# Hope Artificial Intelligence

Assignment-Regression Algorithm

Download Set:

Google Drive Link Click here .

Git Hub Link: https://raw.githubusercontent.com/RamishaRaniK/dataset/main/insurance_pre.csv

## Problem Statement or Requirement:

A client's requirement is, he wants to predict the insurance charges based on the several parameters. The Client has provided the dataset of the same.

As a data scientist, you must develop a model which will predict the insurance charges.

1.) Identify your problem statement
The problem statement in this scenario is to develop an AI model that can predict insurance charges for individuals based on several Inputs contained in the provided dataset.

2.) Tell basic info about the dataset (Total number of rows, columns)
Number of rows 1137
number of columns 6

3.) Mention the pre-processing method if you're doing any (like converting string to number – nominal data)
dataset=pd.get_dummies(dataset,drop_first=True)-
pd.get_dummies method is used to convert the string values into number values.

4.) Develop a good model with r2_score. You can use any machine learning algorithm; you can create many models. Finally, you have to come up with final model.

5.) All the research values (r2_score of the models) should be documented. (You can make tabulation or screenshot of the results.)

6.) Mention your final model, justify why u have chosen the same.

Random Forest

| criterion | n_estimators | Values |
|---|---|---|
| *poisson* | 1000 | 0.854861868 |

R2_Score is better than other models.

Kindly create Repository in the name Regression Assignment.

Upload all the ipynb and final document in the pdf Communication is important (How you are representing the document.)

## Multiple linear Regression

Value 0.789479035

## Support Vector Machine

| | kernel | gamma | Values |
|---|---|---|---|
| | *linear* | *scale* | -0.010102665 |
| | *poly* | *scale* | -0.075699656 |
| Defalult | *rbf* | *scale* | -0.083382386 |
| | *sigmoid* | *scale* | -0.075429243 |
| | *precomputed* | *scale* | ValueError: Precomputed matrix must be a square matrix. Input is a 936x5 matrix. |
| | | | |
| | *linear* | *auto* | -0.010102665 |
| | *poly* | *auto* | -0.075699656 |
| | *rbf* | *auto* | -0.083382386 |
| | *sigmoid* | *auto* | -0.075429243 |
| | *precomputed* | *auto* | ValueError: Precomputed matrix must be a square matrix. Input is a 936x5 matrix. |
| | | | |

## Decision Tree

| criterion | splitter | Values |
|---|---|---|
| *squared_error* | *random* | 0.75412567 |
| *friedman_mse* | *random* | 0.664306406 |
| *absolute_error* | *random* | 0.74092581 |
| *poisson* | *random* | 0.736888486 |
| | | |
| *squared_error* | *best* | 0.681842956 |
| *friedman_mse* | *best* | 0.688845637 |
| *absolute_error* | *best* | 0.698828727 |
| *poisson* | *best* | 0.715818957 |
| | | |

# Random Forest

| criterion | n_estimators | Values |
|---|---|---|
| *squared_error* | 50 | 0.849832932 |
| *friedman_mse* | 50 | 0.850071614 |
| *absolute_error* | 50 | 0.852665599 |
| *poisson* | 50 | 0.849107596 |
| | | |
| *squared_error* | 100 | 0.853830791 |
| *friedman_mse* | 100 | 0.854051894 |
| *absolute_error* | 100 | 0.852009362 |
| *poisson* | 100 | 0.852633426 |
| | | |
| *squared_error* | 1000 | 0.854177812 |
| *friedman_mse* | 1000 | 0.853737812 |
| *absolute_error* | 1000 | 0.853767376 |
| *poisson* | 1000 | 0.854861868 |
| | | |