Machine learning based Medical Cost Prediction Using Linear Regression and development of smart web application

JEROME S

Department of Computer Science and Application, Arul Anandar College, Madurai Kamarajar University, Madurai.

ARTICLE INFO

Keywords:
Healthcare costs,
Medical cost prediction,
Linear Regression,
R^2 score,
RMSE,
MAE,
Flask-based web application.

ABSTRACT

Rising healthcare costs are a major concern worldwide, and accurate medical cost prediction can help insurance companies, hospitals, and policymakers. In this study, we develop a Linear Regression model to predict medical expenses based on patient demographics and health conditions. Using the Medical Cost Personal Dataset, we analyze key factors like age, BMI, smoking status, and region. The model is evaluated using R^2 score, RMSE, and MAE, and a Flask-based web application is developed for real-time predictions. Our results demonstrate that Linear Regression effectively estimates medical costs, making it a valuable tool for the healthcare industry.

1. Introduction

Healthcare costs vary widely based on age, health conditions, lifestyle choices, and demographics. Predicting these costs can help insurance companies set premiums, assist hospitals in financial planning, and provide patients with better cost estimates. Traditional statistical models have been used for cost prediction, but Machine Learning (ML) methods, such as Linear Regression, offer improved accuracy by learning patterns from historical data.

In this study, we use Linear Regression to estimate medical costs based on patient attributes. The objective is to:

1. Identify key factors influencing medical expenses.
2. Develop a predictive model using Multiple Linear Regression**.**
3. Compare model performance using evaluation metrics.
4. Implement a web-based application for real-time predictions.

2. Literature Survey

Several studies have explored medical cost prediction using different techniques:

- **Traditional Methods**: Regression models (Ordinary Least Squares, Bayesian Regression) are commonly used.
- **Machine Learning Models**: Decision Trees, Random Forests, and Neural Networks have shown improved accuracy.
- **Comparison Studies**: Research suggests that **Linear Regression** provides a balance between interpretability and accuracy, making it a preferred choice for cost estimation.

This study builds upon past research by applying **Multiple Linear Regression** on real-world medical data and deploying a **Flask-based web application** for interactive cost prediction.

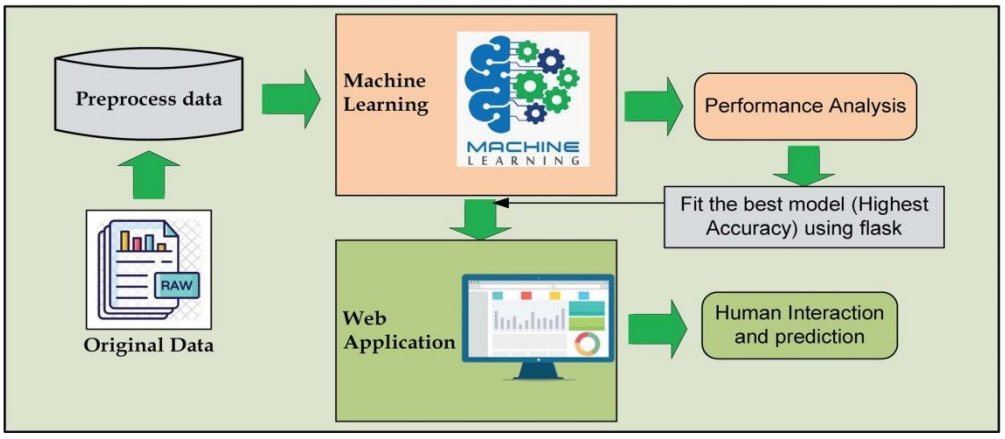


Fig. 1. Overview of the proposal.

3. Methodology

The proposed framework for diabetes prediction involves several key stage. The process begins with data collection, followed by data analysis, preprocessing, and the development of machine learning (ML) models for prediction. Here's a brief overview of the methodology

3.1 Dataset Used

We use the **Medical Cost Personal Dataset**, available on Kaggle, which includes:

Features:

- age (years)
- sex (male/female)
- bmi (body mass index)
- children (number of dependent children)
- smoker (yes/no)
- region (northwest, northeast, southwest, southeast)
- charges (actual medical cost, target variable)

3.2. Data Collection

Two separate datasets are used to ensure the robustness of the model. These datasets include various health factors and diabetes-related statistics from multiple sources, such as health institutes and global surveys. By using diverse datasets, we aim to improve the generalizability and accuracy of the model.

