# Smart Premium: Predicting Insurance Costs with Machine Learning

# 1. Project Title:

* **Smart Premium: Predicting Insurance Costs with Machine Learning**

# 2. Domain:

* Insurance | Finance | Machine Learning | Data Science

# 3. Objective:

* To build a machine learning model that accurately predicts insurance premium amounts based on customer characteristics and policy information.

# 4. Business Use Cases:

* Insurance Companies: Automate premium calculation and reduce manual effort.
* Financial Institutions: Assess customer risk based on predicted premiums.
* Healthcare Providers: Estimate costs based on health metrics and behaviors.
* Customers: Get real-time, personalized premium quotes.

# 5. Technologies Used:

* Python, NumPy, Pandas, Scikit-Learn, XGBoost.
* Stream-lit for deploying a web-based app.
* Jupyter Notebook for development.
* Git/GitHub for version control.

# 6. Dataset Overview:

* Total Rows: ~1200000
* Target Variable: Premium Amount
* Features: Age, Income, Gender, Education, Health Score, Vehicle Age, Credit Score, Previous Claims, etc.
* Type: Mix of categorical, numerical, and textual data
* Issues Handled: Missing values, skewed distributions, encoding categorical variables

# 7. Approach:

**📌 Step 1: Data Understanding**  
- Performed EDA to find correlations, missing values, skewness  
- Visualized distributions and relationships  
  
**📌 Step 2: Data Preprocessing**  
- Handled missing values using median/mode  
- One-hot and ordinal encoding used for categorical features  
- Log transformation applied to skewed target (Premium Amount)  
- Data split into 80% training, 20% testing  
  
**📌 Step 3: Model Building**  
- Tried:  
 - Linear Regression  
 - Decision Tree  
 - Random Forest  
 - XGBoost  
  
**📌 Step 4: Hyperparameter Tuning**  
- Used RandomizedSearchCV  
- Tuned parameters like max\_depth, learning\_rate, gamma, etc.  
- Best score (XGBoost): R² ≈ 0.07  
  
**📌 Step 5: Model Deployment**  
- Deployed using Streamlit  
- Built a clean UI with dropdowns and number inputs  
- Predicted and inverse-transformed (exp) the log of the premium

# 8. Final Stream-lit App Features:

✅ Real-time prediction  
 ✅ Manual input for all user features  
 ✅ Dynamic insurance premium shown on click  
 ✅ Supports log-transformed model output

# 9. Evaluation Metrics:

|  |  |
| --- | --- |
| Metric | Value (XGBoost after tuning) |
| R² Score | 0.0709 |
| MAE (Mean Absolute Error) | ~0.75 |
| RMSE (Root Mean Squared Error) | ~1.05 |
| RMSLE (Logarithmic Error) | ~0.16 |

# 10. Deliverables:

✅ PROJECT\_3\_ML.ipynb for ML Models (Jupyter Notebook)

✅ PROJECT\_3.ipynb for Data Preprocessing (Jupyter Notebook)

✅ Trained model: xgb\_smartpremium\_model.pkl

✅ Deployment script: my\_script.py

✅ Documentation (this file)

✅ Stream-lit app demo

# 11. Learnings:

* Practical understanding of real-world data preprocessing
* Choosing appropriate encoders and transformations
* Building pipelines and tuning hyperparameters
* Streamlit deployment and packaging a full ML product

# 12. Stream-lit App Screenshot:



