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	*friedman_mse*	best	sqrt	0.72430
2	*friedman_mse*	random	sqrt	0.65203
3	*friedman_mse*	best	log2	0.71217
4	*friedman_mse*	random	log2	0.67773
5	*friedman_mse*	best	none	0.70956
6	*friedman_mse*	random	none	0.70130
7	*squared_error*	best	sqrt	0.67651
8	*squared_error*	random	sqrt	0.64483
9	*squared_error*	best	log2	0.60414
10	*squared_error*	random	log2	0.64049
11	*squared_error*	best	none	0.69482
12	*squared_error*	random	none	0.70408
13	*absolute_error*	best	sqrt	0.67773
14	*absolute_error*	random	sqrt	0.72934
15	*absolute_error*	best	log2	0.71816
16	*absolute_error*	random	log2	0.68955
17	*absolute_error*	best	none	0.69449
18	*absolute_error*	random	none	0.73937
19	*Poisson*	best	sqrt	0.70477
20	*Poisson*	random	sqrt	0.69467
21	*Poisson*	best	log2	0.55602
22	*Poisson*	random	log2	0.72802
23	*Poisson*	best	none	0.72601
24	*Poisson*	random	none	0.69319

The Decision Tree Regression use R value (*absolute_error,* random, none) = 0.73937

4. RANDOM FOREST:

SI.NO	N_ESTIMATORS	CRITERION	MAX_FEATURES	R_VALUE
1	50	*friedman_mse*	sqrt	0.86824
2	100	*friedman_mse*	sqrt	0.87065
3	50	*friedman_mse*	log2	0.86557
4	100	*friedman_mse*	log2	0.86930
5	50	*friedman_mse*	none	0.84332
6	100	*friedman_mse*	none	0.85609
7	50	*squared_error*	sqrt	0.86931
8	100	*squared_error*	sqrt	0.86947
9	50	*squared_error*	log2	0.86803
10	100	*squared_error*	log2	0.87411
11	50	*squared_error*	none	0.85497
12	100	*squared_error*	none	0.85713
13	50	*absolute_error*	sqrt	0.87423
14	100	*absolute_error*	sqrt	0.87214
15	50	*absolute_error*	log2	0.87093
16	100	*absolute_error*	log2	0.87358
17	50	*absolute_error*	none	0.85181
18	100	*absolute_error*	none	0.85084
19	50	*Poisson*	sqrt	0.86150
20	100	*Poisson*	sqrt	0.87157
21	50	*Poisson*	log2	0.86948
22	100	*Poisson*	log2	0.87181
23	50	*Poisson*	none	0.84620
24	100	*Poisson*	none	0.85623

The Random Forest Regression use R_value (N_ESTIMATORS=50**,** CRITERION= *absolute_error,* MAX_FEATURES= sqrt)=0.87423

The final machine learning best method of Regression:

1. Random Forest R_value(N_ESTIMATORS=50**,**
   CRITERION= *absolute_error,* MAX_FEATURES= sqrt)= 0.87423
2. Support vector machineR_value(rbf and hyper parameter
   (C=3000.0))=0.86633