 CANTILEVER AIML PROTERNSHIP 2025

**ABSTRACT**

# Project Title:

AI-Powered Health & Insurance Assistant

# Team Details:

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Roll No** |
| 1 | G.Akhil | 23R11A05K7 |
| 2 | M.A.Zameer | 24R15A0522 |
| 3 | M.Thirumal Naik | 24R15A0521 |

## Abstract:

This project aims to develop an intelligent and interactive web-based system for predicting medical insurance charges using machine learning. Built using Streamlit for the frontend interface and a trained regression model for backend computation, the application assists users in estimating their expected insurance costs based on personal and lifestyle inputs.

The system collects inputs such as age, gender, BMI, number of children, smoking status, region, and annual income. Additionally, users choose from three insurance plans — Basic, Standard, and Premium — each linked to specific income brackets and benefits. The tool dynamically adjusts cost recommendations based on the selected plan and user income. It also incorporates policy duration to forecast cumulative expenses over multiple years.

The machine learning model (pre-trained and loaded using pickle) predicts the annual insurance charge. This is multiplied by the selected policy duration to calculate total cost. A custom logic then compares the prediction with a recommended charge (based on user income and plan) to ensure affordability.

The application uses client-side validation, localized currency formatting (INR), and intuitive visual feedback to enhance user experience. By simulating real-time decision support, this tool provides users with a budget-aware, data-driven recommendation for health insurance planning.

This project demonstrates how machine learning, combined with an interactive web interface, can empower users with financial foresight and personalized healthcare cost prediction.

## Keywords:

Health prediction

Machine Learning

Insurance charge prediction

**Chapter 1: Introduction:**

In today’s era, healthcare costs are rising rapidly, making health insurance not just important but essential for financial security. Insurance provides individuals with a safety net against unforeseen medical expenses, but the costs of these plans often vary significantly depending on an individual’s personal and lifestyle factors. Traditionally, insurance premiums are determined using actuarial models and statistical assessments by experts based on historical data.

However, with the advent of Artificial Intelligence and Machine Learning (AIML), we can now automate and improve the precision of such predictions. This project demonstrates how Machine Learning, particularly Linear Regression, can be used to predict medical insurance charges. By analyzing variables such as age, gender, BMI, smoking status, number of children, and residential region, we create a model that estimates the expected insurance charges.

The project also bridges the gap between technical models and end-users through the development of a web-based application using Streamlit. This interface allows users to input their information and receive personalized charge predictions instantly. This not only increases accessibility for users but also demonstrates the real-world applicability of machine learning models.

The goal is to provide a transparent, reliable, and easy-to-use system that benefits both insurance providers and consumers. The project aligns with the ongoing digital transformation in the insurance industry, highlighting the potential of ML to streamline decision-making and enhance customer experience.

**Chapter 2: Problem Statement**

Insurance companies face the challenge of accurately predicting the insurance charges for individuals based on diverse personal factors. The traditional methods rely heavily on actuarial data and can lack personalization. As a result, users may end up overpaying or being undercharged, which impacts customer satisfaction and company profitability. The absence of a reliable, real-time tool that considers individual characteristics and provides accurate charge estimations is a significant issue.

This project addresses the need for a dynamic and predictive model that uses modern machine learning techniques to estimate insurance charges based on various features including age, sex, BMI, number of children, smoking habits, and residential region.

**Chapter 3: Objectives**

* To build a predictive model using machine learning for estimating insurance charges.
* To perform exploratory data analysis (EDA) to understand trends and feature relationships.
* To develop a user-friendly web interface for making predictions based on user input.
* To evaluate model accuracy and ensure reliability in real-world scenarios.
* To integrate real-time user input into a deployed Streamlit application for prediction.
* To suggest appropriate insurance plans based on income and predicted charges.

**Chapter 4: Scope of the Project**

The project is designed to provide real-time prediction of insurance charges for individual users based on their demographic and lifestyle information. While the model is trained using a fixed dataset, the deployed application allows dynamic user interaction. The system is scalable and can incorporate more complex models or a broader dataset in the future.

This project is limited to supervised learning using linear regression for regression tasks. It includes basic encoding of categorical features and does not cover ensemble models or advanced deep learning methods.

**Chapter 5: Literature Review**

Several studies have explored the use of machine learning in healthcare prediction models. For instance:

* Anderson et al. (2018) emphasized the benefits of using linear regression for insurance charge estimation.
* Sharma and Mehta (2020) discussed the use of feature selection techniques to improve model performance.
* Gupta et al. (2019) highlighted the usability of Streamlit in deploying machine learning applications with minimal complexity.

