How to handle larger dataset?

-Deepanshu Sharma
A023119820026

INTRODUCTION

- Datasets are a collection of instances that all share a common attribute. Machine learning models will generally contain a few different datasets, each used to fulfill various roles in the system.
- When any experienced data scientist is dealing with a project related to ML, 60 percent of work is done for analyzing the dataset, which we call as Exploratory Data Analysis(EDA). That means data plays a major role in machine learning. In the real world, we have huge data to work on, which makes computation and reading data with normal pandas is not feasible as if it will take more time and we generally have limited resources to work on. So to make it feasible, many AI researchers come up with a solution to identify different techniques and ways in dealing with large datasets.

Ways to handle larger dataset:

1.Stream Data or Use Progressive Loading:

- Take a random sample of your data, such as the first. Use the smaller sample to work through your problem before fitting a final model on all of your data.

- I think this is a good practice in general for machine learning to give you quick spot-checks of algorithms and turnaround of results.

- You may also consider performing a sensitivity analysis of the amount of data used to fit one algorithm compared to the model skill. Perhaps there is a natural point of diminishing returns that you can use as a heuristic size of your smaller sample.
- Perhaps you can use code or a library to stream or progressively load data as-needed into memory for training.

2. Change the data format:

- You can speed up data loading and use less memory by using another data format.
- A good example is a binary format like GRIB, NetCDF, or HDF. There are many command-line tools that you can use to transform one data format into another that do not require the entire dataset to be loaded into memory. Using another format may allow you to store the data in a more compact form that saves memory, such as 2-byte integers, or 4-byte floats.
- There are many command line tools that you can use to transform one data format into another that do not require the entire dataset to be loaded into memory.

- Using another format may allow you to store the data in a more compact form that saves memory, such as 2-byte integers, or 4-byte floats.

3. Object Size reduction with correct datatypes:

- Generally, the memory usage of the data frame can be reduced by converting them to correct datatypes. Almost all the datasets include object datatype which is generally in string format which is not memory efficient.
- When you consider the date, categorical features like region, city, place names these were in the string which takes more memory so if we convert these to respective data types like DateTime, categorical which makes memory usage reduced by more than 10 times as consumed before.

4. Use a Relational Database:

- Relational databases provide a standard way of storing and accessing very large datasets.
- Internally, the data is stored on disk can be progressively loaded in batches and can be queried using a standard query language (SQL).
- Free open source database tools like MySQL or Postgres can be used and most (all?) programming languages and many machine learning tools can connect directly to relational databases. You can also use a lightweight approach, such as SQLite.
- I have found this approach to be very effective in the past for very large tabular datasets.
- Again, you may need to use algorithms that can handle iterative learning.

5. Using Fast loading libraries like Vaex:

- Vaex is a high-performance Python library for lazy Out-of-Core DataFrames (similar to Pandas), to visualize and explore big tabular datasets.
- It calculates statistics such as mean, sum, count, standard deviation, etc, on an N-dimensional grid for more than a billion (10^9) samples/rows per second.
- Visualization is done using histograms, density plots, and 3d volume rendering, allowing interactive exploration of big data. Vaex uses memory mapping, zero memory copy policy, and lazy computations for best performance (no memory wasted).

Challenges Associated With LOOCV Technique

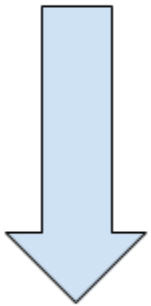
-Gauri Tyagi
A023119820028

What Is Resampling?

RESAMPLING

It is the process of repeatedly drawing samples or subsets from a training set and then refitting the particular model on each sample in order to gain information such as variability of fit.

1 2 3…….N elements



N/2 elements in training set	N/2 elements in validation set

What is Cross-validation?

