

# *Hope Artificial Intelligence*

## 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

- ✓ Insurance charges (continuous numerical value).
- ✓ Factors influencing insurance charges (e.g., age, gender, BMI, number of children, smoking status).
- ✓ Regression problem under the umbrella of supervised learning, as the model will learn from a labeled dataset where the input features correspond to known insurance charges.
- ✓ A predictive model that can generalize well to new data, thereby accurately estimating insurance charges for prospective clients.

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

**\*Total Number of Rows:** This indicates how many individual records are present in the dataset.

**\*Total Number of Columns:** This shows the number of features or variables used to describe each record.

3. mention the pre-processing method if you're doing any (like converthing string to number- nominal data)

**\*Handling Missing Values:** Check for any missing data and decide on a strategy (e.g., removing rows, filling them with a mean/median, or using a more sophisticated method).

**\*Encoding Categorical Variables:** If the dataset contains nominal data (like gender, region, etc.), we need to convert these categorical variables into numerical format. Common techniques include:

**\*Label Encoding:** Assigning each category a unique integer.

**\*One-Hot Encoding:** Creating binary columns for each category.

**\*Scaling Numerical Features:**

**\*Standardization (Z-score Normalization):** This method rescales the data to have a mean of 0 and a standard deviation of 1, which is particularly useful for algorithms that assume normally distributed data.

**\*Min-Max Scaling:** This method scales the features to a fixed range, usually  $[0, 1]$ , which can be beneficial for algorithms that require bounded input.

❖ find the 3- stage problem statement.

### **Stage 1: Machine Learning**

In this initial stage, we recognize that the overall framework for our task is machine learning. This involves using data to train algorithms that can learn patterns and relationships within the data. The aim is to create a model that can make predictions about insurance charges based on input parameters.

### **Stage 2: Supervised Learning**

In this stage, we identify the specific approach we will take within machine learning. Supervised learning is suitable for our problem because we have a dataset with labeled examples, meaning we know the input features (e.g., age, BMI, smoking status) and the corresponding target variable (insurance charges). The model will learn from these labeled examples to make predictions on new, unseen data

### Stage 3: Regression Algorithm

we focus on the type of algorithm we will use to solve the problem. Since the target variable (insurance charges) is continuous, we will employ regression algorithms. Multiple linear regression, svm regression , decision trees, random forests, or more complex models like gradient boosting. The choice of the regression algorithm will depend on the characteristics of the data and the performance metrics we aim to achieve.

**To find following the machine learning regression method using in r\_value**

1. **MULTIPLE LINEAR REGRESSION** (R\_value)=0.78947
2. **SUPPORT VECTOR MACHINE:**

SI.No	HYPER PARAMETER	Linear	rbf	Poly	Sigmoid
1	C=1.0	-0.01010	-0.08338	-0.07569	-0.07542
2	C=10.0	0.46246	-0.03227	0.03871	0.03930
3	C=100.0	0.62887	0.32003	0.06179	0.52761
4	C=500.0	0.76105	0.66429	0.82636	0.44460
5	C=1000.0	0.76310	0.81020	0.85664	0.28747
6	C=2000.0	0.74404	0.85477	0.86055	-0.59395
7	C=3000.0	0.74142	0.86633	0.85989	-2.12441

The **SVM Regression** using hyper tuning parameter with **C = 3000.0** in **rbf** has 0.86633 **Highest Accuracy**.

### 3. DECISION TREE:

SI.NO	CRITERION	SPLITTER	MAX_FEAURES	R VALUE
1	<i>friedman_mse</i>	best	sqrt	0.60967
2	<i>friedman_mse</i>	random	sqrt	0.61316
3	<i>friedman_mse</i>	best	log2	0.65189
4	<i>friedman_mse</i>	random	log2	0.68463
5	<i>friedman_mse</i>	best	none	0.69332
6	<i>friedman_mse</i>	random	none	0.70127
7	<i>squared_error</i>	best	sqrt	0.70383
8	<i>squared_error</i>	random	sqrt	0.71077
9	<i>squared_error</i>	best	log2	0.62264
10	<i>squared_error</i>	random	log2	0.68136
11	<i>squared_error</i>	best	none	0.68010
12	<i>squared_error</i>	random	none	0.71077
13	<i>absolute_error</i>	best	sqrt	0.76355
14	<i>absolute_error</i>	random	sqrt	0.74423
15	<i>absolute_error</i>	best	log2	0.73536
16	<i>absolute_error</i>	random	log2	0.64685
17	<i>absolute_error</i>	best	none	0.66792
18	<i>absolute_error</i>	random	none	0.70688
19	<i>Poisson</i>	best	sqrt	0.75398
20	<i>Poisson</i>	random	sqrt	0.66874
21	<i>Poisson</i>	best	log2	0.70685
22	<i>Poisson</i>	random	log2	0.62464
23	<i>Poisson</i>	best	none	0.71623
24	<i>Poisson</i>	random	none	0.74130

The Decision Tree Regression use R value (*absolute\_error*, best, sqrt) = 0.76355

#### 4. RANDOM FOREST:

SI.NO	N_ESTIMATORS	CRITERION	MAX_FEATURES	R_VALUE
1	50	<i>friedman_mse</i>	sqrt	0.86966
2	100	<i>friedman_mse</i>	sqrt	0.87185
3	50	<i>friedman_mse</i>	log2	0.87033
4	100	<i>friedman_mse</i>	log2	0.87191
5	50	<i>friedman_mse</i>	none	0.84971
6	100	<i>friedman_mse</i>	none	0.85505
7	50	<i>squared_error</i>	sqrt	0.86720
8	100	<i>squared_error</i>	sqrt	0.87168
9	50	<i>squared_error</i>	log2	0.87072
10	100	<i>squared_error</i>	log2	0.86891
11	50	<i>squared_error</i>	none	0.85059
12	100	<i>squared_error</i>	none	0.85317
13	50	<i>absolute_error</i>	sqrt	0.86777
14	100	<i>absolute_error</i>	sqrt	0.87425
15	50	<i>absolute_error</i>	log2	0.86926
16	100	<i>absolute_error</i>	log2	0.86773
17	50	<i>absolute_error</i>	none	0.85032
18	100	<i>absolute_error</i>	none	0.85499
19	50	<i>Poisson</i>	sqrt	0.86733
20	100	<i>Poisson</i>	sqrt	0.86797
21	50	<i>Poisson</i>	log2	0.87112
22	100	<i>Poisson</i>	log2	0.86931
23	50	<i>Poisson</i>	none	0.85174
24	100	<i>Poisson</i>	none	0.85770

The Random Forest Regression use R\_value (N\_ESTIMATORS=100, CRITERION= *absolute\_error*, MAX\_FEATURES= sqrt)=0.87425

## **The final machine learning best method of Regression:**

1. Random Forest R\_value(N\_ESTIMATORS=100, CRITERION= *absolute\_error*, MAX\_FEATURES= sqrt) = 0.87425
2. Support vector machine R\_value(rbf and hyper parameter (C=3000.0))=0.86633