Cross Validation

In machine learning process, a variance problem refers to the scenario where our accuracy obtained on one test is very different to accuracy obtained on another test set using same algorithm. There is always a need to validate the stability of your machine learning model. You need some kind of assurance that your model has got most of the patterns from the data correct and it is not picking up too much on the noise. This assurance can be achieved through Cross validation

Cross validation is a technique to know how our prediction model performs on an unknown dataset. The objective of cross validation is to validate if model is consistent or not.

Cross validation can't be done on unbalanced dataset because model may not be able to learn patterns and thus predicted differ a lot from actual value.

Cross validation gives us a metric (chosen by user) and its standard deviation.

If metric is low, then model is inaccurate. If metric is high, then model is accurate.

If standard deviation is low, the variance is low and thus model is consistent. If it is high, errors are not consistent and are changing drastically from model to model and thus model is inconsistent.

Ensembles models like Random Forest have in built feature of Cross validation.

All cross validation methods follow the below procedure

- Divide the dataset into training and testing

- Train the model on the training set

- Evaluate the model on the testing set

- Repeat steps 1 to 3 for a different set of data points

K-Fold

With K-Fold, we're going to randomly split our dataset into K equally sized parts. We will then train our model K times. For each training run, we select a single partition from our K parts to be the test set and use the rest for training. To obtain a final accuracy measure, we average out the results of each model evaluated on their respective test sets.

Since we use cross validation methods to test the stability of our model, we must first find the optimal value of k, only then we can test the stability of our model. So in k fold cross validation, k must be selected keeping such that we are not increasing bias as well as not increasing variance.

If k is very small, chunks of dataset will be large. Since one large chunk is set aside for testing, we have reduced data for training because of which we won't be able to catch the required patterns and would make a lot of assumptions and thus bias will be increased. So these model which have learnt less and made more assumptions would give low accuracy.

If k=4, fold=500 train size=1500 test size=500

We couldn't catch patterns since test data took lots of samples and train data is small. This increases bias since there is huge difference between actual and predicted value.

If k is very large, chunks of dataset will be small. Since only a small chunk is kept for testing, the error rates while testing model on different chunks of test data would vary a lot and thus variance will be increased. So these models which have learnt lot of noise would give high variance.

If k=10, fold=200 train size=1800 test size=200

Suppose in model m1 10 observations out of 200 were predicted wrong (error rate=5%) but in model m2 60 observations out of 200 were predicted wrong (error rate=30%). This increases variance since there is huge difference in error rate which means error are not consistent.

With a higher number folds, we will be reducing the error due the bias but increasing the error due to variance.

With a lower number of folds, we will be reducing the error due to variance but the error due to bias would be bigger.