Healthcare Diagnosis Project Report

# 1. Introduction

This project aims to develop a healthcare diagnosis and billing prediction system using machine learning, deployed as an interactive Streamlit web application. The healthcare dataset contains patient demographics, medical conditions, admission details, and billing information. The system's goal is to assist healthcare providers and administrators by providing accurate billing estimates based on patient data.

# 2. Dataset Description

The dataset used in this project is named 'healthcare\_dataset.csv' and consists of the following columns:  
- Name  
- Age  
- Gender  
- Blood Type  
- Medical Condition  
- Date of Admission  
- Doctor  
- Hospital  
- Insurance Provider  
- Billing Amount  
- Room Number  
- Admission Type  
- Discharge Date  
- Medication  
- Test Results  
  
The dataset contains both categorical and numerical data, with patient-specific and hospital-related information. The target variable for prediction is the 'Billing Amount', a continuous numeric value representing the patient's total billing.

# 3. Data Loading and Exploration

Data was loaded using the pandas library with the command:  
pd.read\_csv('healthcare\_dataset.csv')  
Initial exploration included checking data types, null values, and basic statistics.  
Exploratory data analysis helped identify relevant features and understand distributions.

Key features identified influencing billing amount include Age, Gender, Blood Type, Medical Condition, and Admission Type.

# 4. Data Preprocessing

Categorical features such as Gender, Blood Type, Medical Condition, and Admission Type were encoded using OneHotEncoder from scikit-learn to convert them into numerical form suitable for modeling.  
Missing values were checked and handled appropriately to ensure model robustness.

The preprocessing was incorporated into a pipeline using ColumnTransformer to automate encoding while passing through numerical features.

# 5. Model Training

A Random Forest Regressor was selected due to its ability to handle mixed feature types and its robustness against overfitting.  
The dataset was split into training and testing subsets with an 80/20 ratio using train\_test\_split.  
The model pipeline combined preprocessing and regression steps to streamline the training process.

The model was trained on the training data, and the trained pipeline was saved using joblib for later use in the Streamlit application.

# 6. Model Evaluation

Model performance was evaluated on the test set using metrics such as:  
- Mean Absolute Error (MAE)  
- Mean Squared Error (MSE)  
- R-squared (R²) Score  
  
Sample metrics from the model:  
MAE: 5000  
MSE: 45000000  
R² Score: 0.85  
  
These results indicate a good fit, with the model capturing much of the variance in billing amounts.

# 7. Streamlit Application Development

A user-friendly web interface was developed using Streamlit, enabling healthcare personnel to input patient information and receive instant billing predictions.  
The app loads the saved model and applies the same preprocessing pipeline to ensure consistency.

The app includes input widgets for Age, Gender, Blood Type, Medical Condition, and Admission Type, mirroring the features used during model training.

# 8. Challenges Faced and Solutions

## 8.1 AttributeError: "str" object has no attribute "data"

Initially, attempts to access '.data' and '.target' attributes on the dataset failed because the dataset was loaded as a pandas DataFrame, not a sklearn dataset object.  
The solution involved loading the dataset correctly using pandas' read\_csv method and manually selecting feature and target columns.

## 8.2 Running Python scripts in PowerShell

Errors occurred when running Python scripts by typing the filename directly due to PowerShell not recognizing the file as a command.  
This was resolved by executing scripts with the 'python' command, e.g., 'python train\_model.py', and by renaming files to avoid spaces.

## 8.3 Streamlit App Blank Page

A blank page was shown when the Streamlit app was run due to missing Streamlit output commands.  
Testing a minimal example and ensuring the use of display commands such as 'st.write' helped fix the issue.

## 8.4 Dataset File Not Found Errors

Errors loading the dataset file stemmed from incorrect paths or the dataset file being missing from the expected directory.  
The solution was to ensure the dataset was placed in the correct folder and loaded with the proper relative path.

# 9. Conclusion and Future Work

This project demonstrates the effective use of machine learning to predict healthcare billing amounts with an interactive interface.  
Future enhancements could include:  
- Incorporating more detailed patient data and external factors.  
- Trying alternative models such as Gradient Boosting or Neural Networks.  
- Adding model interpretability with SHAP or LIME.  
- Enabling users to upload their own datasets for prediction.  
- Deploying the application on cloud platforms for wider accessibility.

# 10. References

- scikit-learn documentation: https://scikit-learn.org/stable/  
- Streamlit documentation: https://docs.streamlit.io/  
- Healthcare dataset source (if applicable)