The adjusted R-squared is a modified version of R-squared that accounts for predictors that are not significant in a regression model. In other words, the adjusted R-squared shows whether adding additional predictors improve a regression model or not.

The Mean Squared Error measures how close a regression line is to a set of data points. It is a risk function corresponding to the expected value of the squared error loss.

A screenshot of a computer

Description automatically generated

There is a lot of randomness, for each run the score changes. At this run we can see that the best performer is ‘VotingRegressor’ model with highest score of Adj. R-squared and R-squared 0.3995 and 0.4136, also it has the lowest mean-squared-error (MSE) of 14.4298.

The worst model performer is ‘KNeighborsRegressor’, which has the lowest Adj R-squared and R-squared scores of 0.1137 and 0.1579, also it has the highest score for the mean-squared-error (MSE) of 20.7217.

‘LinearRegression’ and Ridge models are the second-best performers with 0.3665 and 0.3814 scores for Adj. R-squared and R-squared, and their MSE scores are around 15.22.

The Support Vector Regression (SVR) and ‘DecisionTreeRegressor’ models are with the lower-level performance regarding Adj. R-squared with 0.3551 and 0.3598 scores, also have higher MSE error scores of 15.4974 and 15.3825.

The ‘[VotingRegressor](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingRegressor.html" \l "sklearn.ensemble.VotingRegressor" \o "sklearn.ensemble.VotingRegressor)’ combines conceptually different machine learning regressors and return the average predicted values. Such a regressor can be useful for a set of equally well performing models in order to balance out their individual weaknesses, like last two models (‘KNeighborsRegressor’, ‘TransformedTargetRegressor’)

A ‘VotingRegressor’ can be defined as a special method that combines or ‘ensembles’ multiple regression models and overperforms the individual models present as its estimators.

Hey Vincent,

Negative **R^2 score** means your model fits the data very poorly. In this case ‘DecisionTreeRegressor’ may be too simple, because you have used the default parameters, if you increase the max\_dept to 4 or 5, you will see some changes in the score.

Another choice is to try tune your model's hyperparameters.

The deafult constactor, it does not give good scores for KNN and DTREE, try to tweek them and will see the difrence of the scores, but it is very random for each run.

knn = KNeighborsRegressor(n\_neighbors=5, weights='uniform')  
dt = DecisionTreeRegressor(max\_depth=4)

Hey Jim,

Have a look at my notebook, how you can create a html table, by passing a panda dataframe to the plotly function

from plotly.figure\_factory import create\_table

* **Bias**

**Bias** is exhibited by models that are **too simple** for the data they are trying to match. For example, if you tried to capture the circular red class shown with a **linear decision boundary**, this will invariably produce large errors.

High bias is

A model with a higher bias would not match the data set closely. A low bias model will closely match the training data

A high bias model typically includes more assumptions about the target function or end result. A low bias model incorporates fewer assumptions about the target function.

High variance is deep trees

* To mitigate high bias you increase the dept of the tree – boosting solution, Adding ensemble members sequentially
* To mitigate high variance, you do averaging – bagging solution, Fitting a number of decision trees on samples

The boosting algorithm:

▪ Reduces the bias of weak learners

▪ Does not increase the variance as much as other algorithms

Gradient-boosted trees:

▪ Uses trees for its base model – either decision trees or regression trees

▪ Uses the squared loss as its cost function and cannot be implemented as a simple weight updating scheme

deep trees have high variance and low bias.

the averaging step will help to reduce the variance, you are not so concerned with the component models having a high variance, but you do want to choose models with low bias.

allow each of the trees to grow until all of their leaves are pure.

Since the averaging step will help to reduce the variance, you are not so concerned with the component models having a high variance