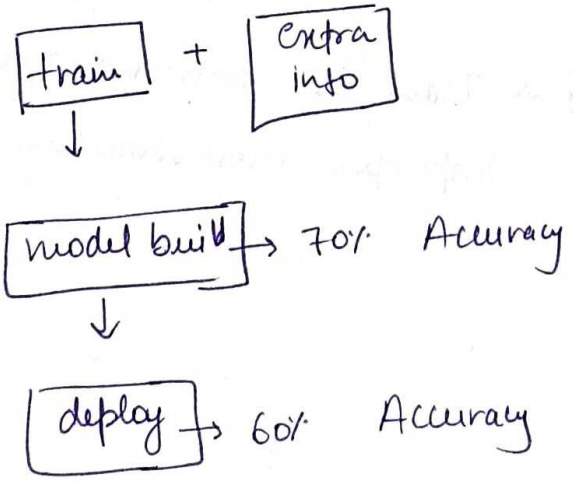
Data leakage

What is Data leakage?

Data leakage, in the content of machine learning and data Science, refers to a problem where information from outside the training dataset is used to create the model. This additional information can come in various forms, but the common characteristic is that it is information that the model wouldn't have access to when it's used for prediction in a real scenario.

This can lead to overly optimistic performer estimates during training and validation, as the model has access to extra information. However when the model is deployed in a production environment, that additional information is no longer available, and the performer of the model can drop significantly. This discrepancy is typically a result of mistakes in the experiment design.



Ways in which Data Leakage can occur

1. Target Leakage:

   Target leakage occur when your predictors (input) include data that will not be available at the time you make predictions.

2. Multicollinearity with target col (output)

Intuition Target Leakage

Value	Website	.....	reversed-transaction	fraud
500	-		1	1
400	-		0	0

This col is not available after deployment bcz reversed-transaction → Yes(1) if fraud → Yes(1).

when we trained data with reversed-transaction and accuracy → 90%. after deployment reversed-transaction col^n not use and accuracy is 80%.

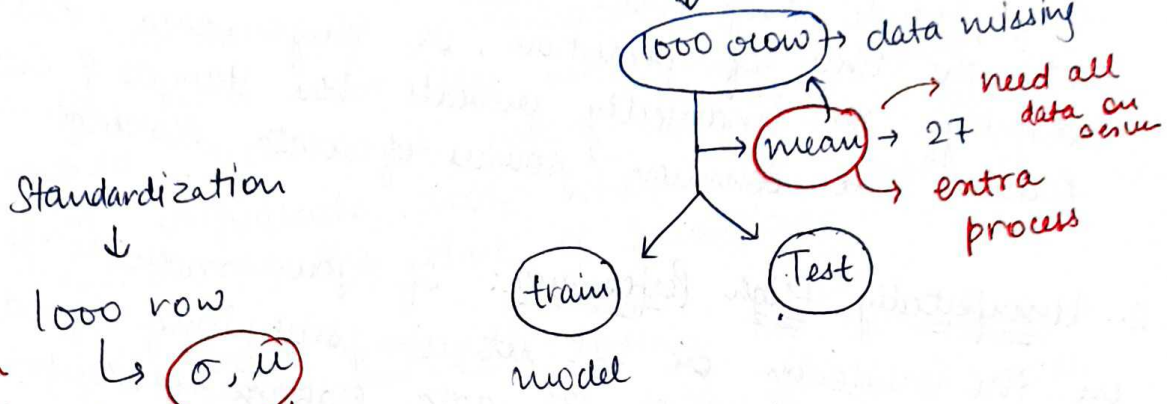
3. Duplicate Data

4. Preprocessing Leakage → Train Test contamination & Improper Cross Validation

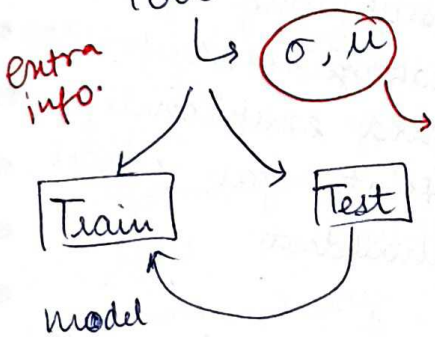
preprocessing → model

- missing
- scale
- Train

If data is missing



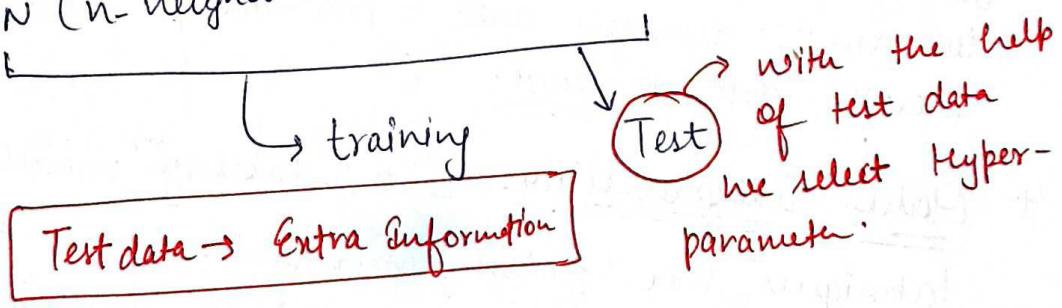
Standardization



σ, μ value extract from Test data too. but when unseen data occur we don't have test data

5. Hyperparameter running

KNN (n_neighbour = 7, metric = -)



How to detect ?

