:

- This step generates a `reason_factor` column. If the leave reason includes sensitive keywords (like "death" or "sick"), it assigns a factor of 1, otherwise 0.

7. Merging Data:

- The leave data is merged with the leave balance data. A new column `total_leave_balance` is calculated as the sum of casual, sick, and compensatory leaves.

8. Checking Columns and Defining Target:

- The code checks if the necessary columns (`leave_frequency`, `reason_factor`) exist. It then defines a new target column `eligible_leave`, determining if the employee is eligible for the requested leave based on their leave balance.

9. Leave Request Approval Logic:

This portion contains the logic for approving or disapproving a leave request based on various conditions like leave balance, leave reason, and requested days:

- This logic calculates penalties for exceeding the leave balance and approves or disapproves leave based on the company's policies.

Output:

- Based on the conditions, it prints whether the leave is approved and calculates any penalties.

When using **shared hosting**, there are certain constraints, such as limited access to the server's configuration, restricted ability to install custom software, and often limited support for running non-PHP scripts (e.g., Python). Given these restrictions, the best approaches would need to work within the shared hosting environment. Here's an evaluation of the approaches in light of shared hosting:

1. AJAX to Trigger Python via PHP Endpoint (Best Option for Shared Hosting)

- **How it fits shared hosting:**
 - Most shared hosting services support running **PHP scripts**, and some even allow executing external commands (like Python) through functions like `exec()`, `shell_exec()`, or `passthru()`.
 - You don't need to install or configure Python on the hosting server manually if it's pre-installed, which is common on many shared hosting platforms.
- **Steps:**
 - Use AJAX to send the form data to a PHP script.
 - The PHP script can run the Python script using `shell_exec()` and return the output to the frontend.
- **Key Advantages:**
 - PHP handles the communication with Python, which makes it easier to work within the shared hosting limitations.
 - No need to reload the page thanks to AJAX, which improves the user experience.
- **Potential Limitations:**
 - Some shared hosting providers may restrict the execution of external scripts through `shell_exec()` or `exec()` for security reasons. You would need to verify with your hosting provider if these functions are allowed.

Why it's good for shared hosting: It works well within the shared hosting environment because PHP is widely supported and allowed on shared hosting servers, and many providers also have Python installed by default.

2. Using Python Backend (Flask/Django) with PHP Frontend (Not Suitable for Shared Hosting)

- **Why it doesn't fit:**
 - Shared hosting environments typically do not allow you to install and run custom web servers like Flask or Django.
 - You would need to configure a dedicated Python environment, which isn't possible on most shared hosting platforms.
- **Better alternative:** If you still want to use Flask/Django, you would need a **VPS (Virtual Private Server)** or **cloud hosting**, where you have full control over the server environment.

3. Using Python Shell Directly from PHP with Page Reload (Good Option for Shared Hosting, Simpler but Less Dynamic)

- **How it fits shared hosting:**
 - This approach uses basic PHP functionality to trigger the Python script directly, which is typically allowed on shared hosting.
 - It doesn't require JavaScript or AJAX, making it simpler to implement, although it results in a page reload after form submission.
- **Steps:**
 - The form submits the data to a PHP script.
 - PHP runs the Python script using `exec()` or `shell_exec()` and captures the output.
 - The page reloads, and the output is displayed to the user.
- **Advantages:**
 - Simpler than the AJAX approach because it avoids JavaScript and uses just PHP.
 - Suitable for shared hosting as it doesn't require additional configurations beyond basic PHP and Python capabilities.
- **Limitations:**
 - The user experience is less smooth due to the page reload, which might be annoying if frequent updates are required.
 - Performance could be slower if the Python script takes time to process the request.

Why it's good for shared hosting: This approach is straightforward and works well in shared hosting environments because it relies on basic PHP and Python capabilities, which are often allowed.

Conclusion:

For **shared hosting**, the two best approaches are:

1. AJAX to trigger Python
via PHP

2. Using Python Shell
Directly from PHP with
Page

You should check with the hosting provider to ensure that:

- Python is installed.
- `shell_exec()`, `exec()`, or `passthru()` are allowed (some shared hosts disable them for security reasons).