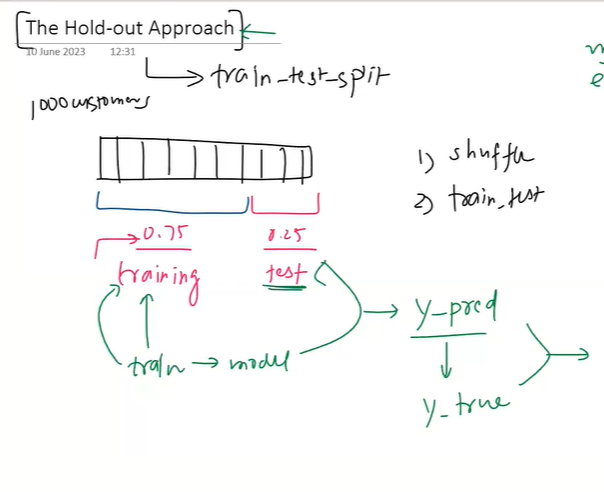
***Why to Learn Cross Validation***

*Cross Validation is atechnique to train a model & evaluate it.*

Abhitak Hum Hold Out Approach use krrhe theah



***Hold Out Approach***

(Train test split)

1k log ka data hai

Step 1. We shuffle data

Step 2. Splitting data

75 % to 80% data on training &

Remaining 25 to 20 % data on Testing set

Step 3. Train your Model on Training Data & Hide the test data

Step4. Run Model on test data fir humme Y\_pred milega we will compare & check the accuracy

It is a simplest approach we use to do Model Evaluation but it is not a only way Yeah Hold Out approach ko itna acha consider karte hie nahi

Ishse better Cross Validation hai

It is improved approach for Model Evaluation.

*Problem With Hold-Out- Approach*

Variability:

Random State change krke accuracy kahi bhi pochjayega

That’s why accuracy reliable nahi hai

1. Variability: The performance of the model can be very sensitive to how the data is divided

into training and testing sets. If the split is unfortunate, the training set may not be

representative of the overall distribution of data, or the test set might contain unusually

easy or difficult examples. This leads to high variance in the estimation of the model's

performance.

1. Data inefficiency: The holdout method only uses a portion of the data for training and a

different portion for testing. This means that the model doesn't get to learn from all

available data, which can be particularly problematic if the dataset is small.

1. Bias in performance estimation: If some classes or patterns are over- or under-represented

in the training set or the test set due to the random split, it can lead to a biased

performance estimation.

1. Less reliable for hyperparameter tuning: If the holdout method is used for hyperparameter

tuning, there's a risk of overfitting to the test set because information might leak from the

test set into the model. This means that the model's performance on the test set might be

overly optimistic and not representative of its performance on unseen data.

***Why is hold-out approach used then?***

1. **Simplicity:**

The holdout method is straightforward and easy to understand. You simply

divide your data into two sets: a training set and a test set. This simplicity makes it

appealing, especially for initial exploratory analysis or simple projects.

1. **Computational Efficiency:**

The holdout method is computationally less intensive than methods like k-fold cross-validation. In k-fold cross-validation, you need to train and test

your model k times, which can be computationally expensive, especially for large datasets

or complex models. With the holdout method, you only train the model once.

1. **Large Datasets:**

For very large datasets, even a small proportion of the data may be

sufficient to form a representative test set. In these cases, the holdout method can work

quite well.

***Cross Validation***

Cross Validation comes under concept of Resampling

Cross Validation is a Resampling technique

The idea of Cross Validation is to divide data into subsets or folds. The model is then trained

On some of the subsets & tested, on the remaining ones.

This process is repeated multiple times with different subset used for validation & training

Each time. The Results from each round are usually averaged to estimate model performance.