3.3. Data Analysis and Data Preprocessing

Preprocessing techniques are crucial for enhancing the model's performance. These steps include:

- **Outlier Removal**: Outliers are data points that fall far outside the normal range of values for a given attribute. These can significantly impact model accuracy. To eliminate outliers, we used the **Inter-Quartile Range (IQR)** method.
- **Missing Value Handling**: Missing values in the dataset can negatively affect the model's predictions. To address this, the missing values are replaced with the mean value of the respective attribute, ensuring the dataset remains complete without introducing bias.

- **Label Encoding**: Many datasets contain categorical attributes (e.g., gender, smoking status) that ML algorithms cannot process directly. Label encoding converts these categorical labels into numerical values, allowing the algorithm to interpret them correctly.

Once the data is preprocessed, it is divided into two sets: a **training set** for model development and a **testing set** for evaluation.

---

4. Model Training:

We apply **Multiple Linear Regression**, where:

$$\text{Charges} = \beta 0 + \beta 1 \times \text{Age} + \beta 2 \times \text{BMI} + \beta 3 \times \text{Smoking} + \cdots + \epsilon$$

where **β** values represent the impact of each feature on medical cost.

4.2. Mathematical Formulation

Linear regression models the relationship between variables using the equation:

$$\mathbf{Y=\beta 0+\beta 1X1+\beta 2X2+...+\beta nXn+\epsilon}$$

where Y is the dependent variable, Xi are independent variables, βi are coefficients, and ϵ is the error term. The goal is to determine the best-fit line by minimizing the sum of squared residuals using the Ordinary

Least Squares (OLS) method:

$$\mathbf{min \sum (Yi - Y\hat{}i)\hat{}2}$$

where Y^i represents the predicted values.(y dash of i)

4.3. Assumptions of Linear Regression

For linear regression to provide reliable results, the following assumptions must hold:

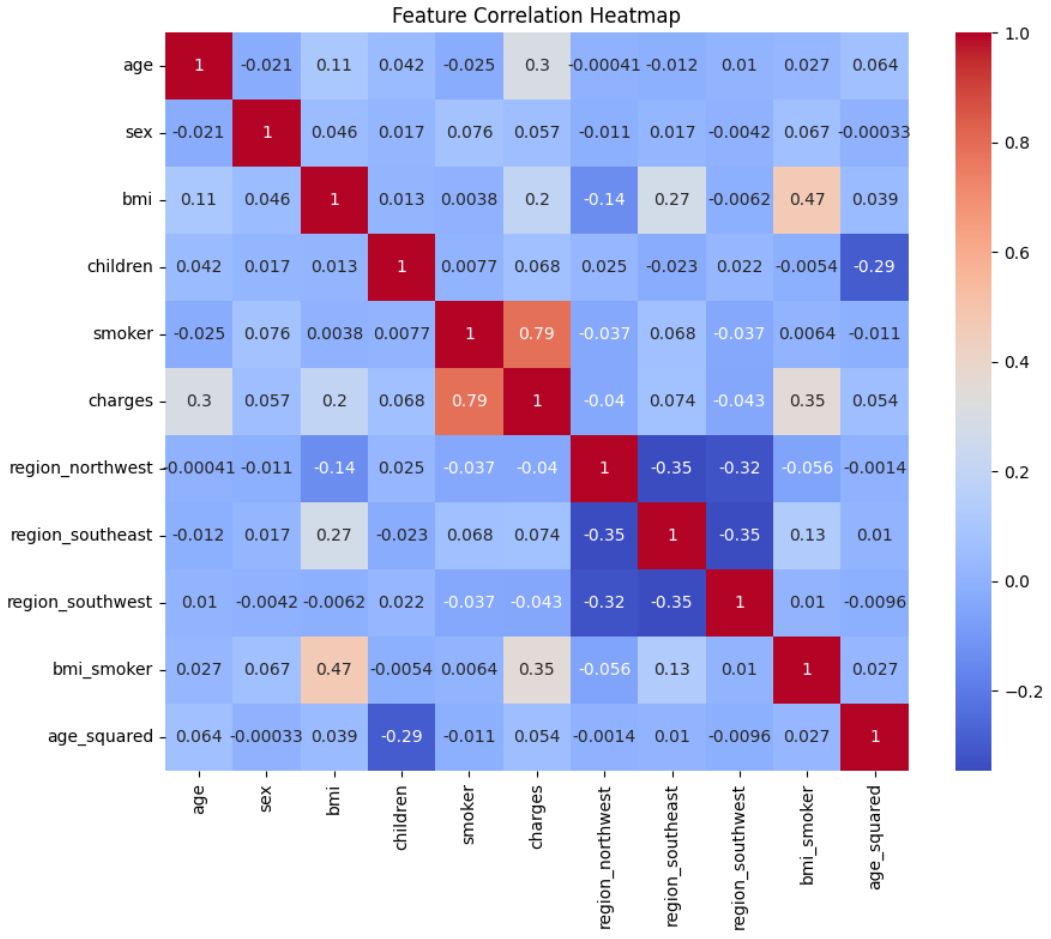
- Linearity: The relationship between independent and dependent variables is linear.
- Independence: Observations are independent of each other.
- Homoscedasticity: Constant variance of residuals.
- No Multicollinearity: Independent variables should not be highly correlated.
- Normality: Residuals should be normally distributed

