**Machine Learning Algorithm**

**Support Vector Machine**

***class*sklearn.svm.SVR(***\****, *kernel****='rbf'***, *degree****=3***, *gamma****='scale'***, *coef0****=0.0***, *tol****=0.001***, *C****=1.0***, *epsilon****=0.1***, *shrinking****=True***, *cache\_size****=200***, *verbose****=False***, *max\_iter****=-1***)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kernel** | **C** | **Coef0** | **R\_Score** |
| Linear | 1.0 | 0.0 | 0.89 |
| Rbf(default) | 10000 | 0.0 | 0.37 |
| poly | 100 | 100 | 0.90 |
| Sigmoid | 10000 | 1000,epsilon=1 | -0.57 |

**Decision Tree**

***class*sklearn.tree.DecisionTreeRegressor(***\****, *criterion****='squared\_error'***, *splitter****='best'***, *max\_depth****=None***, *min\_samples\_split****=2***, *min\_samples\_leaf****=1***, *min\_weight\_fraction\_leaf****=0.0***, *max\_features****=None***, *random\_state****=None***, *max\_leaf\_nodes****=None***, *min\_impurity\_decrease****=0.0***, *ccp\_alpha****=0.0***, *monotonic\_cst****=None***)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criterion** | **Splitter** | **max\_depth** | **Random State** | **R\_Score** |
| Squared error(default) | Best | None | None | 0.92 |
| friedman\_mse | Best | None | None | 0.91 |
| absolute\_error | Best | None | None | 0.95 |
| poisson | Best | None | None | 0.91 |
| Squared error | random | None | None | 0.65 |
| friedman\_mse | random | None | None | 0.65 |
| absolute\_error | Random | None | None | 0.72 |
| poisson | random | None | None | 0.65 |
| Squared error | Best | 1 | 1 | 0.27 |
| friedman\_mse | Best | 1 | 1 | 0.27 |
| absolute\_error | Best | 1 | 1 | 0.26 |
| poisson | Best | 1 | 1 | 0.06 |
| Squared error | random | 1 | 1 | -0.03 |
| friedman\_mse | Random | 1 | 1 | -0.03 |
| absolute\_error | Random | 1 | 1 | -0.02 |
| poisson | random | 1 | 1 | -0.03 |
| friedman\_mse | Best | 2 | 1 | 0.78 |
| absolute\_error | Best | 2 | 1 | 0.69 |
| Squared error | Best | 2 | None | 0.78 |
| poisson | random | 2 | None | 0.70 |