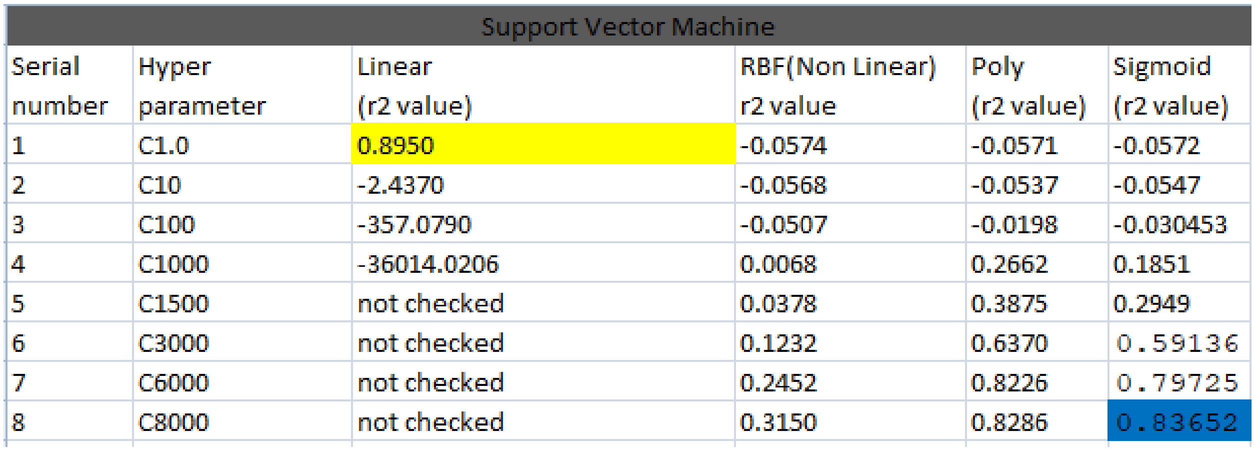
# Machine Learning Regression

## Algorithm Validation:

### 1. SVM (Support Vector Machine)



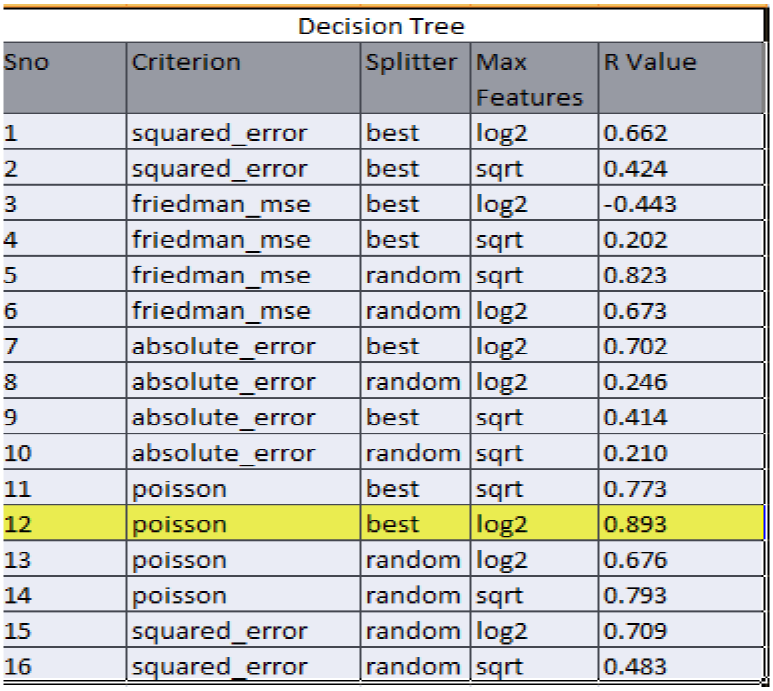
Support Vector Machine

| Serial number | Hyper parameter | Linear (r2 value) | RBF(Non Linear) r2 value | Poly (r2 value) | Sigmoid (r2 value) |
|---|---|---|---|---|---|
| 1 | C1.0 | 0.8950 | -0.0574 | -0.0571 | -0.0572 |
| 2 | C10 | -2.4370 | -0.0568 | -0.0537 | -0.0547 |
| 3 | C100 | -357.0790 | -0.0507 | -0.0198 | -0.030453 |
| 4 | C1000 | -36014.0206 | 0.0068 | 0.2662 | 0.1851 |
| 5 | C1500 | not checked | 0.0378 | 0.3875 | 0.2949 |
| 6 | C3000 | not checked | 0.1232 | 0.6370 | 0.59136 |
| 7 | C6000 | not checked | 0.2452 | 0.8226 | 0.79725 |
| 8 | C8000 | not checked | 0.3150 | 0.8286 | 0.83652 |