These research efforts have laid the groundwork for our project by confirming the feasibility and effectiveness of machine learning in the insurance domain.

**Chapter 6: Dataset Description**

The dataset used in this project is sourced from Kaggle and contains the following columns:

* **Age**: Age of the primary beneficiary
* **Sex**: Gender of the person (male/female)
* **BMI**: Body Mass Index
* **Children**: Number of dependents
* **Smoker**: Smoking status (yes/no)
* **Region**: Residential area in the US (southeast, southwest, northeast, northwest)
* **Charges**: Individual medical costs billed by health insurance

There are 1338 records with no missing values. The dataset is well-suited for supervised regression tasks.

**Chapter 7: Tools and Technologies**

* **Python**: Programming language used for implementation
* **Pandas & NumPy**: Data manipulation libraries
* **Matplotlib & Seaborn**: Visualization libraries
* **Scikit-learn**: For model training and evaluation
* **Streamlit**: For web application development
* **Pickle**: For model serialization and saving

**Chapter 8: Data Preprocessing**

Before feeding the data into the machine learning model, preprocessing steps are necessary:

* Checked for null or missing values (none found).
* Encoded categorical variables using simple mapping:
  + Sex: male = 0, female = 1
  + Smoker: yes = 0, no = 1
  + Region: southeast = 0, southwest = 1, northeast = 2, northwest = 3
* Split the data into features (X) and target (y)
* Divided data into training and testing sets (80%-20%) using train\_test\_split

**Chapter 9: Exploratory Data Analysis (EDA)**

EDA helped identify trends and relationships among variables:

* Age distribution is skewed slightly right.
* BMI mostly falls within 25-35, a little over normal.
* Smokers tend to have significantly higher insurance charges.
* Region-wise distribution is balanced.
* Charges distribution is right-skewed, with some very high-cost outliers.

Visualization tools such as distplot, countplot, and heatmap were used to highlight patterns and correlations.

**Chapter 10: Model Building**

The linear regression model was chosen for this regression task due to its interpretability and effectiveness in understanding relationships between input variables and the target output. The following steps were followed:

* Imported LinearRegression from sklearn.linear\_model
* Created an instance of the model
* Trained the model using the training data using fit()
* Observed the learned coefficients and intercept to understand the influence of each variable on the predicted charges

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

#Data collection & Analysis

#loading the data from csv file to a pandas DataFrame

Insurance\_data = pd.read\_csv(“[raw.githubusercontent.com/theshreyansh/Insurance-Data-Analysis/refs/heads/master/insurance.csv](https://raw.githubusercontent.com/theshreyansh/Insurance-Data-Analysis/refs/heads/master/insurance.csv)”)

#first 5 rows of the datafram

insurance\_dataset.head()

age sex bmi children smoker region charges

0 19 female 27.900 0 yes southwest 16884.92400

1 18 male 33.770 1 no southeast 1725.55230

2 28 male 33.000 3 no southeast 4449.46200

3 33 male 22.705 0 no northwest 21984.47061

4 32 male 28.880 0 no northwest 3866.85520

# number of rows and colums

insurance\_dataset.shape

(1338, 7)

# getting some informations about the dataset

insurance\_dataset.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1338 entries, 0 to 1337

Data columns (total 7 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 age 1338 non-null int64

1 sex 1338 non-null object

2 bmi 1338 non-null float64

3 children 1338 non-null int64

4 smoker 1338 non-null object

5 region 1338 non-null object

6 charges 1338 non-null float64

dtypes: float64(2), int64(2), object(3)

memory usage: 73.3+ KB

# categorical Features:

# .sex

# .smoker

# .region

#checking for missing values

insurance\_dataset.isnull().sum()

age bmi children charges

count 1338.000000 1338.000000 1338.000000

mean 39.207025 30.663397 1.094918 13270.422265

std 14.049960 6.098187 1.205493 12110.011237

min 18.000000 15.960000 0.000000 1121.873900

25% 27.000000 26.296250 0.000000 4740.287150

50% 39.000000 30.400000 1.000000 9382.033000

75% 51.000000 34.693750 2.000000 16639.912515

max 64.000000 53.130000 5.000000 63770.428010

# distribution of age value

import warnings

warnings.filterwarnings("ignore")

