Assignment-Regression Algorithm

Problem Statement:

To predict the insurance charges based on few parameters which are given in the Client's Dataset.

Parameters - ['age', 'bmi', 'children', 'charges', 'sex_male', 'smoker_yes']

Dataset:

The dataset has 6 columns and 1339 rows. The data has customer related information and insurance charges to be used in all possible algorithms to train the model.

Input and Output classification:

Input - ['age', 'bmi', 'children', 'sex_male', 'smoker_yes']

We must predict the insurance charges for given input parameters. So, we took "charges" as output parameters.

Output – ['charges']

Pre-processing Method:

AI will not able handle the categorical data (as string) which contains in the part of input data. So, we must convert them into equal numeric data for further process. For that, we are used **get_dummies** method from Panda's library to convert them into numeric data.

Model accuracy test with R2 value:

R2 value is mainly used to check the model accuracy. So based on this value, we will be deciding the right algorithm and model.

The below machine learning algorithms are used to find the R2 Value .

Multi Linear Regression:

R2 Value = 0.7894790349867009

Support Vector Machine:

HYPER PARAMETER		
Kernal	**C**	**R2 Value**
linear	1.0	-0.010102665316081394
Poly	1.0	0.038716222760231456
rbf	1.0	-0.08338238593619329
sigmoid	1.0	-0.07542924281107188
linear	10	0.46246841423396834
Poly	10	0.038716222760231456
rbf	10	-0.03227329390671052

sigmoid	10	0.03930714378274347
linear	50	0.6093360196637525
Poly	50	0.4152249293360689
rbf	50	0.14783522804823968
sigmoid	50	0.5276103546510407
linear	100	0.6288792857320369
Poly	100	0.6179569624059795
rbf	100	0.3200317832050831
sigmoid	100	0.5276103546510407

Decision Tree:

Criterion	Splitter	max_features	R2 Value
squared_error	best	sqrt	0.6478653265571128
friedman_mse	best	sqrt	0.6984962414228635
absolute_error	best	sqrt	0.7243889903503341
poisson	best	sqrt	0.7141710683857788
squared_error	random	sqrt	0.6482990197785635
friedman_mse	random	sqrt	0.6841266748696917
absolute_error	random	sqrt	0.7257890253339843
poisson	random	sqrt	0.6338081287355766
squared_error	best	log2	0.7035998855195471
friedman_mse	best	log2	0.5653317189953121
absolute_error	best	log2	0.7157326241470466
poisson	best	log2	0.7248292718883504
squared_error	random	log2	0.6746003563755496
friedman_mse	random	log2	0.6423512490267965
absolute_error	random	log2	0.7347729066487239
poisson	random	log2	0.7246602572123371

Random Forest:

Criterion	n_estimators	max_features	R2 Value
squared_error	10	sqrt	0.8520006346682765
friedman_mse	10	sqrt	0.8502777994291519
absolute_error	10	sqrt	0.8574290080917196
poisson	10	sqrt	0.8544955286235119

squared_error	50	sqrt	0.8695836787761578
friedman_mse	50	sqrt	0.8702417511198071
absolute_error	50	sqrt	0.8708144250343052
poisson	50	sqrt	0.8632391369285537
squared_error	10	log2	0.8520006346682765
friedman_mse	10	log2	0.8502777994291519
absolute_error	10	log2	0.8574290080917196
poisson	10	log2	0.8544955286235119
squared_error	50	log2	0.8695836787761578
friedman_mse	50	log2	0.8702417511198071
absolute_error	50	log2	0.8708144250343052
poisson	50	log2	0.8632391369285537

Final Decision:

We are finally decided to create a model with "Random Forest" algorithm. It has 0.87 % of accuracy than other algorithms based on R2 value which we calculated with various algorithms.

Thank you !