Regression Assignment:

Problem identification & Programming
Question : Client wants to predict insurance charges based on the several paramaters. The client has provided the dataset of the same.

Solutions:

1. Problem identification:
Stage 1:
Based on the Dataset provided by Client, AI domain is confirmed as Machine learning based on the output in numerical in form of “insurance charges.”
Stage 2:
The Dataset contains both Input and output values so it is confirmed as Suprevised learning.
Stage 3:
Prediction of the problem is a Numerical so it is confirmed as Regression Algorithm under Machine Learning.

Regression:
Generally Regression means low errors.
Types of Regression Algorithms they are:-
1.Simple Linear Regression – One input and a output
2.Multiple Linear Regression – Multiple input and a output
3.Support Vector Machine – Supports Non Linear Algorithm means the graph of this algorithm will not form a straight line. Otherwise called as non-separable dataset.
4. Random Forest – non orderless tress
5. Decision Tree – Decision done based on condition from the root which forms a tree like structure which consist many branch and sub-branches. Every branch called as Split or segregation, Removal of sub branches(sub-nodes of a branch(decision node) called pruning.

Above Problem Statement is related Multiple Linear Regression based on Multiple Inputs and a Output.

2. Information about Dataset.

a. Input Fields such as age, bmi, children, sex_female,sex_males, smoker_yes, smoker_no
b. Output Fields such as Insurance charges.
c. Total No of rows 1338(incl input & output) & no of columns 8(incl input & output).
d. Train set [896 rows x 1 columns], Test set [442 rows x 1 columns]

3. Pre-processing procedure:

AI algorithm using python cannot understand categorical value so converting to numerical by using two thing they are nominal phase – non-comparable thing, ordinal phase – comparable thing.

4. Finalizing best model using best R2 value:
A.
Multiple Linear Regression R2 value : .7899

```
#model creation and training y=m*x + c
from sklearn.linear_model import LinearRegression
model_create = LinearRegression()
model_create.fit(X_train,Y_train)
```

B.
Support Vector Machine
a. SVM R2 value : -0.0979
#Model Creation & Training
from sklearn.svm import SVR
model_create = SVR(kernel = "rbf")
model_create.fit(X_train, Y_train)

model_create = SVR(kernel = "rbf",C = 2000)
R2 value: 0.8550
model_create = SVR(kernel = "rbf",C = 3000)
R2 value : 0.8648

b. #Standardisation procedure to maintain least different between inputs such as age & bmi:
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
X_train=sc.fit_transform(X_train)
X_test=sc.transform(X_test)
SVM R2 value : -0.0923

C.RandomForest:

#Creating & Training the model.
from sklearn.ensemble import RandomForestRegressor
model_create = RandomForestRegressor(criterion = "absolute_error",n_estimators=1000, max_features="log2")
model_create = model_create.fit(X_train,Y_train)
R2 value : 0.8727

D. DecisionTree:
#Model creation and training
from sklearn.tree import DecisionTreeRegressor
model_create = DecisionTreeRegressor(criterion="mae",splitter="best", max_features = "sqrt")
model_create = model_create.fit(X_train,Y_train)
R2 value : 0.7095