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.
    a. We need to predict the insurance charges (i.e., Numeric value) based on the several parameters. So, need to predict a **Regression** value.
    b. Input and output labels are available. So, **it's a Supervised learning**.
    c. Domain: **Machine Learning**.
2. Tell basic info about the dataset (Total number of rows, columns)
    a. Total no. of rows: 1338
    b. Total no. of columns: 6
        i. Inputs: age, **sex**, bmi, children, **smoker**
        ii. Output: charges
    c. Also, I feel there is no need for standardisation since inputs values are closer.
3. Mention the pre-processing method if you're doing any (like converting string to number – nominal data)
    a. Sex, Smoker inputs are **categorical data** of type **nominal**. Hence, we need to apply **One Hot Encoding** to the Sex and Smoker columns.
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.
    a. Algorithms used to create models and R2Score listed below
        i. Multiple Linear Regression [***Max R2Score: 0.7894790349867***]
        ii. Support Vector Machine [***Max R2Score: 0.595089***]
        iii. Decision Tree [***Max R2Score: 0.708199***]
        iv. Random Forest [***Max R2Score: 0.870614***]
5. All the research values (r2_score of the models) should be documented. (You can make tabulation or screenshot of the results.)

Algorithm	Hyper Tuning parameter's	R2Score
Multiple Linear Regression		**0.789479035**

Algorithm	Hyper Tuning parameter's		R2Score
Support Vector Machine	**Kernel**	**C**	
	linear	0.1	-0.120664
	linear	1	-0.100139
	linear	10	0.113603
	linear	**100**	**0.595089**
	poly	0.1	-0.087137
	poly	1	-0.072459
	poly	10	-0.084252
	poly	100	-0.097206
	rbf	0.1	-0.089586
	rbf	1	-0.088513
	rbf	10	-0.082344
	rbf	100	-0.123247
	sigmoid	0.1	-0.089743
	sigmoid	1	-0.089925
	sigmoid	10	-0.090634
	sigmoid	100	-0.113871

Algorithm	Hyper Tuning parameter's		R2Score
Decision Tree	**Criterion**	**Splitter**	
	squared_error	best	0.685666
	squared_error	random	0.693411
	friedman_mse	best	0.696137
	friedman_mse	**random**	**0.708199**
	absolute_error	best	0.704703
	absolute_error	random	0.699584

Algorithm	Hyper Tuning parameter's		R2Score
Random Forest	**Criterion**	**max_feature**	**R2Score**
	squared_error	**sqrt**	**0.870614**
	squared_error	log2	0.866356
	squared_error	None	0.857086
	friedman_mse	sqrt	0.86782
	friedman_mse	log2	0.869767
	friedman_mse	None	0.853416
	absolute_error	sqrt	0.869547
	absolute_error	log2	0.868601
	absolute_error	None	0.853761
	poisson	sqrt	0.866979
	poisson	log2	0.868611
	poisson	None	0.854629

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

    Hence **Random Forest** has the highest R2 Score. So, **Random Forest** is the Final Model.

Algorithm	R2Score
Multiple Linear Regression	0.789479035
Support Vector Machine	0.595089
Decision Tree	0.708199
Random Forest	**0.870614**