TO find R2 score value by using following machine learning method:

1. Multiple Linear Regression:

R2 Score value in **MLR** = **0.7894790349867009**

2.Support Vector Machine:

Kernal	C=1	C=10	C=100	C=1000
Linear	-0.010102665316081394	0.4624684142339678	0.6288792857320361	0.7649311738597825
rbf	-0.08338238593619329	-0.03227329390671052	0.3200317832050831	0.8102064874808204
Poly	-0.0756996557086089	0.038716222760231456	0.6179569624059799	0.856648767594659
sigmoid	-0.07542924281107188	0.03930714378274347	0.5276103546510407	0.2874706948697563

In svm (kernal=linear,c=1) gives r2_square value is 0.856648767594659

3. DecisionTree:

Splitter =	*criterion=squared_error*	*criterion=friedman_mse*	*criterion= absolute_error*	*criterion=poisson*
best	**0.6822501596455095**	0.6967668434689598	**0.6962255721738235**	**0.7185090770022624**
random	**0.6852272931823247**	**0.7399979459175379**	**0.747379837018312**	**0.7244763266001724**

Here (criterion="squared_error", splitter="best") gives model r2_square value is 0.747379837018312

4.Random Forest:

criterion	n_estimators	max_features	R2_score
squared_error	**100**	***sqrt***	***0.8722842204761092***
absolute_error	**100**	***sqrt***	***0.8757337256192934***
friedman_mse	**100**	***sqrt***	***0.8680297251949043***
poisson	**100**	***sqrt***	***0.8716059600152415***
squared_error	**100**	***log2***	***0.8712922689780294***
absolute_error	**100**	***log2***	***0.8723776640868625***
friedman_mse	**100**	***log2***	***0.871378594460033***
poisson	**100**	***log2***	***0.8721964653950868***

squared_error	**100**	***None***	***0.8547494377130381***
absolute_error	**100**	***None***	***0.8499397360490405***
friedman_mse	**100**	***None***	***0.8549703309699938***
poisson	**100**	***None***	***0.8556145935293948***
squared_error	**500**	***sqrt***	***0.8718583712387833***
absolute_error	**500**	***sqrt***	***0.8746432159550745***
friedman_mse	**500**	***sqrt***	***0.8720804525715373***
poisson	**500**	***sqrt***	***0.873404730521126***
squared_error	**500**	***log2***	***0.8721191310381918***
absolute_error	**500**	***log2***	***0.8738158110772026***
friedman_mse	**500**	***log2***	***0.8718839306472905***
poisson	**500**	***log2***	***0.8709770263689216***
squared_error	**500**	***None***	***0.8562648066140802***
absolute_error	**500**	***None***	***0.8537889935368864***
friedman_mse	**500**	***None***	***0.852692804611620***
poisson	**500**	***None***	***0.855938083383492***
squared_error	**1000**	***sqrt***	***0.8720462505768847***
absolute_error	**1000**	***sqrt***	***0.8738259802417034***

friedman_mse	**1000**	*sqrt*	***0.8729093810690829***
poisson	**1000**	*sqrt*	***0.8718917358341729***
squared_error	**1000**	*log2*	***0.8722115789189225***
absolute_error	**1000**	*log2*	***0.8740641167816261***
friedman_mse	**1000**	*log2*	***0.8724954886409342***
poisson	**1000**	*log2*	***0.8715554217382954***
squared_error	**1000**	*None*	***0.8544108919870267***
absolute_error	**1000**	*None*	***0.8551830267037819***
friedman_mse	**1000**	*None*	***0.8548508184154748***
poisson	**1000**	*None*	***0.8552923617139219***

Best model criterion=*absolute_error*”;n_estimators=100;max_features='sqrt’ Gives R2_SCORE Value is 0.*8757337256192934*

5.ADA BOOOSTING:

N_estimators	Loss	r2_score
1	Square	0.8809576998632572
1	linear	0.8809576998632572

1	exponential	0.8809576998632572
50	square	0.5185321334257109
50	linear	0.8447476929154872
50	exponential	0.6292859305668668
100	square	0.4661497994260698
100	linear	0.8447476929154872
100	exponential	0.538576314578121
500	square	0.4661497994260698
500	linear	0.8447476929154872
500	exponential	0.538576314578121

Best model n_estimators=1;loss='square,linear,exponential' Gives R2_SCORE Value is 0.8809576998632572

6.LightGradient Boosting(LGBM):

loss	criterion	max_features	r2_squre
Squared error	*friedman_mse*	None	0.8833974199848822
Absolute _error	*friedman_mse*	None	0.8588019148018358
huber	*friedman_mse*	None	0.8914390761897757
quantile	*friedman_mse*	None	0.6289330239690663