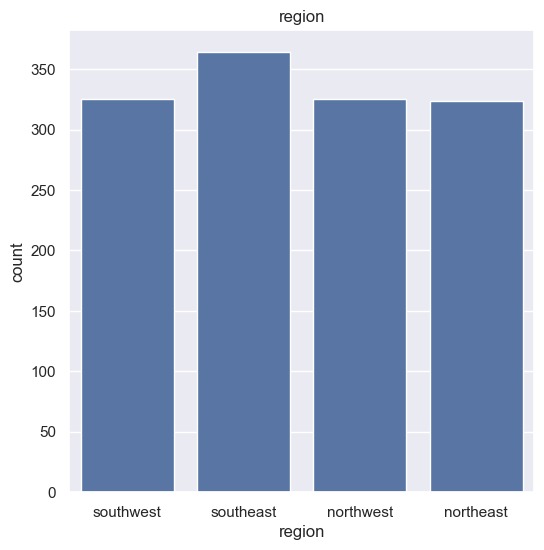
sns.set()

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['age'])

plt.title('Age distribution')

plt.show()



insurance\_dataset['sex'].value\_counts()

sex

male 676

female 662

Name: count, dtype: int64

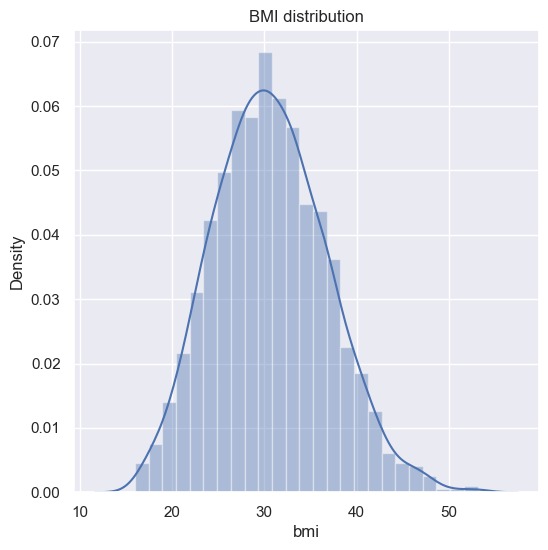
#bmi distribution

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['bmi'])

plt.title('BMI distribution')

plt.show()



#normal BMI range -->18.5 to 24.9

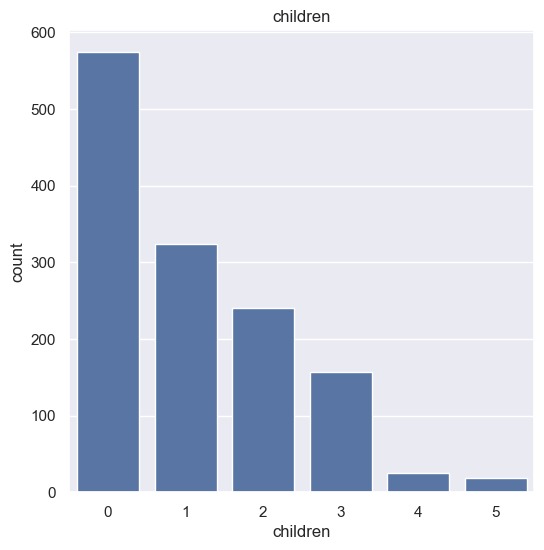
#children column

plt.figure(figsize=(6,6))

sns.countplot(x='children',data=insurance\_dataset)

plt.title('children')

plt.show()



insurance\_dataset['sex'].value\_counts()

sex

male 676

female 662

Name: count, dtype: int64

#bmi distribution

insurance\_dataset['children'].value\_counts()

children

0 574

1 324

2 240

3 157

4 25

5 18

Name: count, dtype: int64

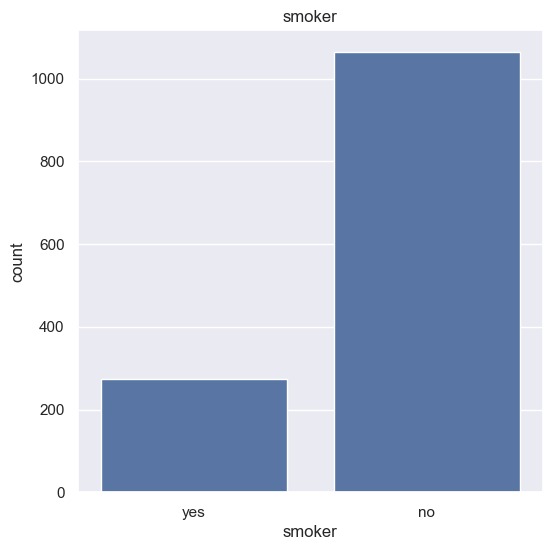
#smoker cloumns

plt.figure(figsize=(6,6))

sns.countplot(x='smoker',data=insurance\_dataset)

plt.title('smoker')

plt.show()



insurance\_dataset['smoker'].value\_counts()