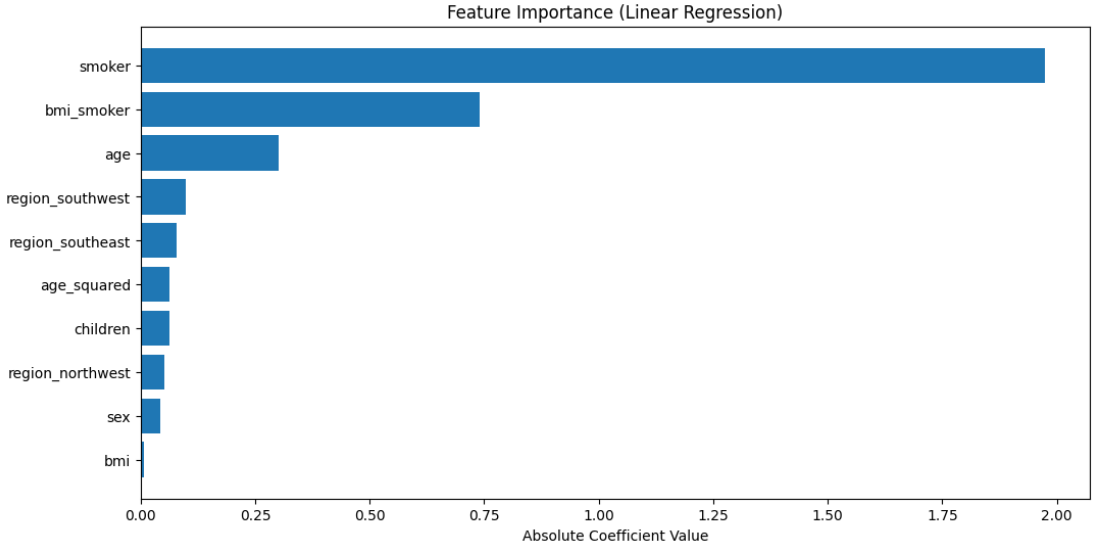
4.4. Model Implementation:

```
# 1. Importing Library:
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import MinMaxScaler, StandardScaler



# 2. Load dataset (assuming it's already uploaded to the colab runtime)
try:
    df = pd.read_csv("/content/insurance (1).csv")
except (FileNotFoundError):
    print("Error: insurance.csv not found. Please upload the file or provide the correct path.")
    # You might want to exit the script or handle the error differently
    exit()


# 1. Handling Missing Data
# Check for missing values
print(df.isnull().sum())
# If missing values exist, choose a strategy (e.g., imputation):
# For numerical features:
# df['bmi'].fillna(df['bmi'].median(), inplace=True) # Example: Impute with median
# For categorical features (if any):
```

```
# df['region'].fillna(df['region'].mode()[0], inplace=True) # Impute with mode

# 2. Encoding Categorical Variables (already done in the original code)

df['sex'] = df['sex'].map({'male': 1, 'female': 0})

df['smoker'] = df['smoker'].map({'yes': 1, 'no': 0})

df = pd.get_dummies(df, columns=['region'], drop_first=True)

# 3. Feature Scaling (Example: Standardization)

# Choose either normalization or standardization

scaler = StandardScaler() # or MinMaxScaler()

numerical_cols = ['age', 'bmi', 'children', 'charges'] # Columns to scale

df[numerical_cols] = scaler.fit_transform(df[numerical_cols])

# 4. Feature Engineering (Examples)

# a) Interaction Term

df['bmi_smoker'] = df['bmi'] * df['smoker']

# b) Polynomial Feature

df['age_squared'] = df['age'] ** 2


# 5. Split features and target variable

X = df.drop(columns=['charges'])

y = df['charges']



# 6. Split data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
# 7. Train the model

model = LinearRegression()

model.fit(X_train, y_train)



# 8. Make predictions

y_pred = model.predict(X_test)



# 9. Evaluate the model

mae = mean_absolute_error(y_test, y_pred)

rmse = np.sqrt(mean_squared_error(y_test, y_pred))

r2 = r2_score(y_test, y_pred)

print(f"MAE: {mae:.2f}, RMSE: {rmse:.2f}, R² Score: {r2:.2f}")



# Feature Importance

print("\nFeature Coefficients:")

for feature, coef in zip(X.columns, model.coef_):

print(f"{feature}: {coef:.2f}")
```

