A PROJECT REPORT

ON

HEALTHPULSE: DATADRIVEN STRATEGIES FOR

HEALTHCARE OPTIMIZATION

Submitted in the partial fulfilment of the requirements for the award of

JUNIOR DATA SCIENTIST

IN

COAPPS.AI

SUBMITTED BY

MOHAMMED ISMAILUDDIN

(ST.PETER’S ENGINEERING COLLEGE. HYDERABAD, TELANGANA)

PROGRAM OUTCOMES (Pos)

Coapps.ai interns are able to:

1: ENGINEERING KNOWLEDGE: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2: PROBLEM ANALYSIS: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using the first principles of mathematics, natural sciences, and engineering sciences.

3: DESIGN/DEVELOPMENT OF SOLUTIONS: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.

4: CONDUCT INVESTIGATIONS OF COMPLEX PROBLEMS: Use research-based knowledge and research methods including design of experiments, analysis, interpretation of data, and synthesis of the information to provide valid conclusions.

5: MODERN TOOL USAGE: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6: THE ENGINEER AND SOCIETY: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues, and the consequent responsibilities relevant to the professional engineering practice

7: ENVIRONMENT AND SUSTAINABILITY: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8: ETHICS: Apply ethical principles and commit to professional ethics and, responsibilities and norms of the engineering practice.

9: INDIVIDUAL AND TEAM WORK: Function effectively as an individual, and as a member or leader in diverse teams, and multidisciplinary settings.

10: COMMUNICATION: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and draft effective reports and design documentation, make an effective presentation, give, and receive clear instructions.

11: PROJECT MANAGEMENT AND FINANCE: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s work, as a member and leader in a team, to manage projects and in a multidisciplinary environment.

12: LIFE-LONG LEARNING: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadcast context of technological changes

DECLARATION

I declare that a Project entitled “HEALTHPULSE: DATADRIVEN STRATEGIES FOR HEALTHCARE OPTIMIZATION” is an Original Work submitted by the following member who have actively contributed and submitted in partial fulfilment for the award of Intern in “Internship in Junior Data Scientist”, at Coapps.Ai, Chennai, and this project work has not been submitted by me to any other Organization for the award of any kind of degree or Certificate

Data Submitted: 27-05-2024 SIGNATURE

Name: Mohammed Ismailuddin MOHAMMEDISMAILUUDIN

TABLE OF CONTENTS

1. INTRODUCTION
2. PROJECT OBJECTIVES
3. DATA DESCRIPTION
4. SYSTEM ARCHITECTURE
5. KEY FEAUTURES
6. METHODOLOGY
7. IMPLEMENTATION DETAILS
8. USE CASES
9. PROJECT CODE
10. CONCLUSION
11. FUTURE WORK

1.INTRODUCTION

The recent COVID-19 pandemic has underscored the importance of effective healthcare management. One crucial aspect of this is predicting the Length of Stay (LOS) for patients, which can greatly enhance hospital efficiency by enabling better resource allocation, such as room and bed planning. This project aims to develop a model to predict LOS for patients at the time of admission using data-driven strategies.

2. PROJECT OBJECTIVES

* Develop A Model To Predict LOS For Hospital Patients To Optimize Resource Management
* Improve Patient Outcomes: Utilize predictive analytics to identify at-risk patients and intervene early.
* Enhance Operational Efficiency: Optimize hospital resource allocation, including staff, equipment, and facilities.
* Support Policy Making: Provide data-driven insights to inform healthcare policies and funding decisions.

3. DATA DESCRIPTION

Train Data (train\_data.csv):

* Contains features related to patients, hospitals, and the LOS.
* The LOS is categorized into 11 classes ranging from 0-10 days to more than 100 days.

Datadictionary(train\_data\_dictionary.csv):

* Provides detailed information about the features in the training data.

Test Data (test\_data.csv):

* Contains features related to patients and hospitals.
* Requires predictions for the LOS for each case\_id.

4. SYSTEM ARCHITECTURE

OVERVIEW

The system architecture for the HealthPulse project comprises several key components, from data ingestion to model deployment.

4.1 COMPONENTS

DATA INGESTION:

Collect and store datasets (train\_data.csv, train\_data\_dictionary.csv, test\_data.csv).

DATA PREPROCESSING:

Handle missing values, encode categorical variables, normalize features, and perform feature engineering.

