# PreMedix-Medical-Insurance-Prediction-System

**SYNOPSIS OF THE PROJECT**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

**BACHELOR OF TECHNOLOGY IN**

COMPUTER SCIENCE & ARTIFICIAL INTELLIGENCE

Batch

(2022-2026)

**SUBMITTED BY**

Rosy

12200724

**SUPERVISED BY**

Dr. Ridhi Kapoor

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING DAV UNIVERSITY**

**JALANDHAR-PUNJAB144012**

**JALANDHAR -144001, PUNJAB**

# Contents

# Chapter 1

# INTRODUCTION AND PROJECT OBJECTIVES

## 1.1 Background and Problem Statement

The global medical insurance sector is characterized by complex premium calculations based on various demographic and health-related factors. The conventional, rule-basedapproachoftenlacksthepredictiveaccuracyneeded in a dynamic market. This project, titled **Smart Medical Insurance Analysis & Premium Prediction (PreMedix System)**, addresses this gap by implementing a robust Machine Learning (ML) model to automate and optimize the premium estimation process. The primary objective is to develop a datadriven tool that provides accurate premium forecasts, thereby improving efficiency and reducing reliance on subjective manual assessment.

## 1.2 Industrial Training Context

The project was executed during the four-week industrial training period. The training focused on the practical application of software development methodologies, specifically the end-to-end Machine Learning pipeline—from data preprocessing and model selection to deployment using the Streamlit framework. This practical experience provided a real-world perspective on building MLOps-ready applications.

## 1.3 Project Objectives

The core objectives guiding the development of the PreMedix System were as follows:

1. **Develop a Data Management Module:** To create a secure and persistent mechanism for adding, storing, and retrieving comprehensive customer data (demographics and health metrics).

2. **Implement a High-Performance ML Model:** To select, train, and integrate an advanced regression model capable of accurately predicting medical insurance charges based on input features.

3. **Build a User-Friendly Application:** To deploy the integrated system as a multi-page web application using Streamlit, ensuring an intuitive interface for insurance analysts.

4. **Perform Risk Assessment:** To automatically categorize the predicted premium into defined risk levels (Low, Medium, High) for immediate analytical use.

5. **Ensure Data Persistence:** To maintain all customer records and prediction results persistently using a file-based storage system.

# Chapter 2

# SYSTEM ANALYSIS AND DESIGN

## 2.1 System Architecture

The PreMedix System adopts a classical three-tier architecture, customized for a modern data application:

1. **Presentation Tier (Front-End):** Built using the **Streamlit** framework, this layer handles all user interactions, data input forms, prediction display, and viewing of the customer database.
2. **Business/ApplicationTier(Back-EndLogic):** ConsistsofPythonscripts, Pandas for data manipulation, and the core prediction logic where the loaded Machine Learning model processes input data.
3. **Data Tier (Persistence):** Utilizes a simple CSV file (‘customers.csv‘) for persistent storage of customer records and prediction history.

### 2.1.1 Data Flow Analysis

The system’s operational integrity is maintained by a defined data flow, ensuring a separation of concerns between modules. The data flow starts at the Customer Input module, moves to the Persistence Layer, and is then consumed by the Prediction and Viewing modules.

- **InputFlow:** User data →_Add_Customer.py→ Data Validation → Append to ‘customers.csv‘.
- **Prediction Flow:** User query (Phone No.) → _Predict_Premium.py → Retrieve Data → Load ML Model → Predict Premium → Update ‘customers.csv‘.

## 2.2 Technology Stack

The project leverages a robust and widely-adopted open-source stack:

- **Programming Language:** Python 3.x
- **Web Framework (UI):** Streamlit (version 1.x)
- **Machine Learning:** XGBoost Regressor
- **Data Handling:** Pandas (for tabular data manipulation)
- **ModelSerialization:** Pickle(forloadingthepre-trainedmodel: insurancemodelf.p

# Chapter 3

# MACHINE LEARNING METHODOLOGY

## 3.1 Data Collection and Preprocessing

The project relied on a standard health insurance dataset comprising features known to influence medical charges. The preprocessing pipeline was critical for model training and integration:

### 3.1.1 Feature Engineering

The core features used for prediction were Age, BMI, Number of Children, and Smoker Status. The preprocessing steps involved:

- **HandlingCategoricalData:** Nominalcategoricalvariables, particularly the crucial ‘smoker‘ status (yes/no), were converted into numerical format (1/0) using techniques like One-Hot Encoding during the original model training phase. The application ensures consistency by transforming the Streamlit user input (e.g., ’yes’) into the expected numerical input (1) for the model at runtime.

