**Running a Pytest on CAS502\_FINAL\_ST**

To ensure the reproducibility of the **CAS 502 \_ FINAL \_ ST** model and facilitate collaborative testing, we recommend following these steps. The latest version of the code is available on **GitHub**, allowing users to clone the repository (using git clone < repository- url>) and navigate to the project directory. Next, a virtual environment should be created and activated to ensure dependency isolation. We used “python -m venv cas 502 \_ env,” followed by source “cas 502 \_ env/bin/activate” (for macOS/Linux). If using Anaconda, as we did, users can set up the environment with “conda create --name cas 502 \_ env python = 3” and activate it via “conda activate cas 502 \_ env.” The required dependencies should be installed using “pip install -r requirements.txt,” which is located in the shared repository, or manually with “pip install pandas scikit- learn jupyter pytest.”

To validate the model, **Jupyter Notebook** can be tested by opening it in the terminal with “jupyter notebook,” restarting the kernel, clearing outputs, and running all cells sequentially to verify the model's functionality. Users can also convert the notebook into a Python script with “jupyter nbconvert CAS502 \_ FINAL \_ ST. ipynb” and execute “python CAS 502 \_ FINAL \_ ST. py” to confirm that the model runs correctly and produces a **predicted Length of Stay (LOS)**. Additionally, an automated validation pipeline is provided in “test\_CAS502\_FINAL\_ST.py,” which can be executed with “pytest test\_CAS502\_FINAL\_ST.py -v--tb = short” to check the preprocessing pipeline, model training, and prediction outputs.

Given that the code is hosted on GitHub, contributors are encouraged to report issues, submit pull requests for improvements, and ensure that any modifications are tested before merging. The test should pass if the setup is correct, ensuring that the model is properly executed and validated. Any failures should be addressed by verifying dataset integrity, as we had to do multiple times before running a successful test. Also, we recommend checking for missing dependencies and ensuring compatibility with **Python 3. 9.**

These steps should provide a comprehensive guideline for replicating and testing the model while maintaining transparency and version control through GitHub.