**Healthcare Dataset Analysis and Predictive Modeling**

*Data Science Project Report*

**Prepared By:** Susan Dhungana

**Date:** 2081/08/27

**Tools Used:** Python (Pandas, Sweetviz, Scikit-Learn)

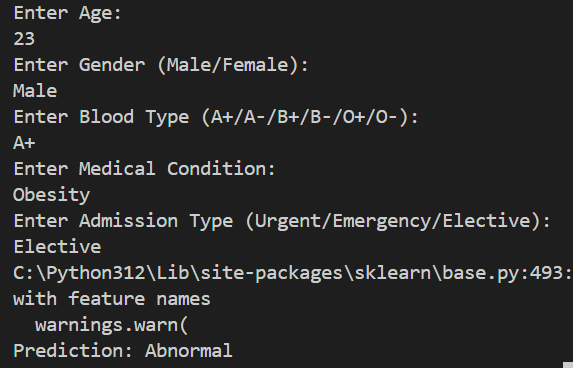
**Overview:**

This project involves the analysis and preprocessing of a healthcare dataset to build a predictive model using a Random Forest Classifier. The steps include data cleaning, feature engineering, handling categorical variables, and training the model.

**Code:**

* Importing all the libraries:
* import pandas as pd
* from sklearn.model\_selection import train\_test\_split
* import sweetviz as sv
* from sklearn.ensemble import RandomForestClassifier
* from sklearn.preprocessing import LabelEncoder
* Loading Dataset:
* data = pd.read\_csv('healthcare\_dataset.csv')
* print(data.head())
* print(data.shape)
* Data Cleaning and Preprocessing:
* data.columns = data.columns.str.strip().str.lower().str.replace(' ', '\_')
* print(data.columns)
* print(data.isnull().sum())
* data.dropna(inplace=True)
* data.drop\_duplicates(inplace=True)
* print(data.shape)
* Exploratory Data Analysis (EDA):
* report = sv.analyze(data)
* report.show\_html('report.html')
* Feature Engineering:
* data['Age\_Group'] = pd.cut(data['age'], bins=[0, 18, 60, 120], labels=["Child", "Adult", "Senior"], right=False)
* data = pd.get\_dummies(data, columns=['Age\_Group','gender', 'blood\_type', 'admission\_type', 'medical\_condition'], drop\_first=True)
* print(data.columns)
* Defining Features (X) and Target (Y):
* X = data.drop(columns=["name", "discharge\_date", "test\_results", "medication"])
* Y = LabelEncoder().fit\_transform(data['test\_results'])
* Data Transformation:
* for col in X.columns:
* if X[col].dtype == 'object':
* X[col] = LabelEncoder().fit\_transform(X[col])
* Splitting The Dataset:
* X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=42)
* Model Training:
* model = RandomForestClassifier(random\_state=42)
* model.fit(X\_train, Y\_train)
* Getting Input from User:
* age = int(input("Enter Age: \n"))
* gender = input("Enter Gender (Male/Female): \n")
* blood\_type = input("Enter Blood Type (A+/A-/B+/B-/O+/O-): \n")
* medical\_condition = input("Enter Medical Condition: \n")
* admission\_type = input("Enter Admission Type (Urgent/Emergency/Elective): \n")
* Input Data Preprocessing:
* input\_data = [[age, gender, blood\_type, medical\_condition, admission\_type]]
* input\_data = pd.DataFrame(input\_data, columns=['age', 'gender', 'blood\_type', 'medical\_condition', 'admission\_type'])
* input\_data['Age\_Group'] = pd.cut(input\_data['age'], bins=[0, 18, 60, 120], labels=["Child", "Adult", "Senior"], right=False)
* input\_data = pd.get\_dummies(input\_data, columns=['gender','medical\_condition', 'blood\_type', 'admission\_type'], drop\_first=True)
* input\_data = input\_data.reindex(columns=X\_train.columns, fill\_value=0)
* Prediction:
* if input\_data.isnull().values.any():
* print("Input data is missing values")
* else:
* input\_data = input\_data.values
* y\_pred = model.predict(input\_data)
* y\_pred= LabelEncoder().fit(data['test\_results']).inverse\_transform(y\_pred)
* print("Prediction:", y\_pred[0])

**Output:**



**Conclusion:**

This project highlights the development of a predictive healthcare model using machine learning, showcasing an end-to-end pipeline from data preprocessing to real-time predictions. Key aspects include effective data cleaning, feature engineering, and the use of a Random Forest Classifier for accurate predictions. The integration of tools like Sweetviz for EDA and real-time user input functionality makes the model practical and insightful. With potential for scalability, feature enhancement, and deployment, this project demonstrates how machine learning can contribute to improving decision-making and outcomes in healthcare.