Cross Validation

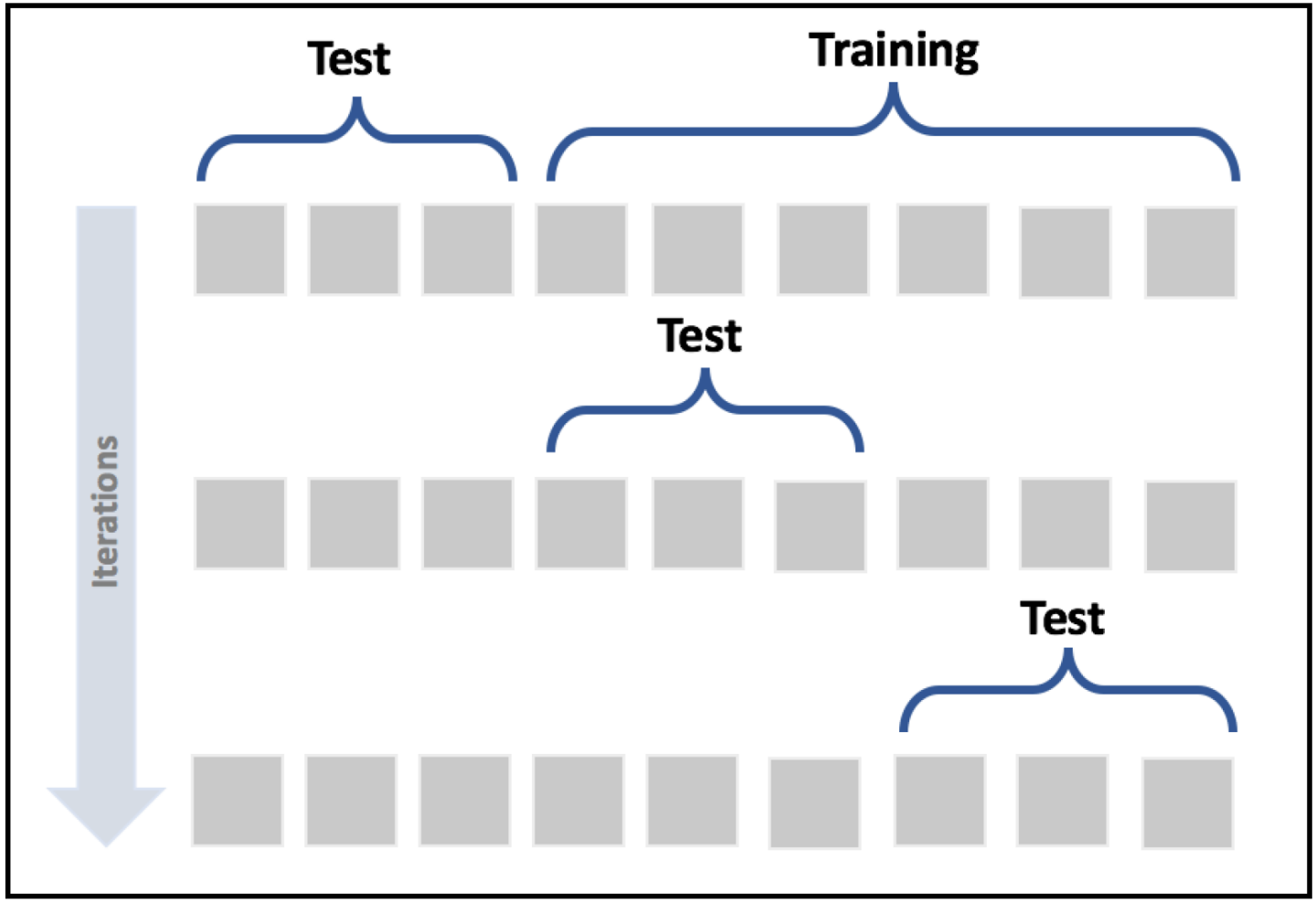
It is a technique for evaluating the performance of a machine learning model based on the resampling technique.

In this technique the original data set is randomly divided into two parts. One being the training dataset and the other being the ‘validation set’.

The model is fitted on the training data set and is then used to predict results on validation set.

The validation set acts a testing data set.


23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85 60 42 02
23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12
92 49 51 13 85 60 42 02
Training Set
Validation Set


Test
Training
Test
Test
Iterations

Leave One Out Cross Validation

LOOCV Technique

It is one of the many types of '**cross validation technique**'.

Here only ONE sample data is kept out as the 'validation data set'.