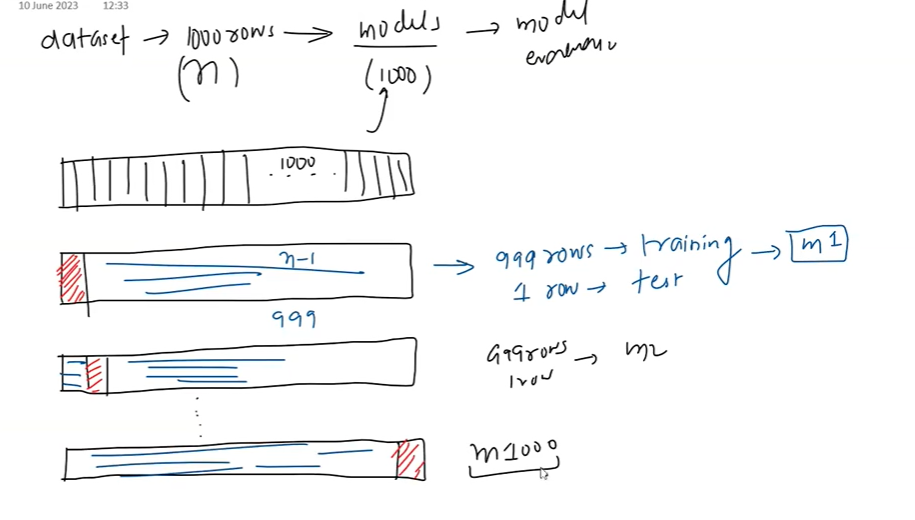
2 Famous Technique of Cross Validation

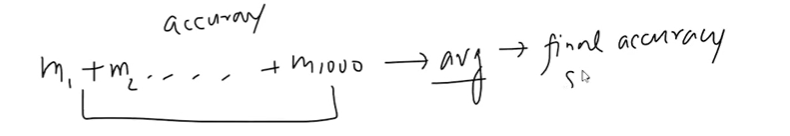
1. LOOCV (Leave One out CV)
2. K- Fold Cross Validation (isme bhi bhaut variations hote hai )

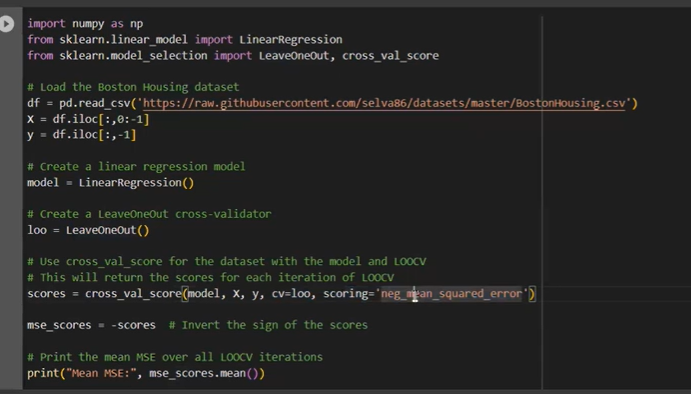
Stratified.

***Leave One Out Cross Validation (LOOCV)***

Dataset -> 1000 Rows -> models -> model evaluation







Yahape r2 Score nahi nikalskte

Advantages:

1. Use of Data: LOOCV uses almost all of the data for training, which can be

beneficial in situations where the dataset is small and every data point is

valuable.

1. Less Bias: Since each iteration of validation is performed on just one data

point, LOOCV is less biased than other methods, such as k-fold cross-

validation. The validation process is less dependent on the random

partitioning of data.

1. No Randomness: There's no randomness in the train/test split, so the

evaluation is stable, without variation in the results due to different random

splits.

*Disadvantages:*

1. Computational Expense: LOOCV requires fitting the model N times, which can be computationally expensive and time-consuming for large datasets.
2. High Variance: LOOCV can lead to higher variance in the model performance since the training sets in all iterations are very similar to each other.
3. Inappropriate Performance Metric: Performance metrics like R^2 are not

appropriate to be used with LOOCV as they are not defined when the

validation set only has one sample.