- **Data Cleaning:** Ensured no missing values or outliers were present in the training set used to generate the final model artifact.

## 3.2 Model Selection: XGBoost Regressor

ExtremeGradientBoosting(XGBoost)wasselectedasthepredictivealgorithm due to its superior performance in structured, tabular data tasks and its ability to handle complex non-linear relationships between input features and the target variable (insurance charges).

### 3.2.1 Working Principle

XGBoost is an optimized distributed gradient boosting library. It sequentially builds decision trees, where each new tree corrects the errors (residuals) of the previous sequence of trees. Its key strengths include regularization (L1 and L2) to prevent overfitting and parallel processing capabilities, which contribute to fast and highly accurate predictions.

## 3.3 Model Training and Hyperparameter Tuning

The XGBoost model was trained on a comprehensive dataset. To achieve optimal performance and generalization capability, an exhaustive search technique, **Grid Search with Cross-Validation (CV)**, was utilized.

### 3.3.1 Hyperparameters Tuned

Key hyperparameters tuned included:

- **N_Estimators:** The number of boosting rounds (trees).
- **Max_Depth:** The maximum depth of a decision tree.
- **Learning_Rate:** The step size shrinkage used to prevent overfitting.

The GridSearch process identified the optimal combination of these parameters by minimizing the **Root Mean Squared Error (RMSE)** on the cross-validationfolds, resultinginthefinal, highlyaccuratemodelsavedasinsurancemodelf.p

## 3.4 Model Persistence

The best-performing model (‘final_model‘) was serialized using the Python pickle library and saved to the ‘model/insurancemodelf.pkl‘ file. This serialized artifact is the cornerstone of the application, as it allows the Streamlit app to load the trained intelligence instantly without needing to retrain the model every time the application starts.

# Chapter 4
# IMPLEMENTATION AND CODING

## 4.1 Application Structure and UI/UX

The Streamlit application (PreMedix System) is organized into a modular, multi-pagelayoutforclearnavigationandfunctionalseparation: app.py(main landingpage), _Add_Customer.py, _Predict_Premium.py, and_View_Customer.py. The UI/UX prioritizes clarity, utilizing standard Streamlit widgets for data input and display.

### 4.1.1 Customer Data Input Module (_Add_Customer.py)

This module implements the full data acquisition pipeline. It collects both auxiliary data (Name, Phone, Email, Address, Gender) and the core predictive features (Age, BMI, Children, Smoker).

```
# Data Persistence Logic (Excerpt from _Add_Customer.py) # Loads existing data,
concatenates the new record, and saves back.
def save_data(data):
    try: df = pd.read_csv(csv_path)
    except: df = pd.DataFrame(columns=[...]) # Define columns if file not found

    new_df = pd.DataFrame([data])
    df = pd.concat([df, new_df], ignore_index=True) df.to_csv(csv_path,
    index=False)
    # ... Streamlit success message
```

## 4.2 Premium Prediction Module (_Predict_Premium.py)

This module is the heart of the system, responsible for retrieving customer data and executing the ML prediction.

### 4.2.1 Model Integration and Execution

1. **Initialization:** Theinsurancemodelf.pklisloadedusingthepickle.load() function at the start of the script.

2. **Feature Mapping:** Customer data is retrieved using the phone number as the unique key. The extracted features are mapped and transformed (e.g., ’yes’ to 1 for smoker status) into the exact format the XGBoost model expects.

3. **Prediction:** ThepreparedinputDataFrameispassedtomodel.predict(input_df), returning the estimated insurance charge.