Squared error	*friedman_mse*	**sqrt**	**0.8900729638277464**
absolute_error	*friedman_mse*	**sqrt**	**0.8767758265920338**
huber	*friedman_mse*	**sqrt**	**0.8904302901210221**
quantile	*friedman_mse*	**sqrt**	**0.6600838602848385**
squared_error	*friedman_mse*	**log2**	**0.8900729638277464**
absolute_error	*friedman_mse*	**log2**	**0.8767758265920338**
huber	*friedman_mse*	**log2**	**0.8904302901210221**
quantile	*friedman_mse*	**log2**	**0.6600838602848385**
squared_error	*squared_error*	**None**	**0.8833974199848822**
absolute_error	*squared_error*	**None**	**0.8588019148018358**
huber	*squared_error*	**None**	**0.8914390761897757**
quantile	*squared_error*	**None**	**0.6266819312165877**
Squared error	*squared_error*	**sqrt**	**0.8900729638277464**
Absolute _error	*squared_error*	**sqrt**	**0.8767758265920338**
huber	*squared_error*	**sqrt**	**0.8904302901210221**
quantile	*squared_error*	**sqrt**	**0.6597975662815307**
squared_error	*squared_error*	**log2**	**0.8900729638277464**

absolute_error	*squared_error*	**log2**	**0.8767758265920338**
huber	*squared_error*	**log2**	**0.8904302901210221**
quantile	*squared_error*	**log2**	**0.6597975662815307**

Best model loss='huber ', criterion='squared_error,*friedman_mse*' , max_features='None' Gives R2_SCORE Value is 0.8914390761897757

7. Extreme Gradient Boosting :

verbosity= 0,1,2,3; booster=gbtree, gblinear; tree_method=auto, exact, approx, hist

n_estimators	learning_rate	max_depth	min_child_weight	verbosity	tree_method	R2_Score
10	0.1	1	10	0	auto	0.5572062730789185
10	0.1	1	10	0	exact	0.5572062730789185
10	0.1	1	10	0	approx	0.5572062730789185
10	0.1	1	10	0	hist	0.5572062730789185
10	0.1	1	10	1	auto	0.5572062730789185
10	0.1	1	10	1	exact	0.5572062730789185
10	0.1	1	10	1	approx	0.5572062730789185
10	0.1	1	10	1	hist	0.5572062730789185
10	0.1	3	20	2	auto	0.7720925807952881
10	0.1	3	20	2	exact	0.7719619870185852

10	0.1	3	20	2	approx	0.7720925807952881
10	0.1	3	20	2	hist	0.7720925807952881
10	0.1	5	50	3	auto	0.7559833526611328
10	0.1	5	50	3	exact	0.7556763887405396
10	0.1	5	50	3	approx	0.7559833526611328
10	0.1	5	50	3	hist	0.7559833526611328
40	0.1	1	10	0	auto	0.7502782344818115
40	0.1	1	10	0	exact	0.7503102421760559
40	0.1	1	10	0	approx	0.7502782344818115
40	0.1	1	10	0	hist	0.7502782344818115
40	0.1	1	10	1	auto	0.7502782344818115
40	0.1	1	10	1	exact	0.7503102421760559
40	0.1	1	10	1	approx	0.7502782344818115
40	0.1	1	10	1	hist	0.7502782344818115
40	0.1	3	20	0	auto	0.8945130705833435
40	0.1	3	20	0	exact	0.8941095471382141
40	0.1	3	20	0	approx	0.8945130705833435
40	0.1	3	20	0	hist	0.8945130705833435

40	0.1	3	20	1	auto	0.8945130705833435
40	0.1	3	20	1	exact	0.8941095471382141
40	0.1	3	20	1	approx	0.8945130705833435
40	0.1	3	20	1	hist	0.8945130705833435
40	0.1	3	20	2	auto	0.8945130705833435
40	0.1	3	20	2	exact	0.8941095471382141
40	0.1	3	20	2	approx	0.8945130705833435
40	0.1	3	20	2	hist	0.8945130705833435
40	0.1	5	50	3	auto	0.886320173740387
40	0.1	5	50	3	exact	0.8869958519935608
40	0.1	5	50	3	approx	0.886320173740387
40	0.1	5	50	3	hist	0.886320173740387
100	0.1	1	10	0	auto	0.7909070253372192
100	0.1	1	10	0	exact	0.7909550666809082
100	0.1	1	10	0	approx	0.7909070253372192
100	0.1	1	10	0	hist	0.7909070253372192
100	0.1	1	10	1	auto	0.7909070253372192
100	0.1	1	10	1	exact	0.7909550666809082

100	0.1	1	10	1	approx	0.7909070253372192
100	0.1	1	10	1	hist	0.7909070253372192
100	0.1	3	20	1	auto	0.8905522227287292
100	0.1	3	20	1	exact	0.8899841904640198
100	0.1	3	20	1	approx	0.8905522227287292
100	0.1	3	20	1	hist	0.8905522227287292
100	0.1	3	20	2	auto	0.8905522227287292
100	0.1	3	20	2	exact	0.8899841904640198
100	0.1	3	20	2	approx	0.8905522227287292
100	0.1	3	20	2	hist	0.8905522227287292
100	0.1	5	50	3	auto	0.8821432590484619
100	0.1	5	50	3	exact	0.8829681873321533
100	0.1	5	50	3	approx	0.8821432590484619
100	0.1	5	50	3	hist	0.8821432590484619