The Kernel type Linear $R^2$ value is 0.8950.This model looks good (1st choice)

The Kernel type Sigmoid $R^2$ value is 0.836(2nd choice)

### 2. Decision Tree



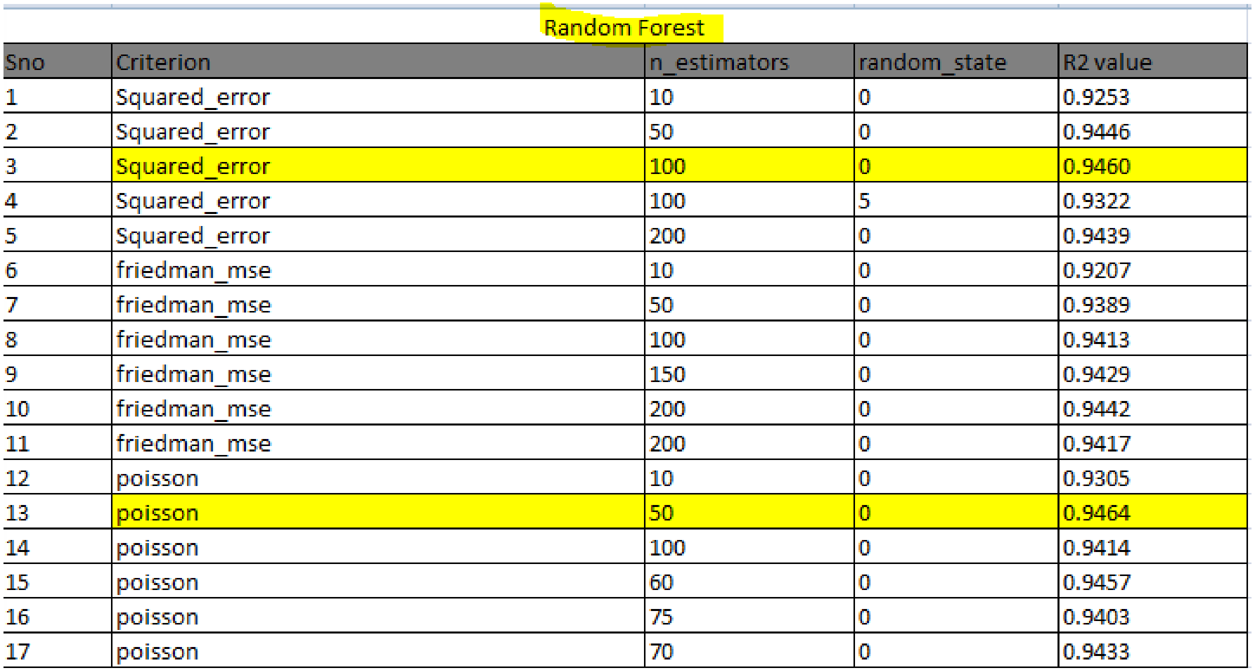
Decision Tree

| Sno | Criterion | Splitter | Max Features | R Value |
|---|---|---|---|---|
| 1 | squared_error | best | log2 | 0.662 |
| 2 | squared_error | best | sqrt | 0.424 |
| 3 | friedman_mse | best | log2 | -0.443 |
| 4 | friedman_mse | best | sqrt | 0.202 |
| 5 | friedman_mse | random | sqrt | 0.823 |
| 6 | friedman_mse | random | log2 | 0.673 |
| 7 | absolute_error | best | log2 | 0.702 |
| 8 | absolute_error | random | log2 | 0.246 |
| 9 | absolute_error | best | sqrt | 0.414 |
| 10 | absolute_error | random | sqrt | 0.210 |
| 11 | poisson | best | sqrt | 0.773 |
| 12 | poisson | best | log2 | 0.893 |
| 13 | poisson | random | log2 | 0.676 |
| 14 | poisson | random | sqrt | 0.793 |
| 15 | squared_error | random | log2 | 0.709 |
| 16 | squared_error | random | sqrt | 0.483 |

