***Data Leakage***

***The Problem***

***Production mai jbh Hum Model ko daalte hai tbh at that time vo alag hie accuracy dene lagta hai & normal training dataset pe banate waqt vo brbr hota hai ekdum esa kai baar hota hai***

***Then ese Problem agar kbhi bhi hota hai tou smjh jaana it’s the problem of***

***Data Leakage***

***What is Data Leakage?***

***Data Leakage in the context of Machine learning& Data science refers to a problem where information from outside 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 real-world scenario.***

***Data leakage in machine learning refers to the inadvertent incorporation of information from the test or evaluation dataset into the training dataset, leading to overly optimistic performance metrics and an inaccurate estimation of a model's true generalization capabilities. It is a critical issue that can severely impact the reliability and effectiveness of machine learning models.***

***Data leakage can occur in various ways, but the common thread is that it exposes the model to information that it would not have access to in real-world scenarios, leading to unrealistic performance during training and testing phases. Here are some common sources of data leakage:***

***1. \*\*Leakage through features\*\*: This occurs when features that are not available during prediction time or reflect future information are included in the training data. For example, if you're predicting customer churn and you include a feature like "total purchases made in the last month," this information would not be available for future predictions and will introduce leakage.***

***2. \*\*Leakage through target variables\*\*: Sometimes, the target variable (the variable you are trying to predict) can inadvertently be derived from other features present in the dataset. This leads to a strong relationship between the target and the features and can cause the model to "cheat" during training.***

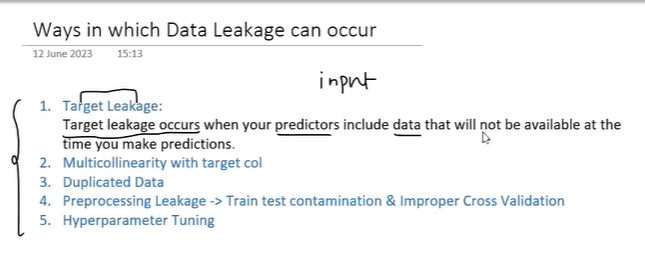
***3. \*\*Time-based leakage\*\*: In time-series data, it's essential to maintain the chronological order of events. Data leakage can occur if future data is used to predict past events, leading to unrealistically good performance.***

***4. \*\*Leakage through data preprocessing\*\*: Data preprocessing steps, such as scaling, normalization, or imputation, should only be applied to the training data and then transferred to the test data. If these steps are applied to the entire dataset before splitting, it can introduce leakage.***

***5. \*\*Leakage through cross-validation\*\*: Improper use of cross-validation can also cause leakage. For example, if the dataset is not properly shuffled before performing cross-validation, each fold may contain instances that are too similar or too different from one another, leading to biased evaluation results.***

***6. \*\*Leakage through data augmentation\*\*: Data augmentation techniques, such as adding noise or perturbations to the data, should only be applied to the training set. Applying them to the test set would introduce artificial patterns that the model hasn't encountered before.***

***Data leakage is a critical concern because it gives a false sense of model performance, making the model appear better than it actually is. To avoid data leakage, it's essential to maintain a clear distinction between training and test data and ensure that the training dataset represents the information available during real-world predictions accurately. Proper data validation, feature engineering, and appropriate cross-validation techniques are essential to prevent data leakage and build reliable machine learning models.***

******

***Eexample of Data Leakage***

Sure! Let's consider a credit card fraud detection model:

Suppose you are building a machine learning model to detect fraudulent credit card transactions. You have a historical dataset with information about previous transactions, including features like transaction amount, merchant category, location, time of day, etc. One of the target variables is "is\_fraud" that indicates whether a transaction is fraudulent or not.

Data Leakage Example:

- Now, imagine that in your dataset, there's a feature called "is\_fraudulent\_customer" that directly tells whether a customer has been involved in fraud in the past or not.

- If you include this "is\_fraudulent\_customer" feature during the training of your fraud detection model, it will have direct access to information about the customer's fraud history.

- The model may then learn that certain customers with a history of fraud are more likely to have fraudulent transactions in the future, which might be true for the training dataset.