1. **Review Your Features**: Carefully review all the features being used to train your model. Do they include any data that wouldn't be available at the time of prediction, or any data that directly or indirectly reveals the target? Such features are common sources of data leakage.

2. **Unexpectedly High Performance**: If your model's performance on the validation or test set is surprising good, this could be a sign of data leakage. Most predictive modelling tasks are challenging, and exceptionally high performance could mean that your model has access to information it shouldn't.

3. **Inconsistent Performance Between Training and Unseen Data**: If your model performs significantly better on the training and validation data compared to new, unseen data, this might indicate that there's data leakage.

4. **Model Interpretability**: Interpretability models, or techniques like feature importance, can help understand what the model is learning. If the model is places too much importance on a feature that doesn't seem directly related to the output, it could be a sign of leakage.

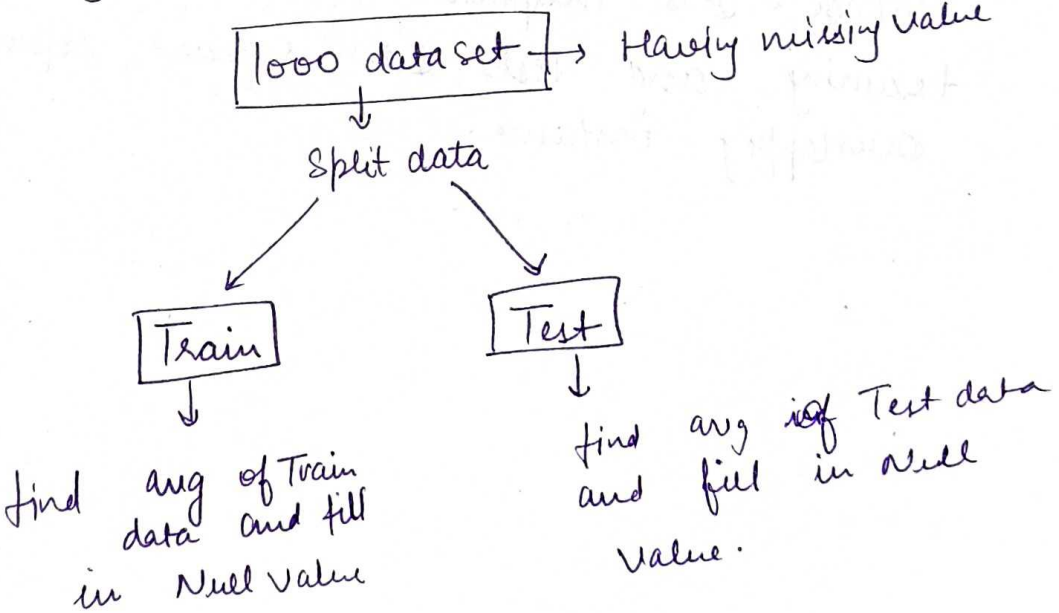
Check → Feature Importance → f_1, f_2

How to remove Data Leakage

1. **Understand the Data and the Task:** Before starting with any kind of data processing or modelling, It's important to understand the problem, the data, and how the data was collected. You should understand what each feature in your data represents and whether it would be available at the time of prediction.

2. **Careful Feature Selection:** Review all the feature used in your model. If any feature includes information that wouldn't be available at the time of prediction, or that directly or indirectly gives away the target variable, it should be removed or modified. eg:- Multicollinearity

3. **Proper Data Splitting:-** Always split your data into training, validation and tuting sets at an early stage of your pipeline, before doing any pre-processing or feature extraction.



4. Pre-processing inside the cross-validation loop:-

If you're using technique like cross-validation, make sure to do any pre-processing inside the cross-validation loop. This ensures that the pre-processing is done separately on each fold of the data, which prevents information from the validation set leaking into training set.

```
# Incorrect way
X_normalized = normalize(X)   # normalize the whole dataset
cross_val_score(model, X_normalized, Y, cv=5)

# Correct way
pipeline = make_pipeline(normalizer, model)
cross_val_score(pipeline, X, Y, cv=5)
```

6. Avoid Overlapping Data: If the same individual, or the same time periods, appear in both your training and test sets, this can cause data leakage. It is important to ensure that the training and tests sets represent seperate, non-overlapping instance.