The Decision tree algorithm gives a better $R^2$ value (0.893) when using

Criterion =poisson, splitter=best and max_features=log2

3. Random Forest Algorithm



Random Forest

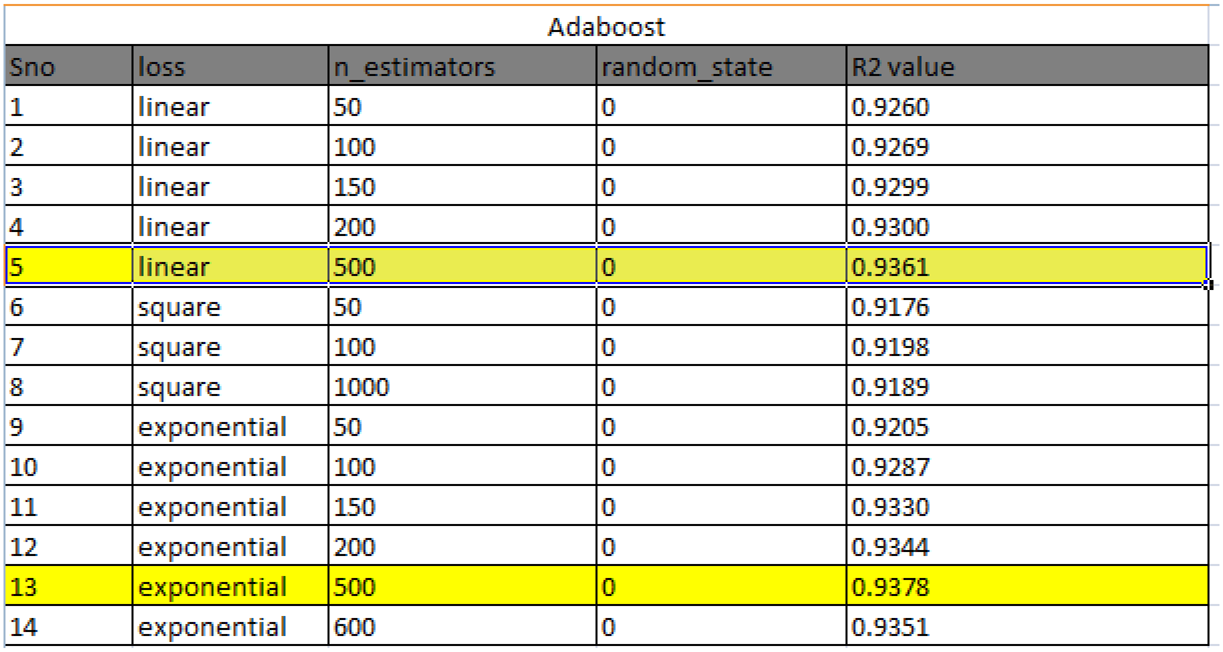
| Sno | Criterion | n_estimators | random_state | R2 value |
|---|---|---|---|---|
| 1 | Squared_error | 10 | 0 | 0.9253 |
| 2 | Squared_error | 50 | 0 | 0.9446 |
| 3 | Squared_error | 100 | 0 | 0.9460 |
| 4 | Squared_error | 100 | 5 | 0.9322 |
| 5 | Squared_error | 200 | 0 | 0.9439 |
| 6 | friedman_mse | 10 | 0 | 0.9207 |
| 7 | friedman_mse | 50 | 0 | 0.9389 |
| 8 | friedman_mse | 100 | 0 | 0.9413 |
| 9 | friedman_mse | 150 | 0 | 0.9429 |
| 10 | friedman_mse | 200 | 0 | 0.9442 |
| 11 | friedman_mse | 200 | 0 | 0.9417 |
| 12 | poisson | 10 | 0 | 0.9305 |
| 13 | poisson | 50 | 0 | 0.9464 |
| 14 | poisson | 100 | 0 | 0.9414 |
| 15 | poisson | 60 | 0 | 0.9457 |
| 16 | poisson | 75 | 0 | 0.9403 |
| 17 | poisson | 70 | 0 | 0.9433 |

Random Forest Algorithm gives a better $R^2$ value (0.9460) while using the following