1. Not Ideal for Imbalanced Data: In classification problems, if you have

imbalanced classes, LOOCV may not provide a reliable estimate of model

performance because the single validation sample in each iteration may not

be representative of the overall class distribution.

***When to use:***

1. Small datasets: LOOCV is most beneficial when you have a limited amount of

data. With small datasets, you want to use as much data as possible for

training to get a reliable model, which is exactly what LOOCV offers by using

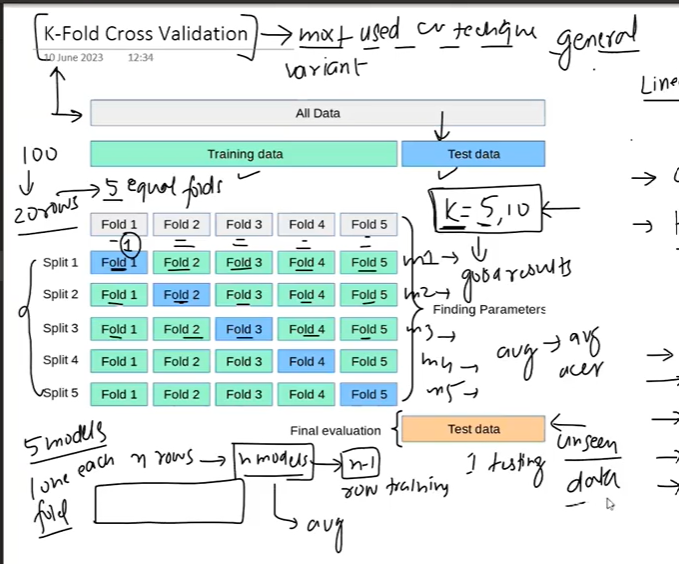
all but one data point for training.

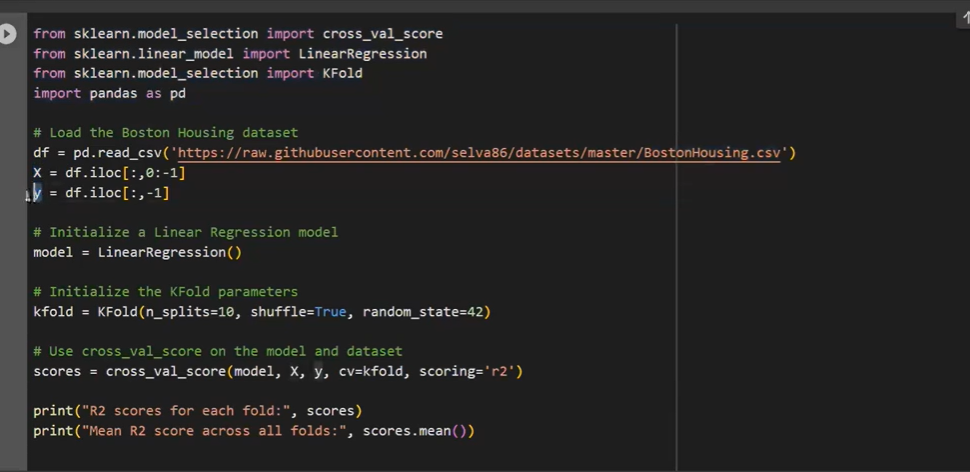
1. Balanced datasets: LOOCV might not perform well on imbalanced datasets, especially in classification problems, because the training set might end up missing some classes. Thus, it's more appropriate to use LOOCV when you have a balanced dataset.
2. Need for less biased performance estimate: Since LOOCV uses nearly all the data for training, it gives a less biased estimate of model performance compared to other methods like k-fold cross-validation.

