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

Step1: Domain selection: Machine learning

Step 2: Learning Selection: Supervised Learning

Step 3: Regression analysis

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

Total number of rows =1338, columns =6 (Before preprocessing 6 columns, after preprocessing columns are 'age', 'bmi', 'children', 'charges', 'sex_male', 'smoker_yes')

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

Using nominal data with one-hot encoding method, sex column is splitted into sex_male and sex_female, and dropping out the first column for dummies, so alphabetically sex_female column is removed. And smoker column as smoker_yes converting into 0's and 1's)

```
[4]: dataset=pd.get_dummies(dataset,dtype=int,drop_first=True) #Removing the first column like dummies
```

```
[5]: dataset.head(5)
```

[5]:

	age	bmi	children	charges	sex_male	smoker_yes
0	19	27.900	0	16884.92400	0	1
1	18	33.770	1	1725.55230	1	0
2	28	33.000	3	4449.46200	1	0
3	33	22.705	0	21984.47061	1	0
4	32	28.880	0	3866.85520	1	0

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.

To develop a good predictive model, multiple machine learning algorithms were trained and evaluated using the R^2 score as the primary performance metric. The following models were compared:

1. Multiple Linear Regression – R^2 = 0.7894
2. Support Vector Machine (SVM) – R^2 values varied between –0.12 and 0.54 depending on the kernel and hyperparameter C; overall performance was not satisfactory.
3. Decision Tree Regressor – R^2 values ranged between 0.63 and 0.72 based on the criterion and splitter used.
4. Random Forest Regressor – The highest R^2 value of 0.8566 was obtained with:
    a. Criterion: friedman_mse
    b. n_estimators: 100

Final Model: Random Forest Regressor

Among all the tested models, the Random Forest Regressor achieved the best performance, with an R^2 score of 0.8566 achieving **85%**.

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

Simple Linear Regression- This algorithm is not used, since we have multiple input and one output.

Multiple Linear Regression- 0.7894790349867009

Support Vector Machine

R score values based on Kernel:

Precomputed kernel will not work since our data is not a square matrix.

Hyper parameter	Linear	RBF	Poly	Sigmoid
Default C=1.0	-0.11166128719608448	-0.08842732776913875	-0.06429258402105531	-0.0899412170256757
C=0.1	-0.12207668380229886	-0.08957624598812952	-0.08625251710262294	-0.08974351910465961
C=10	-0.0016176324886472138	-0.08196910396420853	-0.09311615532848516	-0.09078319814614

C=100	0.5432818196692804	-0.12480367775039669	-0.09976172333666167	-0.11814554828411405

Decision Tree Algorithm

Criterion	Splitter	R score
Squared_error	best	0.6986249709765187
Squared_error	random	0.6334853720103432
friedman_mse	best	0.6802867974416396
friedman_mse	random	0.6987900356205744
absolute_error	best	0.6879158333030672
absolute_error	random	0.727946927047933
poisson	best	0.7182591053832799
poisson	random	0.6892690762093842

Random Forest Regressor

criterion	n_estimators	R_Score
squared_error	100	0.8554517443893913
squared_error	50	0.854553721573263
absolute_error	100	0.8486111117854671
absolute_error	50	0.8497742518994779
friedman_mse	100	0.8566110568870415
friedman_mse	50	0.8499160799641188
poisson	100	0.8515852617164237
poisson	50	0.8526562471572516

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

After comparing the performance of various regression models including Multiple Linear Regression, Support Vector Machine (SVM) with different kernels and hyperparameters, Decision Tree, and Random Forest Regressor, the Random Forest Regressor achieved the highest R^2 score.

Although not every model performed well, the Random Forest Regressor with the friedman_mse criterion provided the best performance among all the tested models. Since it has an accuracy of only 85%, the model is not saved and deployed.