ABB WAQT HAI AGEY BARHNE KA

الخدمت
ALKHIDMAT

بنو قابل

اپنی دُنیا آپ پیدا کر

THE LARGEST SCHOLARSHIP PROGRAM FOR KARACHI'S YOUTH

Automating Jupyter Notebook Execution with Error Tracking

Umer Saeed

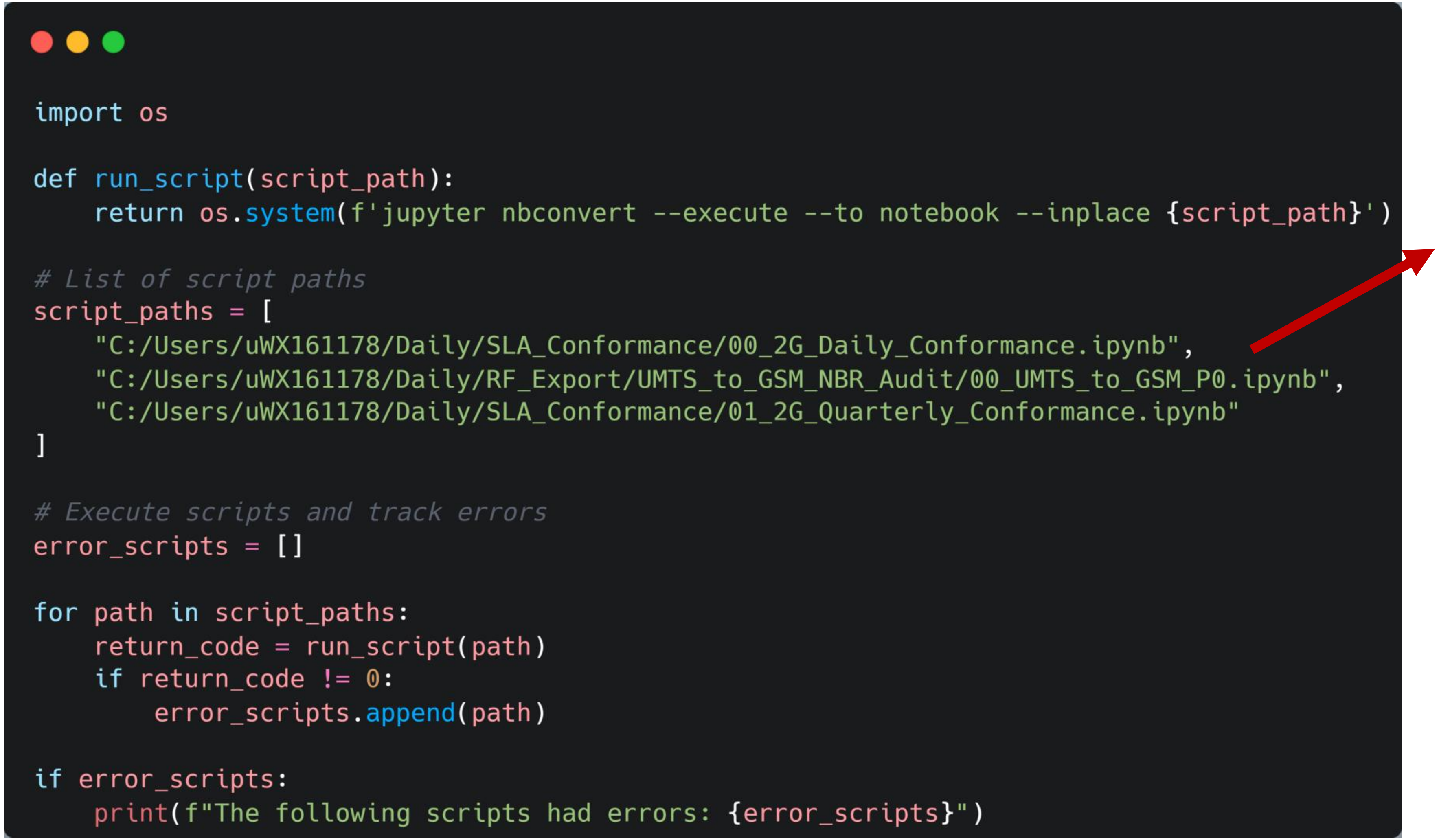
BSc Electrical Engineering, MS Data Science
Department of Information System
School of Business and Economics
University of Management & Technology, Lahore, Pakistan
Sr. RF Optimization and Planning Engineer
umersaeed1981@gmail.com

Automated Jupyter Notebook Execution and Tracking

- ❑ The purpose of this code is to;
    - ❑ Automate the execution of Jupyter notebooks.
    - ❑ Track and identify any errors that occur during the execution process.

The code achieves this by defining a function run_script(script_path) that executes Jupyter notebooks and then iterates through a list of script paths, executing each one and capturing any scripts that encounter errors. The identified error scripts are then printed at the end of the execution.

Automated Jupyter Notebook Execution and Tracking



```python
import os

def run_script(script_path):
    return os.system(f'jupyter nbconvert --execute --to notebook --inplace {script_path}')

# List of script paths
script_paths = [
    "C:/Users/uWX161178/Daily/SLA_Conformance/00_2G_Daily_Conformance.ipynb",
    "C:/Users/uWX161178/Daily/RF_Export/UMTS_to_GSM_NBR_Audit/00_UMTS_to_GSM_P0.ipynb",
    "C:/Users/uWX161178/Daily/SLA_Conformance/01_2G_Quarterly_Conformance.ipynb"
]

# Execute scripts and track errors
error_scripts = []

for path in script_paths:
    return_code = run_script(path)
    if return_code != 0:
        error_scripts.append(path)

if error_scripts:
    print(f"The following scripts had errors: {error_scripts}")
```

These Python scripts operate independently of each other.