Problem Statement or Requirement:

The client's requirement is, they want 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

I had a look at the data set provided for predicting the insurance charges.

	age	sex	bmi	children	smoker	**Charges (to be predicted)**
0	19	female	27.900	0	yes	16884.92400
1	18	male	33.770	1	no	1725.55230
2	28	male	33.000	3	no	4449.46200
3	33	male	22.705	0	no	21984.47061
4	32	male	28.880	0	no	3866.85520

Stage 1: From the given dataset, insurance **charges** to be predicted and it's a numeric output variable. So, we can use **Machine Learning Algorithm.**

Stage 2: Both requirements are clear and input/output are also clear. So we can consider **Supervised Learning**

Stage 3: This has multiple input independent variables and one numeric output variable to predict. So we can consider **multiple linear regression**

Solution using: ML/Supervised/MLR case

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

1338 rows × 6 columns available in the dataset.

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

The input variables age, sex, bmi, no of children and smoker field. From this we need to apply OHE and convert them to nominal and remove first. We need to convert the categorical value columns Sex and Smoker using One Hot coding method to nominal data

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.

I have created 3 models using MLR, DT, RF algorithms and applied many parameters to pick the most accurate model. I found DecisionTree Algorithm was the most time consuming one.

5.) All the research values (r2_score of the models) should be documented.

Model: Linear Regression r2_score applied with no additional parameters

Linear Regression	
r2 score	**0.789479035**

Model: Decision Tree r2_scores applied 3 parameters criterion, splitter, max_features

Decision Tree

criterion	Splitter	max_features	r2 score
squared_error	best	sqrt	0.658970474
squared_error	best	log2	0.785286016
squared_error	random	sqrt	0.653727921
squared_error	random	log2	0.641818923
friedman_mse	best	sqrt	0.715869316
friedman_mse	best	log2	0.631754898
friedman_mse	random	sqrt	0.687178701
friedman_mse	random	log2	0.669021885
absolute_error	best	sqrt	0.683955987
absolute_error	best	log2	0.69094248
absolute_error	random	sqrt	0.722961438
absolute_error	random	log2	0.745356484
poisson	best	sqrt	0.703553506
poisson	best	log2	0.701903708
poisson	random	sqrt	0.752398459
poisson	random	log2	0.701229918

Random Forest r2_Scores applied three parameters n_estimators, criterion, max_features.

RF_Regression

n_estimators	criterion	max_features	r2 score
10	squared_error	sqrt	0.851342099
10	squared_error	log2	0.849142385
10	squared_error	1.0	0.834726819
10	friedman_mse	sqrt	0.857236163
10	friedman_mse	log2	0.848612545
10	friedman_mse	1.0	0.846871859
10	absolute_error	sqrt	0.859449705
10	absolute_error	log2	0.85637219
10	absolute_error	1.0	0.851818636

10	poisson	sqrt	0.861283985
10	poisson	log2	0.852992776
10	poisson	1.0	0.83861262
100	squared_error	sqrt	0.870026064
100	squared_error	log2	0.870964239
100	squared_error	1.0	0.855969288
100	friedman_mse	sqrt	0.872118901
100	friedman_mse	log2	0.871561249
100	friedman_mse	1.0	0.855572882
100	absolute_error	sqrt	0.872749862
100	absolute_error	log2	0.874520156
100	absolute_error	1.0	0.853636902
100	poisson	sqrt	0.871212855
100	poisson	log2	0.871055144
100	poisson	1.0	0.851181454
50	squared_error	sqrt	0.86719018
50	squared_error	log2	0.869332062
50	squared_error	1.0	0.869393963
50	friedman_mse	sqrt	0.868223405
50	friedman_mse	log2	0.864626703
50	friedman_mse	1.0	0.867711344
50	absolute_error	sqrt	0.866641536
50	absolute_error	log2	0.86686056
50	absolute_error	1.0	0.868083282
50	poisson	sqrt	0.865257126
50	poisson	log2	0.866667176
50	poisson	1.0	0.87051624

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

Rainforest Regressor is the chosen model

With Decision Tree, I could get r2_score **0.785286016278048 (78.5%)**

With Linear regression, I could get r2_score **0.7894790349867 (78.9%)**

With Random Forest,

I could get the maximum possible r2_score: **0.874520156 (87.45%) accuracy.**

Since its higher than the other models created, I choose Random Forest for deployment with the parameters shown below.

RandomForestRegressor(n_estimators=100, criterion= 'absolute_error', max_features='log2')