National College of Ireland

Configuration Report

MSc Data Analytics
Research Practicum

Annuncia Marena Yovan
Student ID: x23283491

School of Computing
National College of Ireland

Supervisor: Dr David Hamill

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Annuncia Marena Yovan		
Student ID:	X23283491		
Programme:	M.Sc. Data Analytics	**Year:**	2024-2025
Module:	Research Practicum		
Lecturer:	Dr David Hamill		
Submission Due Date:	11/08/2025		
Project Title:	Predicting Patients Discharge and Optimizing Hospital Bed Management Using AI Models		
Word Count:	999 **Page Count:** 4		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Annuncia Marena Yovan

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Annuncia Marena Yovan
X23283491

1 Introduction

1.1 Purpose of the System

This document provides configuration and operational instructions for the **AI-Powered Hospital Bed Management System**. The system is designed to address two critical challenges in hospital administration:

1. **Predicting Patient Length of Stay (LOS):** It utilizes machine learning models, specifically CatBoost and LightGBM, to accurately forecast the number of days a patient will require hospitalization.
2. **Optimizing Bed Allocation:** It leverages these predictions in a linear optimization model to efficiently assign incoming patients to appropriate wards, balancing patient needs with hospital resources and capacity.

The ultimate goal is to improve patient throughput, reduce wait times, and enhance overall hospital operational efficiency through data-driven decision-making.

1.2 Purpose of this Manual

This manual guides a technical user (such as a data scientist, IT administrator, or hospital analyst) through the process of setting up the environment, configuring the necessary components, and executing the full data processing and modeling pipeline as presented in the Research_work_final.ipynb notebook.

1.3 Target Audience

This document is intended for personnel with a working knowledge of Python, package management (pip), and Jupyter Notebook environments. Familiarity with machine learning concepts is beneficial but not strictly required to run the pre-configured pipeline.

2 System Requirements

2.1 Hardware Requirements

- **CPU:** Standard multi-core processor (e.g., Intel i5/i7, AMD Ryzen 5/7 or equivalent).
- **RAM:** Minimum 16 GB RAM recommended, especially if increasing the dataset sample size.
- **Storage:** Minimum 5 GB of free disk space for the dataset, libraries, and model artifacts.
- **GPU (Optional):** While not required for the default sampled dataset, a CUDA-enabled NVIDIA GPU will significantly accelerate model training if the system is scaled to the full dataset.

2.2 Software Requirements

- **Operating System:** Linux (recommended, as used in the notebook), macOS, or Windows.
- **Python:** Python 3.11 or a compatible version.

- **Environment Manager:** Jupyter Notebook, JupyterLab, Google Colab, or VS Code with the Python/Jupyter extension.
- **Package Installer:** pip.

3 Environment Setup and Installation

3.1 Create a Virtual Environment (Recommended)

To avoid conflicts with other Python projects, it is highly recommended to create a dedicated virtual environment.

```
# 1. Create the virtual environment
python3 -m venv hospital_env

# 2. Activate the environment
# On Linux/macOS:
source hospital_env/bin/activate
# On Windows:
.\hospital_env\Scripts\activate
```

3.2 Install Required Libraries

Install all necessary Python libraries using the following command. This consolidates the packages installed in the notebook.

```
pip install pandas numpy matplotlib seaborn scikit-learn catboost lightgbm autogluon pulp shap Kaggle
```

4 Data Acquisition and Configuration

The system requires the "2010 New York State Hospital Inpatient Discharge" dataset from Kaggle (New York State Hospital Inpatient Discharge).

4.1 Configure Kaggle API

1. Log in to your Kaggle account. Go to **Account** -> **API** and click **Create New Token**. This will download a kaggle.json file.
2. Place the kaggle.json file in the expected directory. For Linux/macOS, this is ~/.kaggle/.

```
# Create the directory if it doesn't exist
mkdir -p ~/.kaggle

# Move the downloaded file and set permissions
cp /path/to/your/kaggle.json ~/.kaggle/
chmod 600 ~/.kaggle/kaggle.json
```

Alternatively, place kaggle.json in the same directory as your Jupyter notebook.

4.2 Download and Unzip the Dataset

Run the following commands within a Jupyter Notebook cell or your terminal (after activating the environment) to download and extract the data.

```
# In a Jupyter cell:
!kaggle datasets download -d thedevastator/2010-new-york-state-hospital-inpatient-discharge
!unzip 2010-new-york-state-hospital-inpatient-discharge.zip
```

5 Execution Workflow

Execute the cells in the Research_work_final.ipynb notebook in sequential order. The key stages are outlined below.

5.1 Data Loading, EDA, and Preprocessing

- Run the initial cells to import libraries and load the hospital-inpatient-discharges-sparcs-de-identified-2010-1.csv file.
- Execute the Exploratory Data Analysis (EDA) cells to visualize data distributions and relationships.
- **Important:** The notebook processes a sample of **50,000 records** for computational efficiency. This is a configurable parameter (see Section 6).

5.2 Model Training

- Run the cells under the "Baseline Models" section to train Linear Regression, Random Forest, and SVM models.
- Proceed to the "CatBoost Model" and "LightGBM Implementation" sections to train the advanced gradient boosting models. The saved **CatBoost model (cat_model) is used in the subsequent optimization step.**

5.3 Bed Allocation Optimization

- Execute the cells in the "Optimization & Allocation Models" section.
- **Note:** The Ward feature is simulated for demonstration purposes. In a real-world scenario, this would come from the hospital's data.
- The script automatically adjusts ward capacities to ensure the optimization problem is solvable.
- Run the final cells in this section to solve the optimization problem and visualize the resulting bed assignments versus ward capacity.

5.4 Model Explainability

- Run the cells under the "SHAP for Global + Local Explainability" section to generate plots that explain the model's predictions.

6 Key Configuration Parameters

You can modify the following parameters in the notebook to customize the system's behavior.

- **Dataset Sample Size:**
  - **Location:** In the cell containing sample_df = df.sample(n=50000, random_state=42).
  - **Configuration:** Change the value of n to increase or decrease the sample size. A larger sample may yield a more accurate model but will require more RAM and processing time.
- **Model Hyperparameters:**
  - **Location:** In the CatBoostRegressor(...) and lgb.train(...) function calls.
  - **Configuration:** Parameters like iterations, learning_rate, depth (for CatBoost), and num_leaves (for LightGBM) can be tuned to improve model performance.
- **Optimization Constraints (Ward Capacity):**
  - **Location:** In the optimization section, the ward_capacity dictionary is defined.
  - **Configuration:** These values should be updated to reflect the **actual bed capacities** of the hospital's wards. The current implementation simulates

capacity based on patient counts (int(patient_counts[ward] * 1.5)); this line should be replaced with real data.

```
# Example with real capacities
ward_capacity = {
   'Cardiology': 50,
   'Surgery': 65,
   'Pediatrics': 30,
   'General Medicine': 80
}
```

7 Conclusion

By following this manual, you have successfully set up, configured, and executed the AI-Powered Hospital Bed Management System. The pipeline ingests raw patient discharge data, trains predictive models to forecast Length of Stay, and uses these predictions to run an optimization model for efficient bed allocation.

This system serves as a powerful proof-of-concept for integrating machine learning and operations research into hospital management. Further steps could include:

- Integrating the system with a live hospital database for real-time predictions.
- Deploying the model and optimization logic as a web-based dashboard for hospital administrators.
- Scaling the models to the full dataset on more powerful hardware for enhanced accuracy.
- Conducting hyperparameter tuning to further refine model performance.