AdaBoost:

| Estimator | # n_estimator | # Learning_Rate | Loss | Random_State | # R2_score |
|---|---|---|---|---|---|
| None | 50 | 1 | Linear | None | 0.84615 |
| None | 50 | 1 | Square | None | 0.51845 |
| None | 50 | 1 | Exponential | None | 0.61534 |
| None | 100 | 1 | Linear | None | 0.85216 |
| None | 100 | 1 | Square | None | 0.49955 |
| None | 100 | 1 | Exponential | None | 0.53765 |
| None | 200 | 1 | Linear | None | 0.85323 |
| None | 200 | 1 | Square | None | 0.45852 |
| None | 200 | 1 | Exponential | None | 0.51716 |
| None | 300 | 1 | Linear | None | 0.86884 |
| None | 300 | 1 | Square | None | 0.45712 |
| None | 300 | 1 | Exponential | None | 0.48972 |
| None | 400 | 1 | Linear | None | 0.86715 |
| None | 400 | 1 | Square | None | 0.45112 |
| None | 400 | 1 | Exponential | None | 0.48207 |
| None | 500 | 1 | Linear | None | 0.85725 |
| None | 500 | 1 | Square | None | 0.447406 |
| None | 500 | 1 | Exponential | None | 0.47607 |

XGBOOST:

| n_estimators | max_depth | eta | subsample | colsample_bytree | loss | booster | r2score |
|---|---|---|---|---|---|---|---|
| 100 | 7 | 0.1 | 0.7 | 0.8 | Squared_Error | gbtree | 0.81798 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | Squared_Error | gblinear | 0.71768 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | Squared_Error | dart | 0.87486 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | absolute_error | gbtree | 0.76745 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | absolute_error | gblinear | 0.71523 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | absolute_error | dart | 0.763108 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | huber | gbtree | 0.82504 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | huber | gblinear | 0.733105 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | huber | dart | 0.80441 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | Quantile | gbtree | 0.80094 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | Quantile | gblinear | 0.69832 |
| 100 | 7 | 0.1 | 0.7 | 0.8 | Quantile | dart | 0.81376 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | Squared_Error | gbtree | 0.83726 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | Squared_Error | gblinear | 0.72345 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | Squared_Error | dart | 0.81443 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | absolute_error | gbtree | 0.77589 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | absolute_error | gblinear | 0.73882 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | absolute_error | dart | 0.82635 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | huber | gbtree | 0.80014 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | huber | gblinear | 0.72037 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | huber | dart | 0.79243 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | Quantile | gbtree | 0.83039 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | Quantile | gblinear | 0.83039 |
| 500 | 7 | 0.1 | 0.7 | 0.8 | Quantile | dart | 0.79408 |

LGBM:

| n_estimators | learning_rate | boosting_type | r2_score |
|---|---|---|---|
| 50 | 0.1 | gbdt | 0.79849 |
| 50 | 0.1 | dart | 0.86154 |
| 100 | 0.1 | gbdt | 0.83048 |
| 100 | 0.1 | dart | 0.80595 |
| 500 | 0.1 | gbdt | 0.79859 |
| 500 | 0.1 | dart | 0.80007 |

Conclusion:

The Insurance charge prediction data set performed good  in the boosting algorithms ie.

AdaBoost regressor,XGBoost regressor, LGBM regressor. The r2score value for these boosting algorithms are  < than 0.99. By comparing these algorithm XGboost  perfomed 1% greater than the Adaboost and LGBM.