MODEL DEVELOPMENT:

Split data, select models, train models, evaluate, and perform hyperparameter tuning.

|  |
| --- |
| DATA INGESTION |
| DATA PREPROCESSING |
| MODEL DEVELOPMENT |
| MODEL EVALUATION |
| PREDICTION |
| Deployment |

5. KEY FEATURES

The datasets contain various features that can be broadly categorized into:

* Patient-related features: Demographics, medical history, etc.
* Hospital-related features: Hospital type, location, facilities, etc.

6. METHODOLOGY

6.1.DATA PREPROCESSING

* Data Cleaning: Handling missing values, outliers, and incorrect data entries.
* Feature Engineering: Creating new features from existing data to improve model performance.
* Normalization/Standardization: Ensuring all features are on a similar scale for better model training.

6.2. EXPLORATORY DATA ANALYSIS (EDA)

* Descriptive Statistics: Summarizing the central tendency, dispersion, and shape of the dataset’s distribution.
* Visualization: Using plots (histograms, scatter plots, box plots) to identify patterns, correlations, and anomalies in the data.

6.3. MODEL BUILDING

* Model Selection: Choosing appropriate models (e.g., Random Forest, Gradient Boosting, Neural Networks) for predicting LOS.
* Hyperparameter Tuning: Optimizing model parameters to enhance performance.
* Cross-Validation: Ensuring the model generalizes well to unseen data by splitting the training data into training and validation sets.

6.4. MODEL EVALUATION

* Metrics: Using accuracy, precision, recall, F1-score, and confusion matrix to evaluate model performance.
* Validation: Ensuring the model performs well on both training and validation datasets to prevent overfitting.

6.5. PREDICTION AND SUBMISSION

* Predictions: Generating LOS predictions for the test dataset.
* Submission Format: Creating a submission file with case\_id and predicted Stay.

7. IMPLEMENTATION DETAILS

1. LIBRARIES AND TOOLS

* Python 3.X
* Libraries: pandas, numpy, scikit-learn, matplotlib, seaborn

1. DATA PRECOCESSING
2. MODEL SELECTION AND TRAINING
3. PREDICTION AND SUBMISSION

8. USE CASES

* Resource Allocation: Predicting LOS can help hospitals allocate beds, rooms, and medical staff more efficiently.
* Treatment Optimization: Identifying high LOS risk patients early allows for tailored treatment plans to reduce their hospital stay.
* Infection Control: Minimizing LOS reduces the risk of infections spreading among patients, staff, and visitors.
* Logistics Management: Improved prediction aids in better planning for food, medication, and other logistical needs.
* Financial Planning: Helps in budgeting and financial forecasting by understanding patient turnover rates

9. PROJECT CODE

IN JUPITER NOTEBOOK:

1. . Import numpy as np

import pandas as pd

import os

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.preprocessing import LabelEncoder

2. health\_data = pd.read\_csv("C:\\Users\\91800\\Desktop\\Health care\\train\_data.csv")

3. health\_data.head()

4. health\_data.info()

5. health\_data.describe()

6. health\_data.shape

7. #Target variable distribution with custom figure size

plt.figure(figsize=(10, 6))

sns.countplot(x='Stay', data=health\_data)

plt.title('Distribution of Length of Stay')

plt.xticks(rotation=45)

plt.show()

8. # Example for exploring 'Department' with custom figure size

plt.figure(figsize=(10, 6))

sns.countplot(x='Department', data=health\_data)

plt.title('Distribution of Department')

plt.xticks(rotation=45)

plt.show()

9. #Instantiate LabelEncoder

label\_encoder = LabelEncoder()

# Fit and transform the 'Department' column

health\_data ['Department'] = label\_encoder.fit\_transform(health\_data['Department'])

health\_data ['Hospital\_type\_code']= label\_encoder.fit\_transform(health\_data['Hospital\_type\_code'])

health\_data ['Hospital\_region\_code']= label\_encoder.fit\_transform(health\_data['Hospital\_region\_code'])

health\_data['Ward\_Type']= label\_encoder.fit\_transform(health\_data['Ward\_Type'])

health\_data['Ward\_Facility\_Code']= label\_encoder.fit\_transform(health\_data['Ward\_Facility\_Code'])

health\_data['Type of Admission']= label\_encoder.fit\_transform(health\_data['Type of Admission'])

health\_data['Severity of Illness']= label\_encoder.fit\_transform(health\_data['Severity of Illness'])

health\_data['Age']= label\_encoder.fit\_transform(health\_data['Age'])

health\_data['Stay']= label\_encoder.fit\_transform(health\_data['Stay'])

1. # Splitting hospitals depending on their codes

HC\_dict = {}

for i in range(1,33):

HC\_dict[i]=0

for i in health\_data['Hospital\_code']:

