# 💰 Capstone Project – AI Medical Insurance Cost Prediction Tool

🎯 *Leveraging Machine Learning to Improve Cost Transparency in Healthcare*

**Dr Subramani Suresh**

## ✅ Project Overview

Healthcare insurance premiums can vary significantly depending on lifestyle, demographics, and health-related risk factors. To address this challenge, I developed an **AI-powered web application** capable of predicting the expected cost of medical insurance using real-world, data-driven insights.

This project demonstrates end-to-end machine learning integration — from data preprocessing and model optimization to a fully deployed, interactive web interface.

🌐 **Live Demo:**
https://ckd-prediction-6dif.onrender.com/insurance/

**Github location:**

- Model: https://github.com/aisubramani/Hope_Ai_Assignment/blob/main/Week5_7_Web_Project/Medical_insurance/Medical_Insurance_Cost_Prediction.ipynb
- Django app: https://github.com/aisubramani/Hope_Ai_Assignment/tree/main/Week5_7_Web_Project/Medical_insurance/insurance

## 📊 Dataset Summary

The model is built on a dataset containing **1338 records** and **6 key features**:

- **Age** – Age in years
- **Sex** – Male / Female
- **BMI** – Body fat index
- **Children** – Number of dependents
- **Smoker** – Lifestyle habit risk
- **Charges** – Final insurance cost (USD)

Exploratory data analysis highlighted three strong contributors to increased cost:
✅ Age

✅ BMI
✅ Smoking Status (highest impact)

# 🧪 Supervised Regression Model Comparison

To find the most accurate prediction engine, multiple machine learning algorithms were trained and evaluated:

| Regression Model | $R^2$ Score |
|---|---|
| Linear Regression | 0.78 |
| Ridge Regression | 0.78 |
| Lasso Regression | 0.78 |
| **Random Forest (Tuned)** | **0.87** ✅ |
| AdaBoost (Tuned) | 0.85 |

## ✅ Why Random Forest Wins

- Handles complex, non-linear relationships
- Resistant to outliers
- Superior generalization accuracy

After hyperparameter tuning with GridSearchCV, Random Forest achieved the best performance and became the final production model.

# 🧠 Technical Stack

**Programming & ML**

- Python
- Scikit-learn
- Pandas, NumPy

**Web Development**

- Django
- HTML5 / CSS3

**Deployment**

- Render Cloud Platform
- Gunicorn
- WhiteNoise for static file delivery

This ensures scalability, security, and a clean user experience.

## 🧩 System Architecture

1. **Data Preparation**
   - Encoding categorical values
   - Scaling numerical features
2. **Model Training**
   - Train-Test split
   - MAE, RMSE, $R^2$ evaluation
3. **Model Optimization**
   - GridSearchCV tuning
4. **Serialization**
   - Pickle saved model
5. **Web Integration**
   - User inputs → real-time predictions

## 🎨 User-Friendly Web Interface