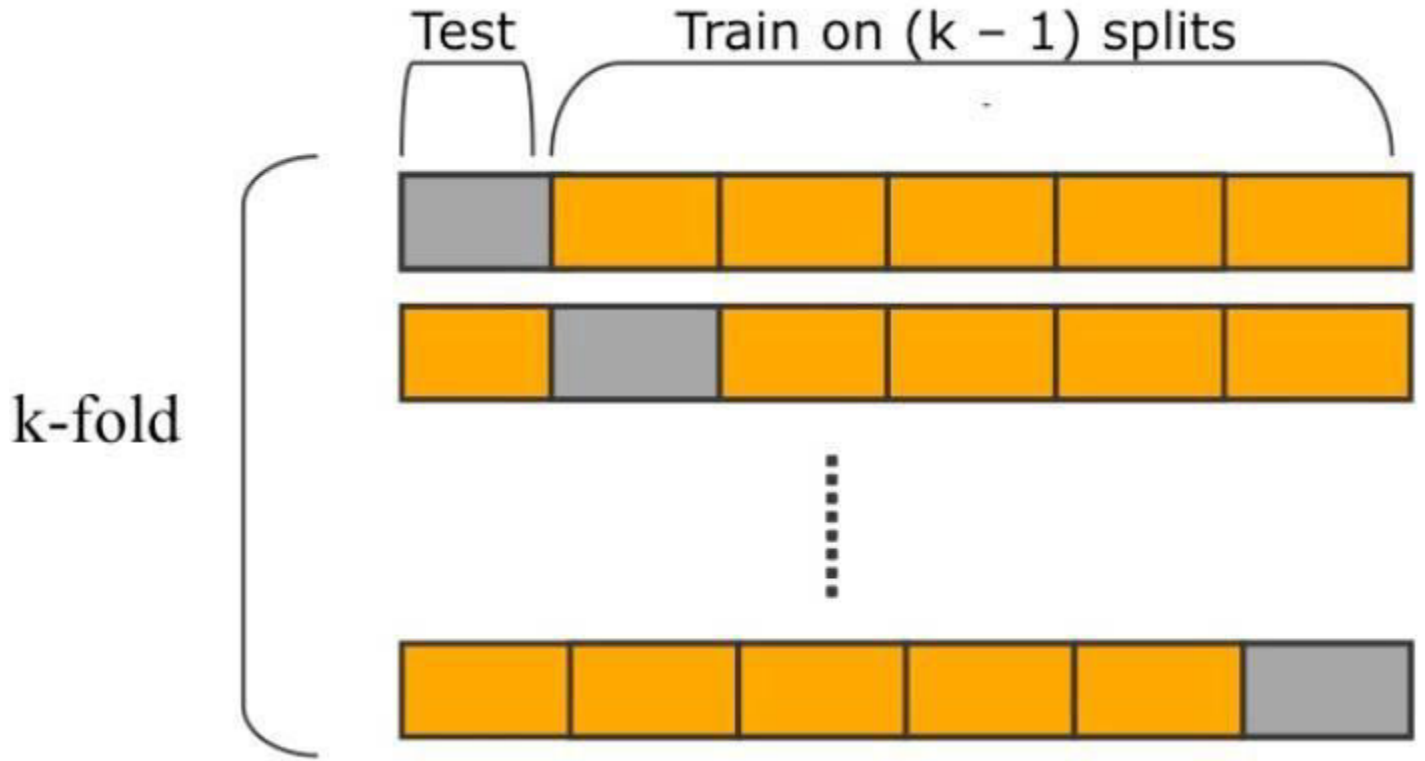
The model is evaluated or tested for each held out observation.

