Feature Engineering

- The process of creating new features or transforming existing features in your data set to improve model performance
- Important step in the machine learning pipeline

Why is Feature Engineering important?

- Can improve model accuracy and generalization
- Can help to extract meaningful information from raw data
- Can help to reduce the complexity of the model
- It divide into two techniques:
    - Feature Extraction
    - Feature Selection
    - Feature Transformation

Feature Extraction

- Creating new features from existing ones
- Example: extracting year from a date feature

Feature Selection

- Selecting the most important features for the model
- Example: using feature importance from a decision tree model to select the most important features

Feature Transformation

- Transform the values for a feature into more meaningful values.
- Ex: songs in seconds → songs in minutes (s → m)

PCA (Principal Component Analysis)

- Technique to reduce the dimensionality of the data set
- Can help to deal with the curse of dimensionality
- Example: reducing a high-dimensional data set to a lower-dimensional one for better visualization or modeling

Splitting the Data?

- Why Split the Data?
- Types of Splitting

Why Split the Data?

- To evaluate the model performance on new, unseen data
- To prevent overfitting and increase model generalization

Types of Splitting

- Training set: used to train the model
- Validation set: used to tune model hyperparameters
- Test set: used to evaluate the model performance

Let’s practice with some **CODE**

Tips & Tricks

Tips & Tricks

- Unbalanced Data
- Split the Data before Preprocessing
- Start Simple
- Make a Feature for Null Values
- Use ffill, or bfill in time series data
- Ordinal Encoding Sorting Issue

Unbalanced Data

- When one class in the data set has significantly fewer observations than the other class(es)
- Techniques:
    - Oversampling: increase the number of samples in the minority class
    - Undersampling: decrease the number of samples in the majority class

Split the Data Before Preprocessing

- To prevent data leakage: preprocessing steps (e.g. imputation) should be applied separately to each split

Start Simple

- Avoid overcomplicating the model or the preprocessing steps

- Sometimes, Simplicity works

Make a Feature for Null Values

- Create a new feature that indicates whether a feature had missing values or not

Filling Null Values

- Use forward fill (ffill) or backward fill (bfill) to fill missing values in time series data
- Use interpolation or mean/median/mode to fill missing values in other types of data

Ordinal Encoding Sorting Issue

- When encoding categorical features, make sure to assign numerical values that reflect the order of the categories
- Example: assigning 1 to "low", 2 to "medium", and 3 to "high"