Criterion: Squared_error , n_estimators=100 and random_state=0

Criterion: poisson , n_estimators=50 and random_state=0

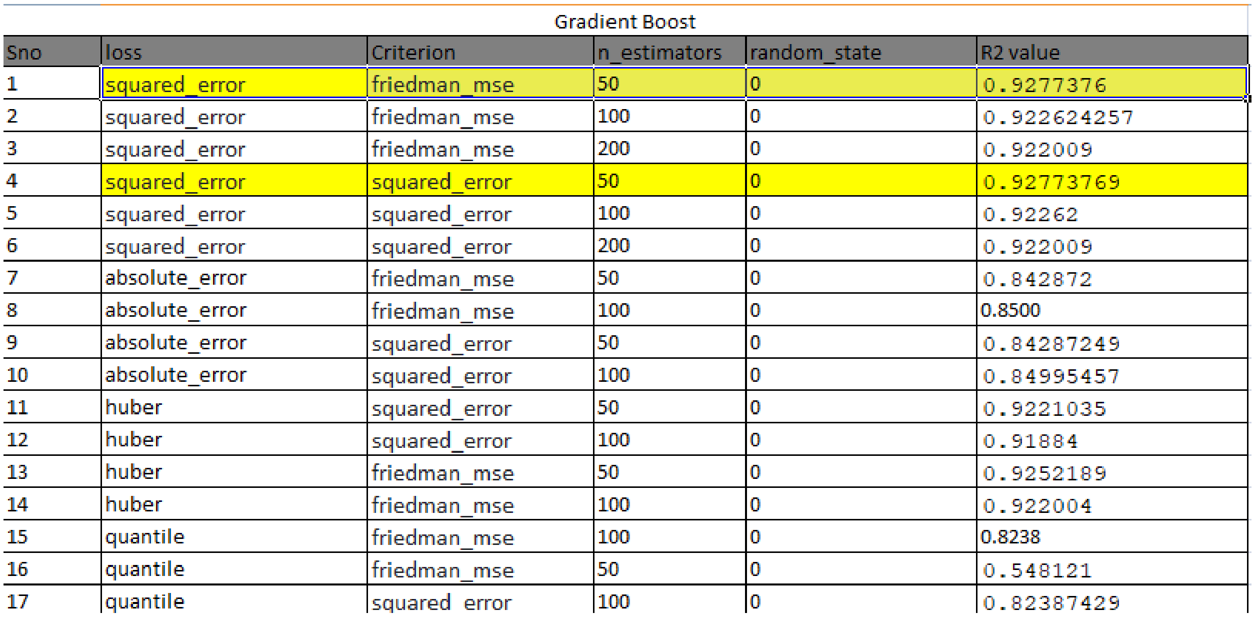
4.Adaboost algorithm



Adaboost

| Sno | loss | n_estimators | random_state | R2 value |
|---|---|---|---|---|
| 1 | linear | 50 | 0 | 0.9260 |
| 2 | linear | 100 | 0 | 0.9269 |
| 3 | linear | 150 | 0 | 0.9299 |
| 4 | linear | 200 | 0 | 0.9300 |
| 5 | linear | 500 | 0 | 0.9361 |
| 6 | square | 50 | 0 | 0.9176 |
| 7 | square | 100 | 0 | 0.9198 |
| 8 | square | 1000 | 0 | 0.9189 |
| 9 | exponential | 50 | 0 | 0.9205 |
| 10 | exponential | 100 | 0 | 0.9287 |
| 11 | exponential | 150 | 0 | 0.9330 |
| 12 | exponential | 200 | 0 | 0.9344 |
| 13 | exponential | 500 | 0 | 0.9378 |
| 14 | exponential | 600 | 0 | 0.9351 |

## 5. Gradient Boost  Algorithm



Gradient Boost

| Sno | loss | Criterion | n_estimators | random_state | R2 value |
|---|---|---|---|---|---|
| 1 | squared_error | friedman_mse | 50 | 0 | 0.9277376 |
| 2 | squared_error | friedman_mse | 100 | 0 | 0.922624257 |
| 3 | squared_error | friedman_mse | 200 | 0 | 0.922009 |
| 4 | squared_error | squared_error | 50 | 0 | 0.92773769 |
| 5 | squared_error | squared_error | 100 | 0 | 0.92262 |
| 6 | squared_error | squared_error | 200 | 0 | 0.922009 |
| 7 | absolute_error | friedman_mse | 50 | 0 | 0.842872 |
| 8 | absolute_error | friedman_mse | 100 | 0 | 0.8500 |
| 9 | absolute_error | squared_error | 50 | 0 | 0.84287249 |
| 10 | absolute_error | squared_error | 100 | 0 | 0.84995457 |
| 11 | huber | squared_error | 50 | 0 | 0.9221035 |
| 12 | huber | squared_error | 100 | 0 | 0.91884 |
| 13 | huber | friedman_mse | 50 | 0 | 0.9252189 |
| 14 | huber | friedman_mse | 100 | 0 | 0.922004 |
| 15 | quantile | friedman_mse | 100 | 0 | 0.8238 |
| 16 | quantile | friedman_mse | 50 | 0 | 0.548121 |
| 17 | quantile | squared_error | 100 | 0 | 0.82387429 |