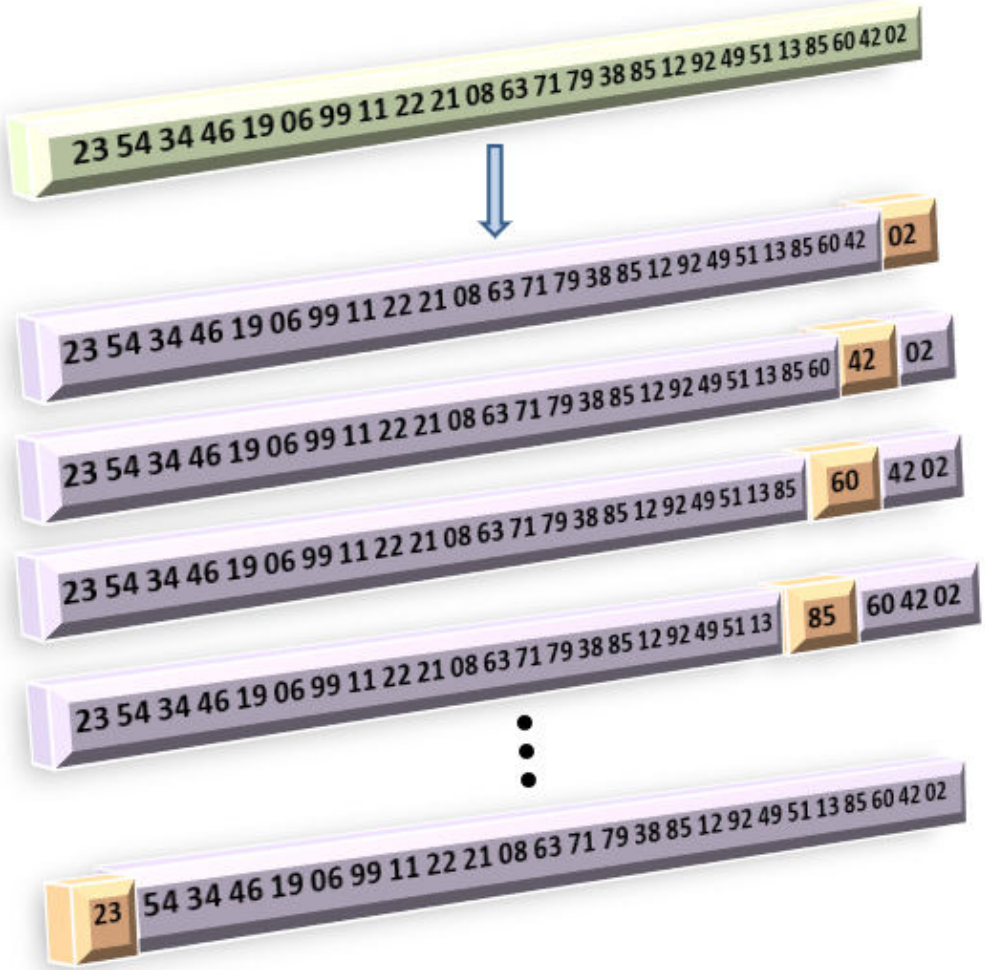
The final evaluation is then calculated by taking the mean of all individual evaluations.

At times it is also called as a special case of K-Fold Cross Validation Technique, where the value of K is the number of observations taken.

Every single point will be used in a validation set, we will create n models, for n-observations in the data.

Each point/sample is used once as a test set while the remaining data/samples form the training set.