### 4.2.2 Risk Categorization and Persistence

Post-prediction, a simple business logic classifies the numeric premium into qualitative risk levels:

- **Low Risk:** Predicted Premium < ₹10,000

- **Medium Risk:** Predicted Premium < ₹20,000

- **High Risk:** Predicted Premium ≥ ₹20,000

The final predicted value and the calculated risk level are then saved back to the customer’s record in the ‘customers.csv‘ file, completing the end-to-end process.

# Chapter 5

# RESULTS AND CONCLUSIONS

## 5.1 Project Achievements

The PreMedix System successfully delivered on all stated objectives, providing a proof-of-concept for an intelligent insurance analysis tool.

- **Functional Accuracy:** The integrated XGBoost model provides highly accurate predictions, significantly reducing estimation time from hours to seconds.
- **Modularity:** The application's multi-page structure ensures a clear separation of concerns, making the system easy to maintain and scale.
- **Data Management:** The persistent CSV storage mechanism ensures all customer records and their associated prediction history are traceable and secure.

## 5.2 Key Results

The application demonstrates the substantial impact of key variables on premium prediction:

- **Smoker Status:** Remains the most significant determinant of high premiums, aligning with domain expectations. The model captured this non-linear dependency effectively.
- **BMI and Age:** These factors show a combined effect, where high BMI in older individuals significantly elevates the predicted premium.

## 5.3 Conclusion

The Industrial Training Project was successful in developing a functional, data-driven application that automates medical insurance premium estimation. The use of Python, Pandas, and XGBoost, deployed via Streamlit, provided a comprehensive learning experience in building a modern analytical tool. The PreMedix System serves as a robust prototype for future deployment in an insurance analysis environment, demonstrating how Machine Learningcanbedirectlyleveragedtoenhancebusinessefficiencyanddecisionmaking accuracy.

# Chapter 6

# FUTURE SCOPE AND RECOMMENDATIONS

## 6.1 Recommendations for Scalability

To transition the PreMedix System into a robust, enterprise-level application, the following recommendations are crucial:

### 6.1.1 Database Migration

The current CSV-based persistence layer must be migrated to a dedicated relational database (e.g., PostgreSQL or MySQL). This would resolve current limitations regarding concurrent access, ensure data integrity, and enable more complex querying and reporting features that are essential for auditing and regulatory compliance.

### 6.1.2 Enhanced Security and Authentication

The system currently lacks user authentication. A production-ready version must implement:

- **User Authentication:** Secure login using standard protocols.
- **Role-Based Access Control (RBAC):** Implementing user roles (e.g., Analyst, Admin) to control access to the prediction and customer viewing modules.

## 6.2 Model Improvement and Maintenance

### 6.2.1 Automated Retraining Pipeline

Prediction accuracy can degrade over time due to changes in market dynamics (concept drift). A future scope enhancement involves creating an automated MLOps pipeline to:

- Monitor the model’s prediction accuracy in production.

- Automatically trigger model retraining on new incoming data quarterly.
- Deploy the updated model seamlessly (A/B testing) without interruptingservice.

### 6.2.2 Incorporating External Factors

The model’s accuracy could be further improved by incorporating more localized or time-sensitive features, such as regional inflation rates, policy coverage details, or temporal health trends.

# Chapter 7

# APPENDICES

## 7.1 Bibliography / References

1. Python Documentation, *Official Website*.

2. Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

3. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

4. Streamlit Documentation, *Official Website*.

5. *Dataset Source:* The original health insurance charges dataset used for model training.

## 7.2 Data Dictionary of Key Features

- **age:** Age of primary beneficiary (Integer).

- **bmi:** Body mass index (Float). Used to determine if the individual is within a healthy weight range.

- **children:** Number of dependents covered by insurance (Integer).

- **smoker:** Smoker status (Categorical: ’yes’, ’no’). Critical independent variable for prediction.

- **region:** Policyholder’sresidentialarea(Categorical: ’northwest’, ’southeast’, etc.). Encoded for model input.

- **charges:** Thetargetvariable: Individualmedicalinsurancecosts(Float).