**4.1 Training Regression Model in Scikit-Learn**

4.1.1 Support Vector Regression Model(SVR)

For the kernel functions, I choose rbf, because of it’s complexity is in average among those models. Which give an average accuracy, not much overfit, and reasonable running time. For example, linear is too simple, it running time is super fast, but the accuracy is really low, and since the result cant be plot by a linear line. For the poly, it is too complex, the running time is like forever, and it is very easy to overfit, therefore I would not like to choose it.

For the epsilon and C. they are both use to prevent overfitting.

The constant C is the box constraint, a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin (ε) and helps to prevent overfitting (regularization). This value determines the trade-off between the flatness of f(x) and the amount up to which deviations larger than ε are tolerated.

The linear ε-insensitive(epsilon) loss function ignores errors that are within ε distance of the observed value by treating them as equal to zero.





The value I choose for epsilon is 0.5, C is 100

As I see the value of the prediction time is quite large, therefore the C can be quite large as well. And the epsilon is better to be larger than the default one (0.1), this may also due to the same reason.

4.1.2 Gradient Boosting Regression Tree Model(GBRT)

For the loss function, I choose least squares(ls),because its superior computational properties. Because other loss function may be not suitable or overfit more. For example, Least absolute deviation ('lad'): The initial model is given by the median of the target values. As our data has result with difference race distance, sample getting the median may not be a good choice. And also other with a larger running time and more easy to overfit, that’s why I use the simple model ‘ls’.

And for the critical parameters, I use learning\_rate: 0.01, n\_estimators:500, max\_depth: 4.

The learning rate is a regularization strategy to protect the model from over fitting. Therefore, I choose 500, a small number when compare to the training data.

Estimators support arguments to control the fitting behavior, it controls the action of early stopping when the model get a pretty good accuracy. It is relate to the number of features, and I choose the value of 500, as we have many features.

Max depth is the max depth of the tree, for a tree with depth n, it will have 2 to the power n of nodes. And when the value of max depth is high, it is easy to get over fitting and the running will be very slow. And so I choose 8, which will have a reasonable running time and accuracy.

For the learning rate, I choose 0.05. Learning rate is a parameter within [0.0, 1.0], it also can prevent overfitting, but too low will also make the model too simple. With trying several different learning rates, I find 0.05 quite fit our model (other parameter). And it small number prevent the model overfit, but also a reasonable value for boosting the running time.