HC\_dict[i]= HC\_dict[i]+1

1. #Target variable distribution with custom figure size

plt.figure(figsize=(10, 6))

sns.countplot(x='Stay', data=health\_data)

plt.title('Distribution of Length of Stay')

plt.xticks(rotation=45)

plt.show()

1. # Example for exploring 'Department' with custom figure size

plt.figure(figsize=(10, 6))

sns.countplot(x='Department', data=health\_data)

plt.title('Distribution of Department')

plt.xticks(rotation=45)

plt.show()

1. #Example for exploring correlation with numerical features with custom figure size

plt.figure(figsize=(10, 8))

correlation\_matrix = health\_data.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

1. #Example for exploring the relationship between 'Age' and 'Stay' with custom figure size

plt.figure(figsize=(12, 6))

sns.boxplot(x='Stay', y='Age', data=health\_data)

plt.title('Relationship between Age and Length of Stay')

plt.show()

1. #Example for visualizing the distribution of 'Available Extra Rooms in Hospital' with custom figure size

plt.figure(figsize=(10, 6))

sns.histplot(health\_data['Available Extra Rooms in Hospital'], bins=20, kde=True)

plt.title('Distribution of Available Extra Rooms in Hospital')

plt.show()

1. Features = ['Hospital\_code', 'City\_Code\_Hospital', 'Hospital\_type\_code', 'Available Extra Rooms in Hospital', 'Ward\_Facility\_Code', 'Bed Grade','Ward\_Type','Department','City\_Code\_Patient', 'Visitors with Patient', 'Age', 'Admission\_Deposit','Stay']

for data in Features:

plt.title(data)

sns.distplot(health\_data[data])

plt.show()

1. for data in Features:

plt.title(data)

health\_data[data].hist(bins = 50)

plt.show()

PROJECT CODE IN PYTHON:

#1.Load Data

import pandas as pd

# Load the dataset

data = pd. read\_csv("C:\\Users\\91800\\Desktop\\Health care\\data\\test\_data.csv")

print  (data.head())

#2.Data preprocessing

#basic data preprocessing

#handling missing values

data.fillna(method='ffill', inplace=True)

#convert categorical columns to numerical

data =pd.get\_dummies(data, drop\_first=True)

print(data.info())

#3.Feature Engineering

import pandas as pd

from sklearn.model\_selection import train\_test\_split

# Let's assume 'Age', 'patient id', 'Type of Admission' are features

# and 'case\_id' is the target variable.

# Selecting features and target variable

features = data[['Age', 'patientid', 'Type of Admission']]

target = data['case\_id']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.3, random\_state=42)

#4.model training

from sklearn.ensemble import RandomForestClassifier

# Initialize the model

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the model

model.fit(X\_train, y\_train)

#1.Load Data

import pandas as pd

# Load the dataset

data = pd. read\_csv("C:\\Users\\91800\\Desktop\\Health care\\data\\test\_data.csv")

print  (data.head())

#2.Data preprocessing

#basic data preprocessing

#handling missing values

data.fillna(method='ffill', inplace=True)

#convert categorical columns to numerical

data =pd.get\_dummies(data, drop\_first=True)

print(data.info())

#3.Feature Engineering

import pandas as pd

from sklearn.model\_selection import train\_test\_split

# Let's assume 'Age', 'patient id', 'Type of Admission' are features

# and 'case\_id' is the target variable.

# Selecting features and target variable

features = data[['Age', 'patientid', 'Type of Admission']]

target = data['case\_id']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(features, target, test\_size=0.3, random\_state=42)

#4.model training

from sklearn.ensemble import RandomForestClassifier

# Initialize the model

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

# Train the model

model.fit(X\_train, y\_train)

#5.model evalution

from sklearn.metrics import accuracy\_score, classification\_report

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

print(f'Classification Report:\n{report}')

#5.model evalution

from sklearn.metrics import accuracy\_score, classification\_report

# Make predictions

y\_pred = model.predict(X\_test)

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

report = classification\_report(y\_test, y\_pred)

print(f'Accuracy: {accuracy}')

print(f'Classification Report:\n{report}')

10. CONCLUSION

Predicting Patient Length Of Stay Is A Vital Component For Optimizing Hospital Management And Resource Allocation. This Project Showcases A Data-Driven Approach To Accurately Predict LOS Using Machine Learning Techniques, Thereby Enabling Better Healthcare Management And Improving Overall Hospital Efficiency.

11. FUTURE WORK

* Incorporate Additional Data Sources For Richer Feature Sets.
* Explore Advanced Models And Ensemble Techniques.
* Continuously Update The Model With New Data For Improved Accuracy.