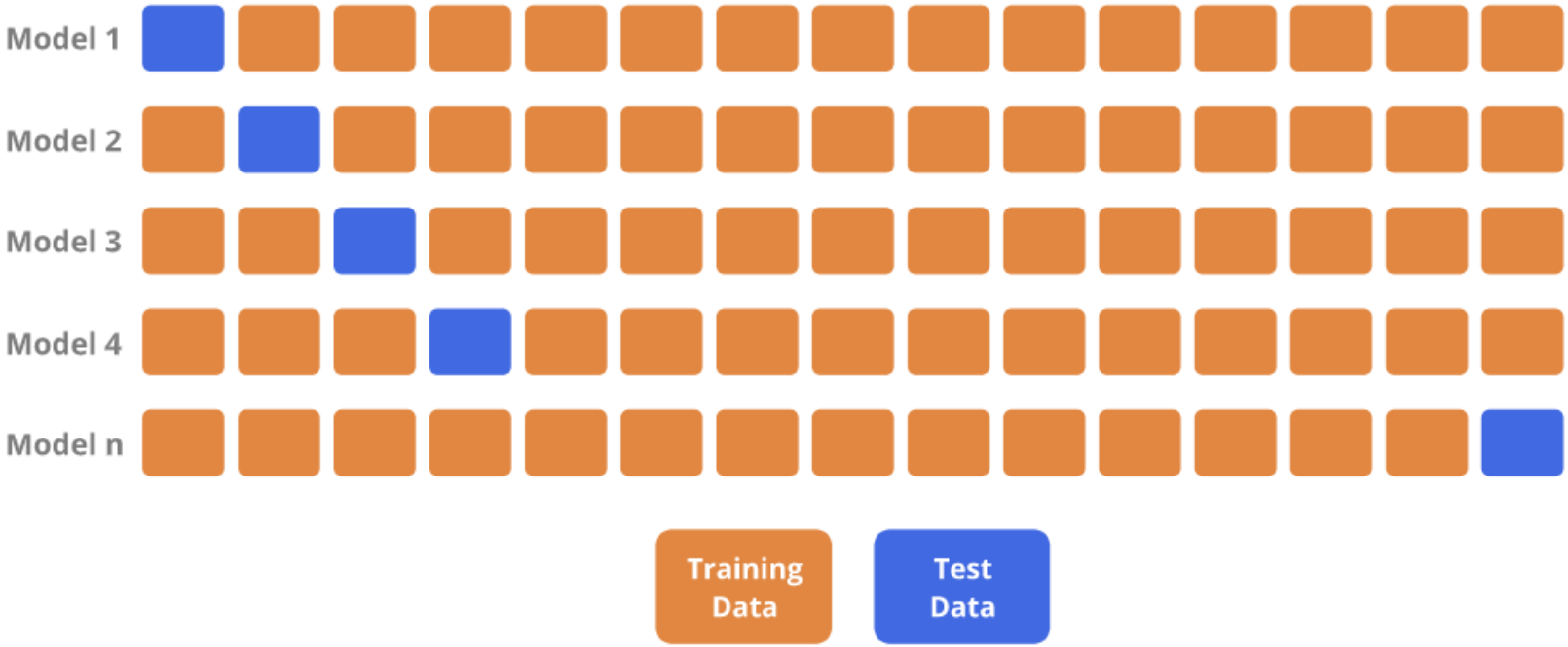

Test
Train on (k – 1) splits
k-fold


23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85 60 42 02
23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85 60 42
02
23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85 60
42
02
23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85
60
42 02
23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13
85
60 42 02
23 54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85 60 42 02
23
54 34 46 19 06 99 11 22 21 08 63 71 79 38 85 12 92 49 51 13 85 60 42 02

LOOCV Algorithm

1. Choose one sample from the dataset which will be the test set
2. The remaining n – 1 samples will be the training set
3. Train the model on the training set. On each iteration, a new model must be trained
4. Validate on the test set
5. Save the result of the validation
6. Repeat steps 1 – 5 n times as for n samples we have n different training and test sets
7. To get the final score average the results that you got on step 5.