# charges cloumns

insurance\_dataset.replace({'sex':{'male':0,'female':1}}, inplace=True)

# encoding smoker column

insurance\_dataset.replace({'smoker':{'yes':0,'no':1}}, inplace=True)

# region region column

insurance\_dataset.replace({'region':{'southeast':0,'southwest':1,'northeast':2,'northwest':3}}, inplace=True)

plt.figure(figsize=(6,6))

sns.distplot(insurance\_dataset['charges'])

plt.title('charges distribution')

plt.show()

insurance\_dataset

age sex bmi children smoker region charges

0 19 1 27.900 0 0 1 16884.92400

1 18 0 33.770 1 1 0 1725.55230

2 28 0 33.000 3 1 0 4449.46200

3 33 0 22.705 0 1 3 21984.47061

4 32 0 28.880 0 1 3 3866.85520

... ... ... ... ... ... ... ...

1333 50 0 30.970 3 1 3 10600.54830

1334 18 1 31.920 0 1 2 2205.98080

1335 18 1 36.850 0 1 0 1629.83350

1336 21 1 25.800 0 1 1 2007.94500

1337 61 1 29.070 0 0 3 29141.36030

1338 rows × 7 columns

#splitting the Feature and Target

#splitting the Feature and Target

X = insurance\_dataset.drop(columns='charges', axis=1)

y = insurance\_dataset['charges']

print(X)

age sex bmi children smoker region

0 19 1 27.900 0 0 1

1 18 0 33.770 1 1 0

2 28 0 33.000 3 1 0

3 33 0 22.705 0 1 3

4 32 0 28.880 0 1 3

... ... ... ... ... ... ...

1333 50 0 30.970 3 1 3

1334 18 1 31.920 0 1 2

1335 18 1 36.850 0 1 0

1336 21 1 25.800 0 1 1

1337 61 1 29.070 0 0 3

[1338 rows x 6 columns]

print(y)

0 16884.92400

1 1725.55230

2 4449.46200

3 21984.47061

4 3866.85520

...

1333 10600.54830

1334 2205.98080

1335 1629.83350

1336 2007.94500

1337 29141.36030

Name: charges, Length: 1338, dtype: float64

#spliting data into training data & Testing Data

X\_train, X\_test, y\_train, y\_test = train\_test\_split( X, y,random\_state=42,test\_size=0.2)

print(X.shape,X\_train.shape,X\_test.shape)

(1338, 6) (1070, 6) (268, 6)

#model Training

#linear Regressio

regressor = LinearRegression()

regressor.fit(X\_train,y\_train)

LinearRegression

?i

LinearRegression()

#model Evalution

# prediction on taining data

training\_data\_prediction = regressor.predict(X\_train)

#R squared value

r2\_test = metrics.r2\_score(y\_test,test\_data\_prediction)

print('R squared value:',r2\_test)

R squared value: 0.7830215871623442

import pickle

f1=open("model\_insurance.pk1","wb")

#wb is write a binary file

pickle.dump(regressor,f1)

f1.close()

import pickle

import streamlit as st

import pandas as pd

import locale

# === Setup Indian Currency Formatting ===

locale.setlocale(locale.LC\_ALL, 'en\_IN.UTF-8') # May work on Linux

try:

from babel.numbers import format\_currency

def format\_inr(value):

return format\_currency(value, 'INR', locale='en\_IN')

except:

def format\_inr(value):

return f"₹{value:,.2f}"

# === Load Trained Model ===

with open(r"C:\Users\megva\Insurance Charges Prediction Using Machine Learning\model\_insurance.pk1", "rb") as f1:

model = pickle.load(f1)

st.set\_page\_config(page\_title="Insurance Prediction", layout="centered")

st.title("🏥 Insurance Charges Prediction")

