


# Insurance Premium Prediction

By Anand Chavan

# Content

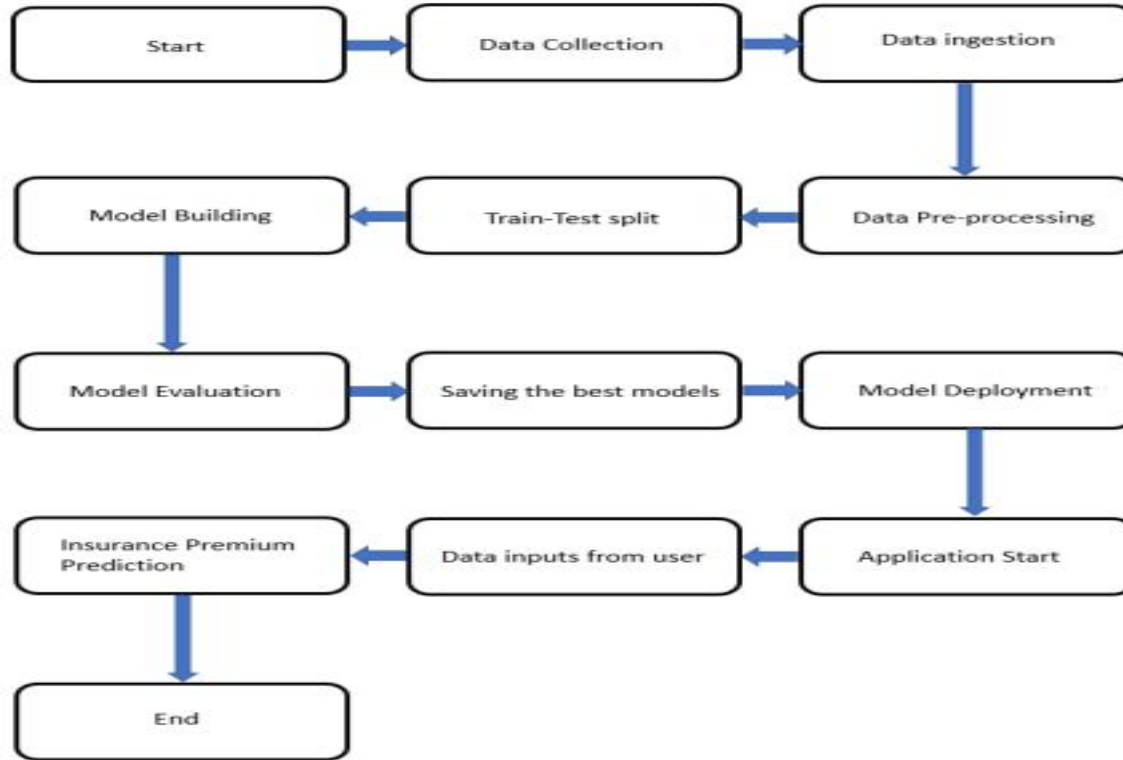
- Objectives
  - Architecture
  - Data set
  - EDA
  - Data Pre-Processing
  - Model Building & evaluation
  - Model Deployment
  - Key Performance Indicators(KPI)
  - Q & A
- 

# Objective

To give people an estimate of how much they need based on their individual health situation. After that, customers can work with any health insurance carrier and its plans and perks while keeping the projected cost from our study in mind. I am considering variables as age, sex, BMI, number of children, smoking habits and living region to predict the premium. This can assist a person in concentrating on the health side of an insurance policy rather than the ineffective part.



# Architecture



# Dataset

For training and testing the model, I used the public data set available in Kaggle, “Insurance Premium Prediction” by nursnaaz

URL: <https://www.kaggle.com/noordeen/insurance-premium-prediction>

	age	sex	bmi	children	smoker	region	expenses
0	19	female	27.9	0	yes	southwest	16884.92
1	18	male	33.8	1	no	southeast	1725.55
2	28	male	33.0	3	no	southeast	4449.46
3	33	male	22.7	0	no	northwest	21984.47
4	32	male	28.9	0	no	northwest	3866.86