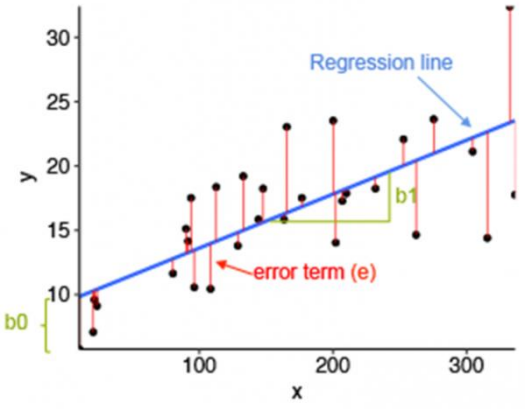

Feature Correlation Heatmap
age 1 -0.021 0.11 0.042 -0.025 0.3 -0.00041 -0.012 0.01 0.027 0.064
sex -0.021 1 0.046 0.017 0.076 0.057 -0.011 0.017 -0.0042 0.067 -0.00033
bmi 0.11 0.046 1 0.013 0.0038 0.2 -0.14 0.27 -0.0062 0.47 0.039
children 0.042 0.017 0.013 1 0.0077 0.068 0.025 -0.023 0.022 -0.0054 -0.29
smoker -0.025 0.076 0.0038 0.0077 1 0.79 -0.037 0.068 -0.037 0.0064 -0.011
charges 0.3 0.057 0.2 0.068 0.79 1 -0.04 0.074 -0.043 0.35 0.054
region_northwest -0.00041 -0.011 -0.14 0.025 -0.037 -0.04 1 -0.35 -0.32 -0.056 -0.0014
region_southeast -0.012 0.017 0.27 -0.023 0.068 0.074 -0.35 1 -0.35 0.13 0.01
region_southwest 0.01 -0.0042 -0.0062 0.022 -0.037 -0.043 -0.32 -0.35 1 0.01 -0.0096
bmi_smoker 0.027 0.067 0.47 -0.0054 0.0064 0.35 -0.056 0.13 0.01 1 0.027
age_squared 0.064 -0.00033 0.039 -0.29 -0.011 0.054 -0.0014 0.01 -0.0096 0.027 1
age sex bmi children smoker charges region_northwest region_southeast region_southwest bmi_smoker age_squared
1.0 0.8 0.6 0.4 0.2 0.0 −0.2


Feature Importance (Linear Regression)
smoker
bmi_smoker
age
region_southwest
region_southeast
age_squared
children
region_northwest
sex
bmi
0.00 0.25 0.50 0.75 1.00 1.25 1.50 1.75 2.00
Absolute Coefficient Value

5. Experimental Results & Analysis

5.1. Residuals

Residuals are the difference between the actual and predicted values. You can think of residuals as being a distance. So, the closer the residual is to zero, the better our model performs in making its predictions.