- However, in real-world scenarios, the model won't have access to a customer's fraud history during the transaction, making the information useless for accurate predictions.

In this case, including the "is\_fraudulent\_customer" feature in the training data leads to data leakage, as it exposes the model to information it won't have during actual predictions. The model might perform well during training and testing because it's essentially "cheating" by using future information that it shouldn't have access to.

To prevent data leakage, you should avoid using features that directly reveal the target variable or any related information about it in the training dataset. Instead, focus on using only features that would be available during real-time predictions to build a more reliable fraud detection model.

*Multicolinearity With Target Columns*

*E.g:*

*Smartphone dataset hai jo vaha humm generally dekhe Ratings jinka High hota hai Unka utna Price High hota hai.*

*So yeah direct Multicolinearity ka scene hai yaha*

*But in Real World Humme jbh Price predict krna hoga at that time humare pass Ratings nahi hoga*

*So yaha data leakage problem aasakta hai*

*Hyperparameter Tunining*

*E.g Hyperparameter jo hote ML Models mai*

*Isme hum alag alag values try out krte hai & we see ki best result kisme aaraha hai.*

*Abh Problem kya hai isme ki*

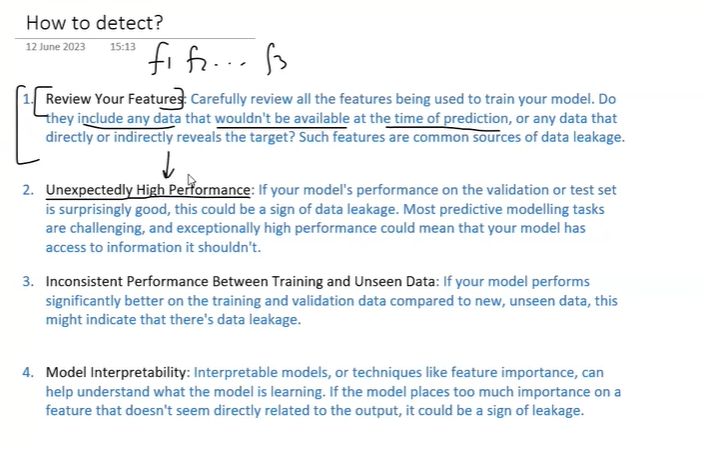
*Jbh Hum Hyperparameter tuning krte hai tou*

*Alag alag Values pe training krte hai & then Hum test set kesath Validate krte hai then*

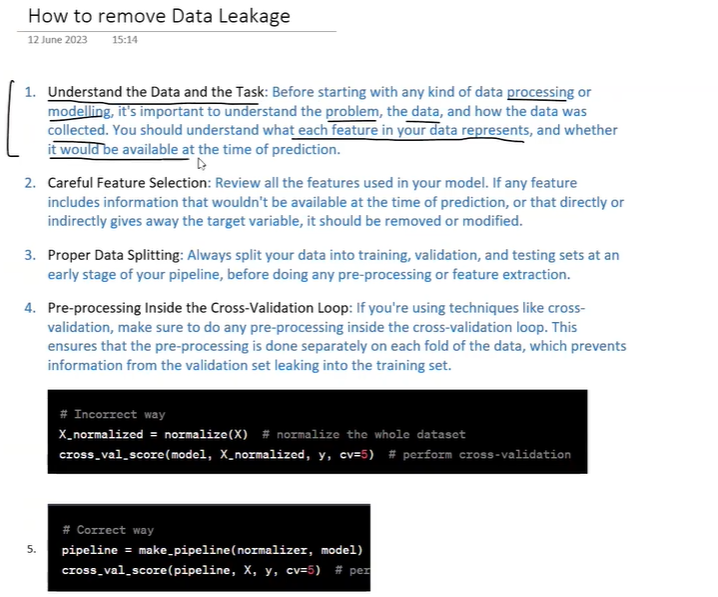
*Jo Test data hai it is acting as Extra Info*

*So yeah Problem hoti hai*

*How to Detect Data Leakage?*



*How to Remove Data Leakage*

**