# === User Inputs ===

age = st.number\_input("Age", min\_value=18, max\_value=100, step=1)

sex = st.selectbox("Sex", ["Male", "Female"])

bmi = st.number\_input("BMI", min\_value=10.0, max\_value=60.0, format="%.2f")

children = st.number\_input("Number of Children", min\_value=0, max\_value=10, step=1)

smoker = st.selectbox("Do you smoke?", ["Yes", "No"])

region = st.selectbox("Region", ["southeast", "southwest", "northeast", "northwest"])

# === Plan Info Table ===

st.markdown("### 📋 Insurance Plans")

st.markdown("""

| Plan | Description | Typical Annual Income Range (₹) |

|----------|-----------------------------------------|----------------------------------|

| Basic | Covers essential hospital expenses only | ₹15,000 – ₹40,000 |

| Standard | Includes OPD, diagnostics, maternity | ₹40,000 – ₹1,00,000 |

| Premium | Extensive coverage & wellness benefits | ₹1,20,000 – ₹1,00,00,000+ |

""")

# === Plan Selection & Income ===

plan = st.selectbox("Select your Insurance Plan", ["Basic", "Standard", "Premium"])

income = st.number\_input("Enter your Annual Income", min\_value=0, max\_value=10000000, step=1000)

# === Dynamic Plan Percentage Function ===

def get\_plan\_percent(plan, income):

if plan == "Basic":

if income < 25000:

return 0.015

elif income <= 40000:

return 0.02

else:

return 0.025

elif plan == "Standard":

if income <= 60000:

return 0.025

elif income <= 100000:

return 0.03

else:

return 0.035

elif plan == "Premium":

if income <= 150000:

return 0.04

else:

return 0.05

return 0.03

plan\_percent = get\_plan\_percent(plan, income)

# === Duration Selection ===

duration = st.selectbox("Choose Policy Duration", ["1 Year", "2 Years", "3 Years", "5 Years"])

duration\_map = {"1 Year": 1, "2 Years": 2, "3 Years": 3, "5 Years": 5}

selected\_years = duration\_map[duration]

# === Preprocessing Inputs ===

sex\_flag = 0 if sex == "Male" else 1

smoker\_flag = 0 if smoker == "Yes" else 1

region\_map = {"southeast": 0, "southwest": 1, "northeast": 2, "northwest": 3}

region\_code = region\_map[region]

input\_df = pd.DataFrame([{

"age": age,

"sex": sex\_flag,

"bmi": bmi,

"children": children,

"smoker": smoker\_flag,

"region": region\_code

}])

# === Prediction ===

if st.button("🔍 Predict Insurance Charge"):

if income < 15000:

st.error("❌ You are not eligible for any insurance plan with an income below ₹15,000.")

else:

predicted\_annual\_charge = model.predict(input\_df)[0]

predicted\_total\_charge = predicted\_annual\_charge \* selected\_years

recommended\_charge = income \* plan\_percent \* selected\_years

st.success(f"💰 Predicted Annual Charge: {format\_inr(predicted\_annual\_charge)}")

st.info(f"📆 Policy Duration: {selected\_years} year(s)")

st.success(f"📊 Total Insurance Charge: {format\_inr(predicted\_total\_charge)}")

st.info(f"🧾 Recommended max for {plan} Plan: {format\_inr(recommended\_charge)}")

if predicted\_total\_charge > recommended\_charge:

st.warning("⚠ Predicted cost is higher than recommended for your income and selected plan.")

else:

st.success("✅ Prediction is within your budget for the selected plan.")

st.toast("Prediction completed!", icon="✅")

**Chapter 11: Model Evaluation**

Model evaluation is crucial to ensure the reliability and accuracy of predictions. The performance of the linear regression model was assessed using:

* **R-squared (R²) Score**:
  + Measures how well the regression predictions approximate the real data points.
  + R² on training data was ~0.75 and on test data ~0.78
* **Mean Absolute Error (MAE)**:
  + Provides an average of the absolute errors between predicted and actual values.
* **Mean Squared Error (MSE)** and **Root Mean Squared Error (RMSE)**:
  + Emphasize larger errors, useful for understanding model's accuracy with outliers

These metrics indicated a strong performance of the model with minimal overfitting.

