Assignment-Regression Algorithm

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:***

They provide dataset in excel sheet. So we will take machine learning. After requirement is clear. Input and output are present here. So we will take Supervised learning. Then out put's are numerical value so we take regression.

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

The dataset in 6 column and 1338 Rows

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

Dataset in 2 column are categorical values 1.sex 2.smoker sex column in male/female. Smoker column in yes/no I take one hot encoding .

dataset=pd.get_dummies(dataset,drop_first=True)

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.***

<u>The random forest use R2 value (max samples=100) value is 0.8891614314936465</u>

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

Linear Regression				
Parameters		***parameters***		***R2_value***
fit_intercept	***True***	***copy_X***	***True***	**0.7891345484786**
n_jobs	***None***	***positive***	***True***	**0.7890989734446606**

The Linear regression use R2 value (n_jobs=None,positive=True) value is 0.7890989734446606

Support Vector Machine				
Parameters		***parameters***		***R2_value***
kernel	***rbf***	***degree***	***3***	**-0.088442509991301**
kernel	***linear***	***degree***	***3***	**-0.111536454002005**
kernel	***poly***	***degree***	***5***	**-0.064569828857377**
kernel	***sigmoid***	***degree***	***5***	**-0.089943469577214**
gamma	***0.5***	***coef0***	***1.0***	**-0.089642050059384**
tol	***0.001***	***C***	***1.0***	**-0.088442509991301**
epsilon	***1.1***	***shrinking***	***False***	**-0.088442509991301**
cache_size	***200***	***verbose***	***False***	**-0.088442509991301**
max_iter	***-1***	***verbose***	***False***	**-0.088442509991301**

The Support vector machine use R2 value (kernel=poly,degree=5) value is -0.064569828857377

Decision Tree				
Parameters		***parameters***		***R2_value***
criterion	***squared_error***	***splitter***	***best***	**0.7085743991050055**
criterion	***friedman_mse***	***splitter***	***best***	**0.6918373077227303**
criterion	***absolute_error***	***splitter***	***best***	**0.68102966744982**
criterion	***poisson***	***splitter***	***best***	**0.6930956538848834**
criterion	***squared_error***	***splitter***	***random***	**0.7234007222004302**
criterion	***friedman_mse***	***splitter***	***random***	**0.7466273654674349**
criterion	***absolute_error***	***splitter***	***random***	**0.7662068106405571**
criterion	***poisson***	***splitter***	***random***	**0.703069509287678**

max depth	***None***	***min samples split***	***500***	**0.723568879780053**
min samples leaf	***50***	***min weight fraction leaf***	***0.0***	**0.863184657936654**
max features	***sqrt***	***random state***	***None***	**0.770134629785375**
max features	***log2***	***random state***	***None***	**0.766855863238149**
max leaf nodes	***None***	***min impurity decrease***	***1.1***	**0.6913411253284489**
ccp alpha	***1.5***			**0.7075014399396881**

The Decision Tree use R2 value (criterion=absolute error,splitter=random) value is 0.7662068106405571

Random Forest				
Parameters		***parameters***		***R2 value***
n estimators	***100***	***criterion***	***squared_error***	**0.8520558149800667**
n estimators	***100***	***criterion***	***absolute_error***	**0.858586910200183**
n estimators	***100***	***criterion***	***friedman_mse***	**0.8551586327393688**
n estimators	***100***	***criterion***	***poisson***	**0.8532299098269371**
max depth	***None***	***min samples split***	***20***	**0.882648852243253**
min_samples_leaf	***50***	***min_weight_fraction_leaf***	***0.0***	**0.868460781692047**
max features	***1.0***	***max leaf nodes***	***None***	**0.852745592823832**
max features	***sqrt***	***max leaf nodes***	***None***	**0.865065410155875**
max features	***log2***	***max leaf nodes***	***None***	**0.862554737674528**
min impurity decrease	***10.10***	***bootstrap***	***True***	**0.8535293702495795**
oob score	***True***	***n jobs***	***None***	**0.8551238015197515**
random state	***None***	***verbose***	***0***	**0.8538624057792985**
warm start	***True***	***ccp alpha***	***10.10***	**0.852396708566563**

		max_samples	***100***	**0.8891614314936465**

The random forest use R2 value (max_samples=100) value is 0.8891614314936465

Result :

Random forest is the best method for the given for the given model, because when we combine these two parameter

(max_samples=100) we get the highest r2 value and the r2 vale is 0.8891614314936465

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

Why I have take random forest means its r2_score value is nearly 0.90 it is a good model so I taked random forest .

Linear regression,support vector machine,decision tree are values are nearly 0.80 something so I don’t take this.