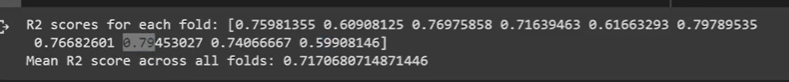
*K-Fold Cross Validation*

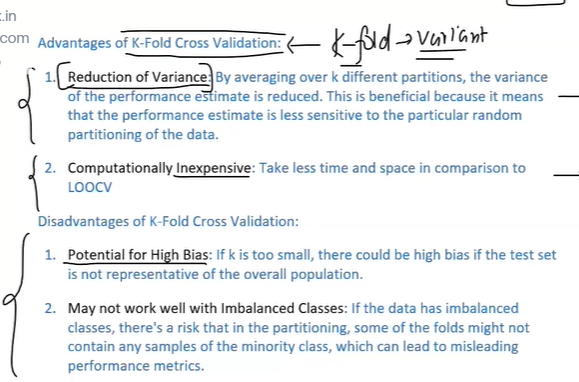
It is most used Cross Validation Technique

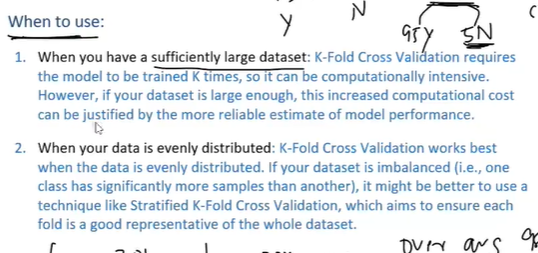
Isme K = 5,10 proven hai ki yeah ache results dete hie hai esa Papers mai Proven hai









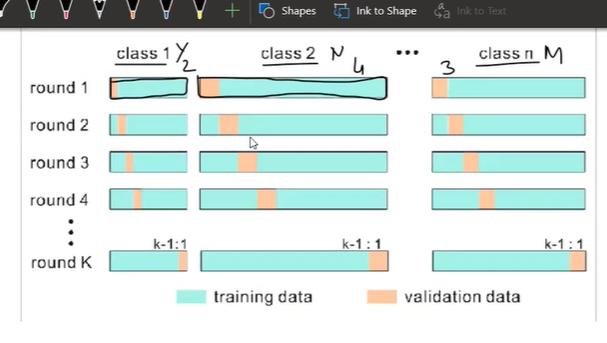


**Don’t use K-Fold On Imbalance Dataset uske jageh**

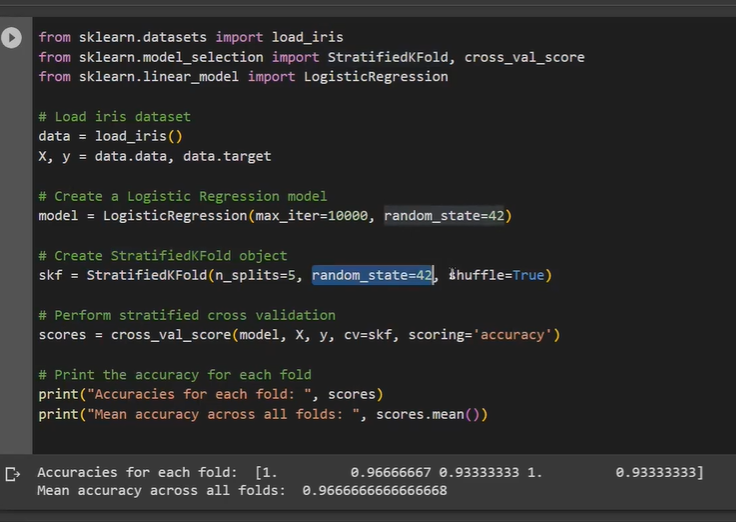
**Stratified K-Fold lagao**

***Stratified K-Fold Cross validation***

**As K-Fold ka sbhse bada Fault yeah tha ki Imbalance data mai Reliable Result nahi deta**

****

**In every fold har class ke values rhenge & with Equal Ratio in all folds , that’s why ishe Hum Imbalance data ke sath use krskte hai**

****