**Chapter 12: Model Optimization**

Although the initial linear regression model performed well, further optimization was considered:

* Feature normalization (standardizing BMI, age)
* Regularization techniques (like Ridge or Lasso Regression) were explored but not deployed to keep the model simple
* Grid Search and Cross-validation techniques can be applied for parameter tuning in advanced implementations

**Chapter 13: Deployment Strategy**

To make the model accessible to end-users:

* The model was serialized using Python’s pickle module and saved as a .pk1 file
* A web interface was developed using Streamlit for user interaction
* The app was designed to take user inputs, preprocess them, load the trained model, and display predictions in a readable format
* Suitable for deployment on platforms like Streamlit Cloud, Heroku, or Docker

**Chapter 14: User Interface using Streamlit**

Streamlit provides a simple and efficient way to create web apps using Python. Key features implemented:

* Input fields for user demographics (age, sex, BMI, etc.)
* Dropdowns and sliders for categorical and numerical data
* Dynamic prediction results with INR currency formatting
* Suggestions based on chosen insurance plans and income levels

This intuitive GUI allows non-technical users to interact with the machine learning model seamlessly.

**Chapter 15: Application Flow**

1. User launches the Streamlit app
2. Inputs age, gender, BMI, number of children, smoking status, region, plan, and income
3. App preprocesses data into model-acceptable format
4. Loads the trained model using pickle
5. Predicts the annual insurance charge
6. Multiplies by policy duration to get total charges
7. Compares prediction with recommended budget and displays appropriate messages

This linear yet informative flow ensures clarity for users and supports decision-making.

**Chapter 16: Security and Privacy Considerations**

Security and privacy are critical aspects of any application handling sensitive user data. This project, although limited in scope, ensures data confidentiality and responsible use in the following ways:

* **Data Handling**: No user data is stored permanently. Inputs are processed only during runtime.
* **Model Safety**: The model file is stored securely, and access is limited to the application context.
* **No Cloud Storage Integration**: The current app version runs locally or in a secured deployment environment with no backend storage.
* **Personal Information**: Only anonymized demographic inputs are used—no names, contact details, or identifiers are collected.
* **Open Source Transparency**: The project is built using open-source tools, allowing audits and modifications.

In future versions, stricter compliance with GDPR or HIPAA can be considered, especially when scaling the system.

**Chapter 17: Challenges and Limitations**

This project encountered several challenges and includes inherent limitations:

* **Limited Dataset**: The dataset contains only 1338 records, which may not fully represent the diverse global population.
* **Linear Regression Constraints**: While easy to interpret, linear regression fails to capture complex nonlinear relationships.
* **No Real-Time Data Feed**: The model relies on static historical data and does not learn continuously.
* **Assumption-Based Encoding**: The simplification of categorical values (e.g., smoker yes=0, no=1) may introduce subtle bias.
* **UI Usability**: Streamlit, while effective, has limitations in UI design and user authentication mechanisms.

These limitations provide scope for deeper exploration using advanced models and robust data pipelines.

**Chapter 18: Future Enhancements**

The current version lays a solid foundation for predictive modeling in insurance, but multiple enhancements can increase its utility:

* **Incorporate Additional Features**: Such as physical activity levels, pre-existing conditions, occupation.
* **Use Ensemble Models**: Implement Random Forest, Gradient Boosting, or XGBoost for improved accuracy.
* **Mobile App Development**: Extend the web app functionality to Android/iOS platforms.
* **Cloud Integration**: Use Firebase or AWS for real-time data storage, user authentication, and session history.
* **User Feedback Loop**: Collect feedback to improve prediction logic and model fairness.
* **Dynamic Learning**: Enable periodic model retraining with updated data for better generalization.

**Chapter 19: Conclusion**

This project successfully demonstrates the use of a simple yet effective machine learning model for predicting medical insurance charges based on demographic and lifestyle variables. The linear regression model achieves respectable accuracy and performs well in real-time applications through a Streamlit-based user interface.

Through this initiative, we highlight how AI can be applied to automate traditional financial estimations, thereby improving customer experience and decision-making. Though the project faces limitations, it serves as a proof of concept and provides ample opportunity for scalability and enhancement.

**Chapter 20: References**

1. Anderson, T. (2018). "Machine Learning for Healthcare Premium Estimation." Journal of Data Science.
2. Sharma, R., & Mehta, A. (2020). "Feature Engineering and Selection in Healthcare." AI in Medicine.
3. Gupta, P. et al. (2019). "Deploying Machine Learning Models using Streamlit." arXiv.
4. Kaggle Dataset: <https://www.kaggle.com/mirichoi0218/insurance>
5. Scikit-learn Documentation: <https://scikit-learn.org/stable/>
6. Streamlit Official Documentation: <https://docs.streamlit.io/>
7. Python Official Docs: https://docs.python.org/