

## Medical Cost Prediction Web Application

### Overview

This project is a Flask-based web application that predicts annual medical costs based on user inputs such as age, sex, BMI, number of children, smoking status, and region. The application uses a pre-trained Random Forest model to make predictions and provides an interactive interface for users to enter details and view the results.

### Features

- User Authentication: Login system with a predefined username and password.
- Interactive UI: Users enter details, submit the form, and receive predictions dynamically.
- Machine Learning Integration: Uses a Random Forest model for medical cost prediction.
- Session Management: Ensures only authenticated users can access prediction features.
- AJAX-based Prediction: Real-time prediction updates without reloading the page.
- Error Handling: Displays meaningful error messages for invalid inputs.

### Tech Stack

Backend: Flask (Python)

Frontend: HTML, CSS, Bootstrap, jQuery

Machine Learning: Scikit-learn (Random Forest model)

Session Management: Flask Sessions

### Workflow

#### User Login:

The user enters a username and password.

If credentials match, they are redirected to index.html.

#### Navigation to Prediction Page:

A button on index.html redirects the user to predict.html.

#### Prediction Process:

The user enters medical details and submits the form.

AJAX sends the input data to the Flask backend (/predict).

The ML model predicts the medical cost and returns the result.

The predicted cost is displayed dynamically without page reload.

Session Management & Logout:

If a user is not logged in, they are redirected to the login page.

The user can log out at any time, which clears their session.

## Machine Learning Model Used in the Project

### Overview of the ML Model

This project uses a Random Forest Regressor to predict annual medical costs based on various input features. The model is trained on a dataset containing demographic and health-related factors, and it learns to estimate the cost of medical expenses based on patterns in the data.

### Machine Learning Workflow

#### 1. Dataset Used

The model is trained on the Medical Cost Personal Dataset, which includes the following features:

age: Age of the individual

sex: Gender (Male/Female)

bmi: Body Mass Index (BMI), an indicator of body fat

children: Number of dependents

smoker: Whether the person smokes (Yes/No)

region: Geographic region (northwest, southwest, northeast, southeast)

charges: The target variable (medical cost in USD)

#### 2. Data Preprocessing

Before training the model, the dataset undergoes the following preprocessing steps:

Handling Categorical Data:

"sex", "smoker", and "region" are converted to numerical values using One-Hot Encoding.

Feature Scaling:

Standardization of numerical features (age, bmi, children) to improve model performance.

Train-Test Split:

The dataset is split into 80% training data and 20% testing data to evaluate model accuracy.

### 3. Model Selection: Random Forest Regressor

The Random Forest Regressor is chosen for its robustness and ability to handle non-linear relationships. It works by training multiple decision trees on different parts of the dataset and averaging their predictions to improve accuracy and reduce overfitting.

Why Random Forest?

- ✓ Handles both numerical and categorical data efficiently.
- ✓ Works well with non-linear relationships.
- ✓ Reduces overfitting by averaging multiple decision trees.

Hyperparameters Used:

n\_estimators = 100 (Number of trees)

max\_depth = None (Full depth trees)

min\_samples\_split = 2

random\_state = 42 (For reproducibility)

### 4. Model Training

The model is trained using scikit-learn's RandomForestRegressor:

### 5. Model Evaluation

After training, the model is tested on unseen data to measure its performance:

### 6. Model Deployment in Flask

Once the model is trained and saved as random\_forest\_model.pkl, it is loaded into the Flask web application to make real-time predictions.

#### How It Works in the Web App:

- 1 User enters age, BMI, number of children, smoking status, and other details in the form.
- 2 The Flask app processes the input and passes it to the trained Random Forest model.
- 3 The model predicts the medical cost based on the given inputs.
- 4 The predicted medical cost is displayed on the web page in real time.

#### 7. Model Integration in Flask

The trained model is integrated into a Flask-based web application, which allows users to enter their details and get a medical cost prediction.

#### Flask Backend Implementation

The backend loads the model and serves predictions through a POST request.

#### 8. Frontend (HTML & CSS)

The frontend consists of an interactive web page where users enter their details. The Flask backend then processes the input and returns the predicted medical cost.

#### 9. Running the Application

To run the Flask application:

Ensure you have all dependencies installed:

```
flask numpy pandas scikit-learn
```

#### 10. Summary of Features

Machine Learning Model: Random Forest Regressor

Preprocessing Techniques: One-hot encoding, feature scaling

Web Framework: Flask (Python-based backend)

Frontend: HTML, CSS, JavaScript (for interactivity)

Deployment Ready: Flask API for model inference

This project provides an interactive way for users to predict their medical costs using an intuitive web interface powered by a Random Forest Regressor