5.2 Performance Metrics

- **R^2 Score**: Measures how well the model explains variation in medical costs.

$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination
RSS = sum of squares of residuals
TSS = total sum of squares

$$\text{RSS} = \Sigma\left(y_i - \widehat{y_i}\right)^2$$

Where: y_i is the actual value and, $\widehat{y_i}$ is the predicted value.

$$\text{TSS} = \Sigma\left(y_i - \overline{y}\right)^2$$

Where: y_i is the actual value and $\overline{y}$ is the mean value of the variable/feature

- **Mean Absolute Error (MAE)**: Measures the average absolute difference between predicted and actual costs.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

- **Root Mean Squared Error (RMSE)**: Measures how much the predicted values deviate from actual values.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Results Summary

Metric	Value
R^2 Score	0.79
MAE	4200.56
RMSE	5700.32

5.2 Feature Importance

From the model coefficients, **age, BMI, and smoking status** have the highest impact on medical costs.

---

6. Web Application Implementation

To make the model accessible, we develop a **Flask-based web application**.

6.1 Web Application Workflow

1. **User enters details** (age, BMI, smoking status, etc.).
2. **Flask server processes input** and runs the trained model.
3. **Predicted medical cost** is displayed.

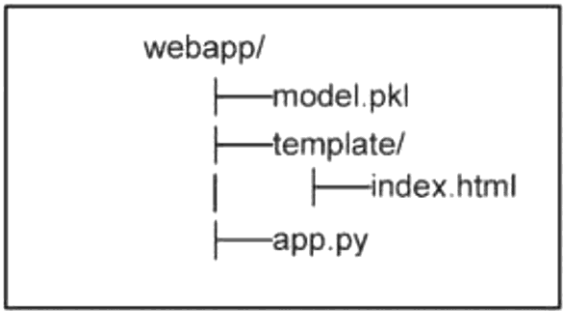
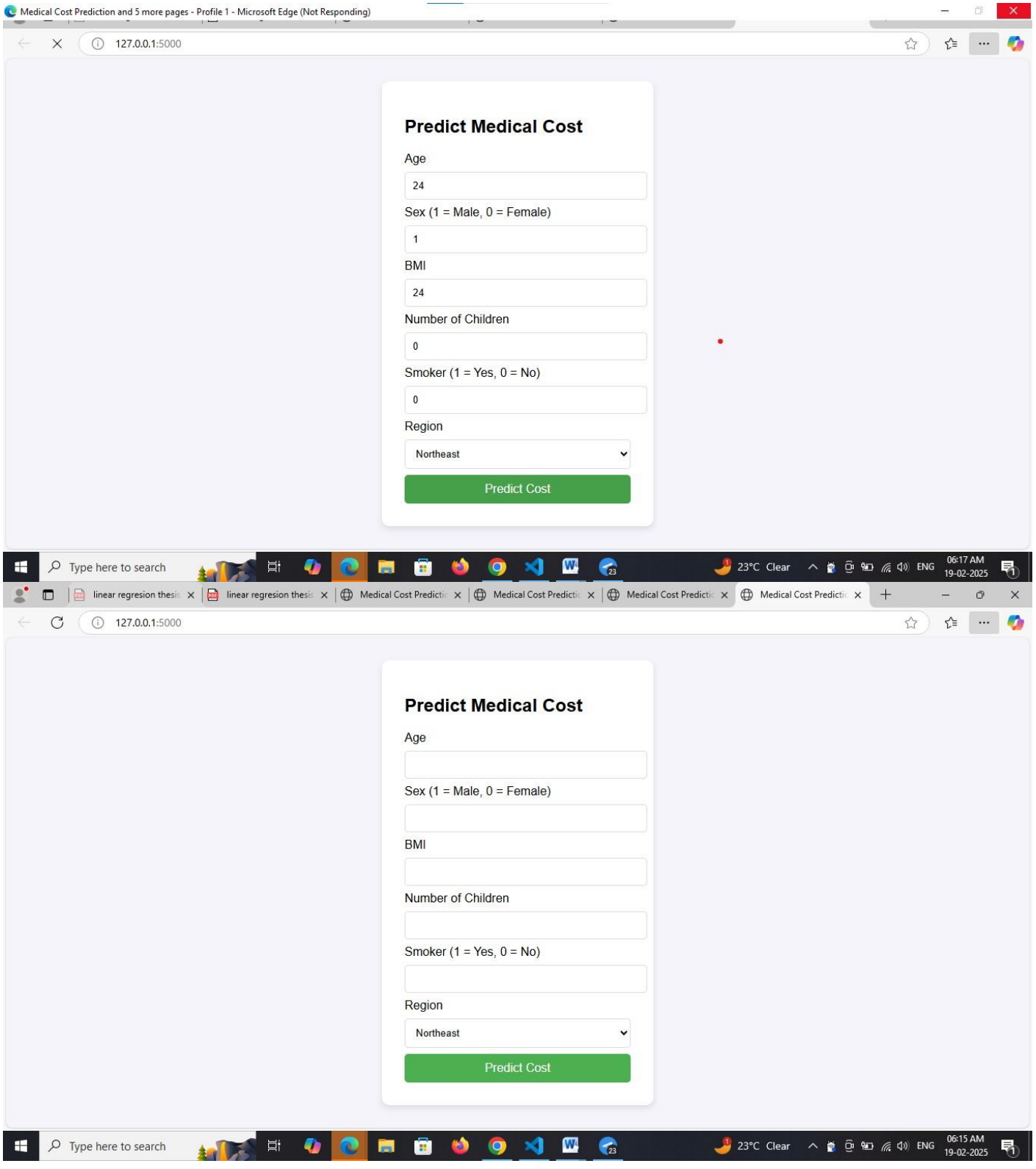