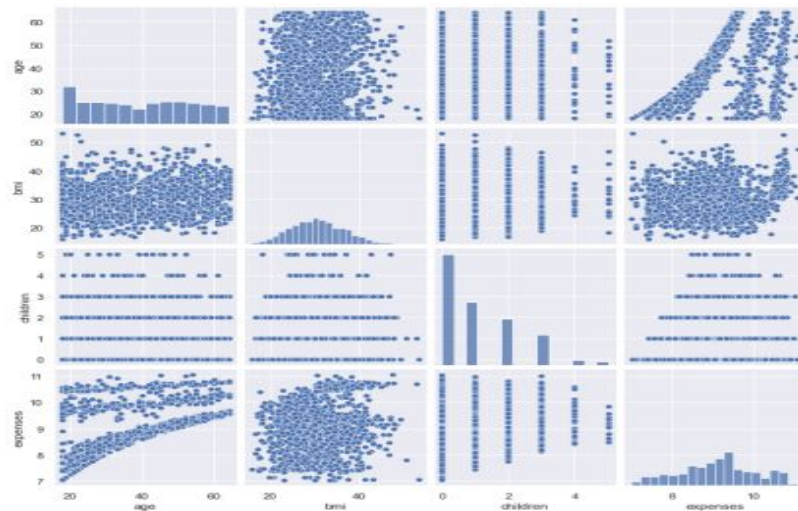
# EDA- Data Cleaning

- The given data set has 7 features and each one is quantitative in nature.
- This data set doesn't have any missing values.
- The shape of the data set is 1338 rows  $\times$  7 columns.



# EDA- Data Visualization

- Age values are constrained between 18-64 years.
- BMI values varies from 16 to 53.1.
- Age & BMI have spread data points.
- With age expenses are also increasing following linear positive relationship.
- With number of children expenses also follow linear positive relationship.



# Data Pre-Processing

- Split the data-frame into the training (75% of the data) and testing (25% of the data) data-frames respectively.
- For building linear regression models, used the scaled data obtained by scaling the features using the Standard scaler. Tree models use the original data i.e., without scaling as they aren't affected by the feature scaling.
- Then both the training and testing data-frames were further split into `X_train`, `X_test`, `y_train` and `y_test`. Here the data-frames with `X` indicate independent features while those with `y`. indicate the dependent or the target feature.





# Model Building & Evaluation

- Experimented with linear, ridge & lasso regression models & Tree models such as Decision Tree, Random Forest and Boosting model as Gradient Boosting.
- Evaluation as per R2-score, Adj. R2-score & RMSE value

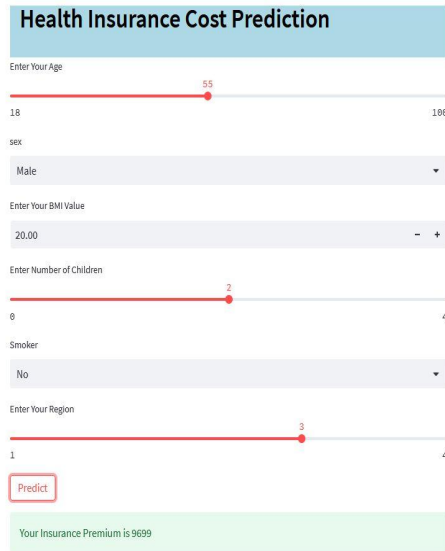
$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}$$

$$R^2 = 1 - \frac{SS_{\text{Regression}}}{SS_{\text{Total}}}$$

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

# Model Deployment

Saved Linear regression model and the Linear regression Model into the “Lr\_models” directory. Then deployed the Linear regression model using the Streamlit and linked to web application Deployed on web using GitHub and Render.



**Health Insurance Cost Prediction**

Enter Your Age: 55 (Range: 18 to 100)

SEX: Male (Dropdown)

Enter Your BMI Value: 20.00 (Range: - to +)

Enter Number of Children: 2 (Range: 0 to 4)

Smoker: No (Dropdown)

Enter Your Region: 3 (Range: 1 to 4)

**Predict**

Your Insurance Premium is 9699

# Key Performance Indicators(KPI)

- Time and workload reduction using the regression models.
- Comparison of the  $R^2$  scores and the Adjusted  $R^2$  scores of the model on both the training and the testing data.
- Comparison of the RMSE scores of the model on both the training and the testing data.



# Q&A

- Why feature scaling is not necessary for the tree-based models ?

Answer: The tree-based models are not sensitive to the scale of the features. If we consider a decision tree algorithm, it splits a node based on a single feature and this is not influenced by the other features, i.e., there won't be any effect of the remaining features if a split is performed based on one single feature.

- What are the different stages of deployment?

Answer: First, deployed the model locally using Flask (a micro web framework) which works as a backend application. The frontend application is a web page designed using HTML5 with CSS styling. So, when a user enters the data and hit "predict" button, model in the backend flask application makes prediction and it will be displayed in the frontend application which user can take a note. Then, deployed this application on web using Heroku and Gunicorn (a python web server gateway interface HTTP server).

# Q&A

- How are logs managed?

Answer: The entire project is divided into two stages - the development stage and the deployment stage. The logs recorded during the development stage are stored in the "development\_logs.log" file while the logs recorded during the local deployment are stored in the "deployment\_logs.log" file.

- Explain importance of virtual environment?

Answer: Virtual environment gives us an independent platform to decouple and isolate versions of Python and its associated pip packages. This allows end-users to install and manage their own set of packages that are independent of those provided by the system.