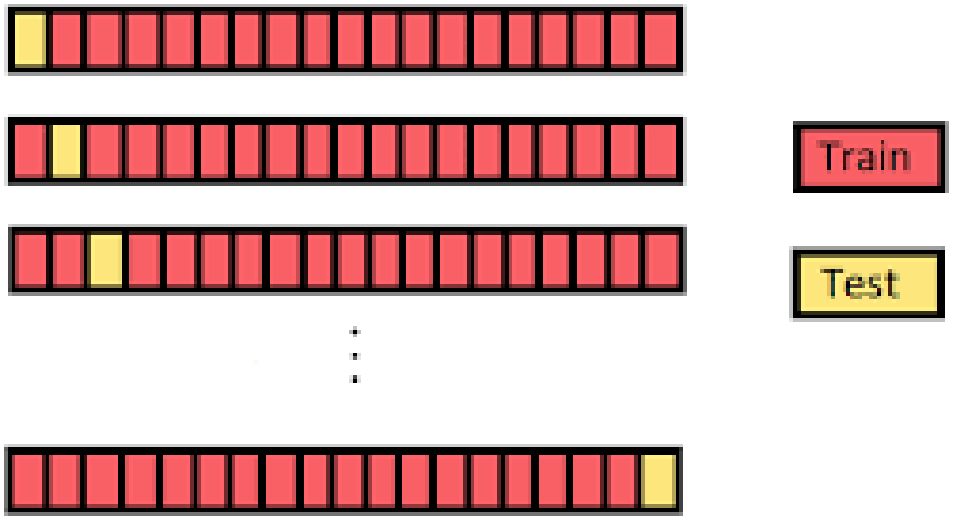
Leave-One-Out Cross Validation



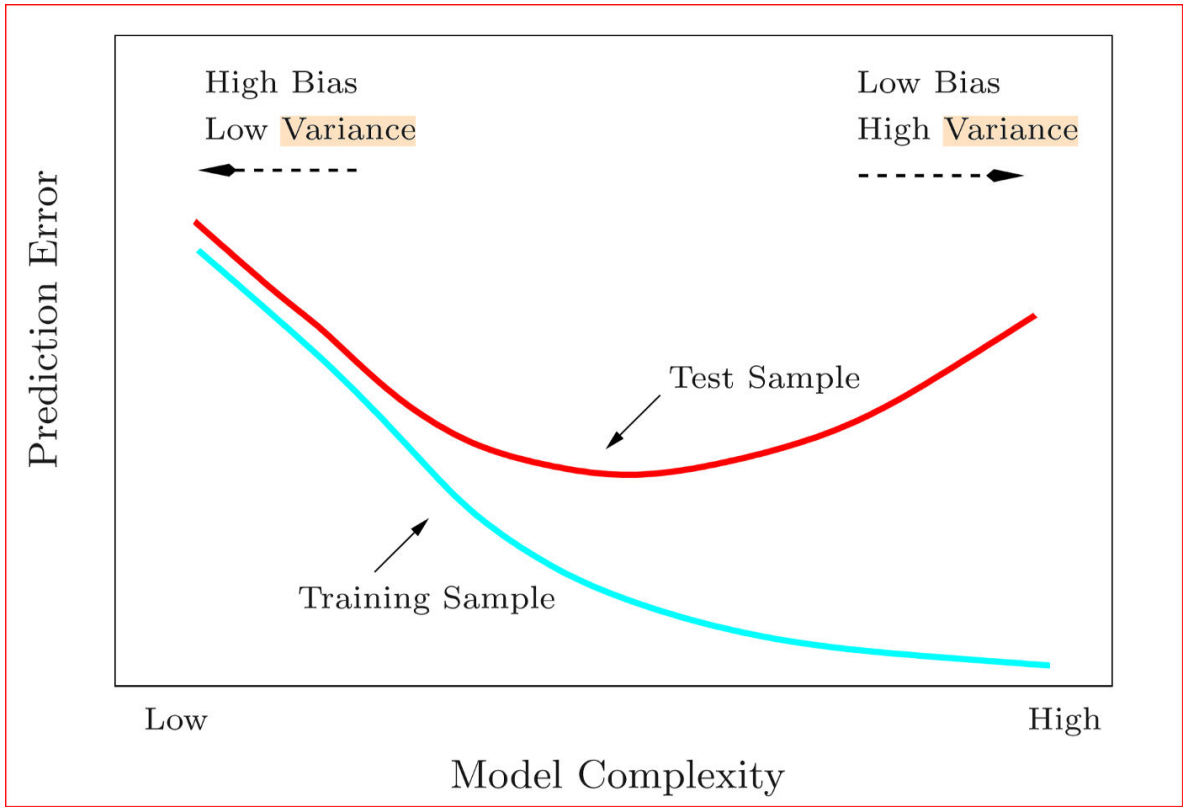
Challenges With LOOCV

It is computationally expensive if the size of data set is large.

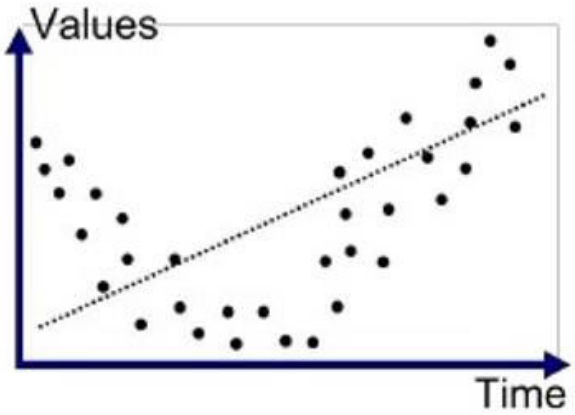
It is computationally expensive if the model takes substantial time to complete the learning process once.