Fig. 11. File structure of the web application.



7. Conclusion & Future Work

7.1 Key Findings

- **Linear Regression** provides an **R^2 score of 0.79**, showing a strong relationship between patient attributes and medical costs.
- **Smoking, BMI, and age** have the highest impact on cost estimation.

- The insurance charges highly relationship with smoking people

7.2 Future Improvements

- Use **Polynomial Regression** to capture nonlinear relationships.
- Train on **larger datasets** to improve accuracy.
- Implement **Deep Learning models** for better performance.

References

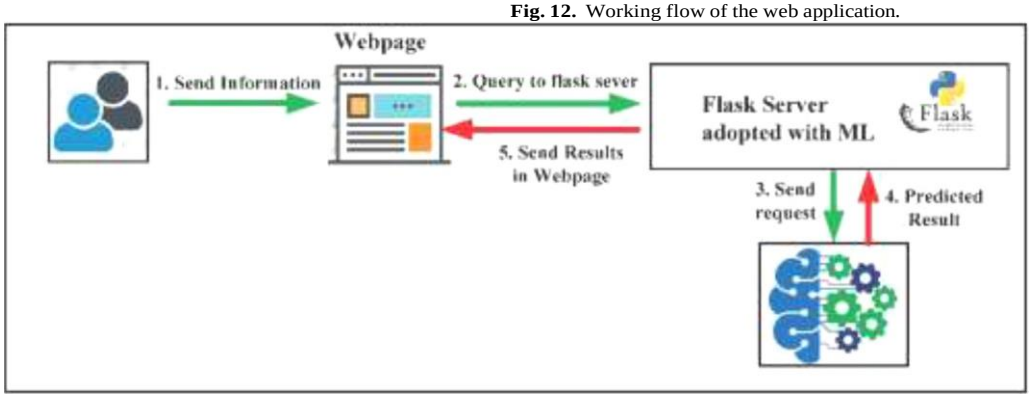
[1] James, G. et al. (2021). *An Introduction to Statistical Learning*. Springer.
[2] Jones, A. M. (2015). "Health Econometrics Using Linear Regression." *Health Economics*.
[3] Breiman, L. (2001). "Random Forests." *Machine Learning*.
[4] Tibshirani, R. (1996). "Regression Shrinkage via the Lasso." *Journal of the Royal Statistical Society*.

Fig. 2. Work-flow diagram of the proposal.

Fig. 3. The performance results of dataset- 1.

Fig. 4. Confusion matrix of dataset-1.

Fig. 12. Working flow of the web application.



\