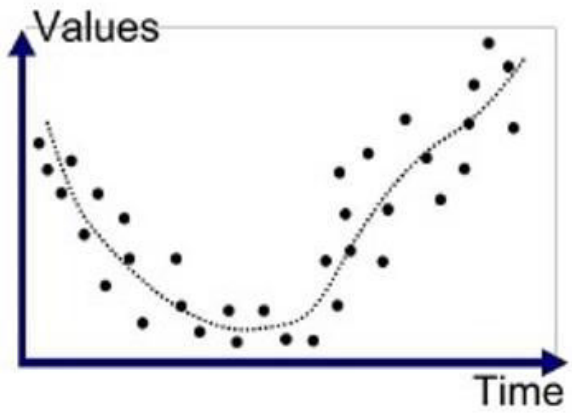
It can lead to high variance



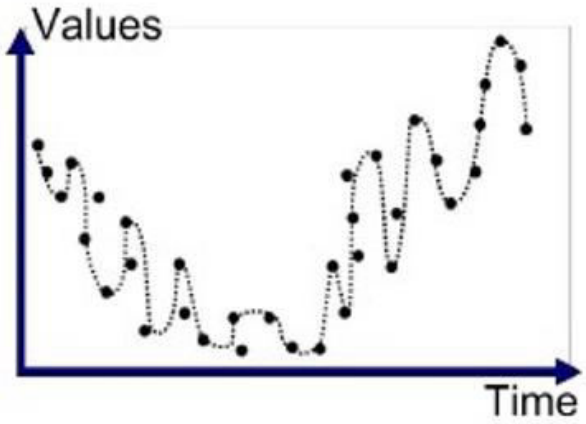
It can lead to overfitting of the model



Underfitted



Good Fit/Robust



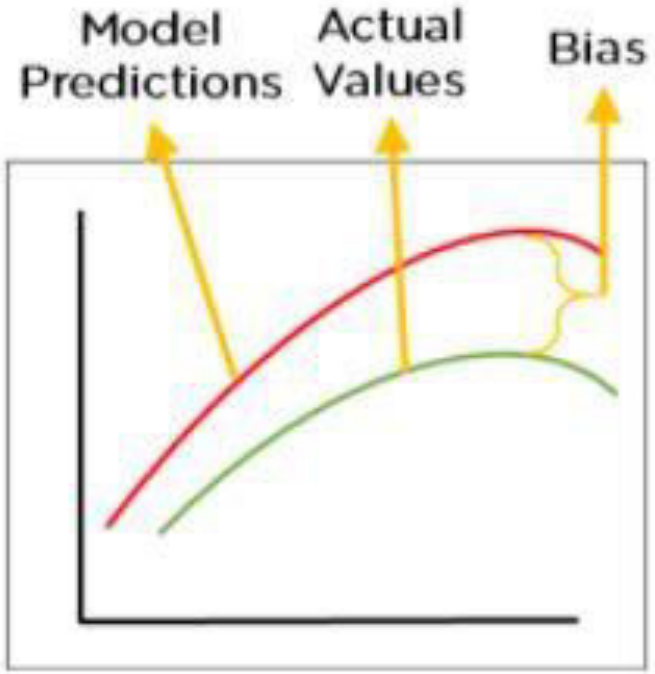
Overfitted

THANK YOU

When the Bias is high, assumptions made by our model are too basic.

The model can’t capture the important features of our data.

This means that our model hasn’t captured patterns in the training data and hence cannot perform well on the testing data too.



We can see that our model has learned extremely well for our training data, which has taught it to identify cats.

But when given new data, such as the picture of a fox, our model predicts it as a cat, as that is what it has learned.

This happens when the Variance is high, our model will capture all the features of the data given to it, including the noise, will tune itself to the data, and predict it very well but when given new data, it cannot predict on it as it is too specific to training data.