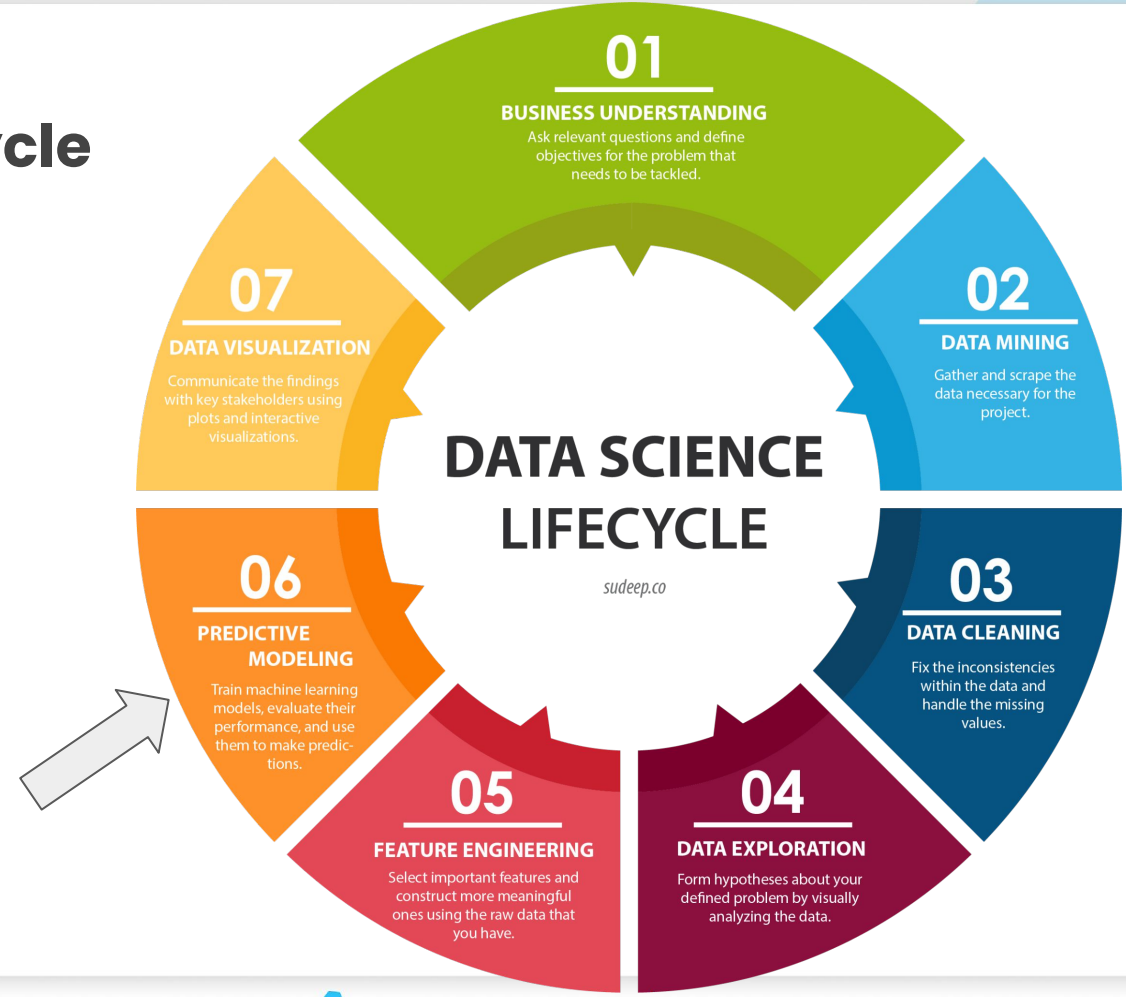
IRON HACK

Hyperparameter Tuning & Cross Validation

Week 7

Lifecycle

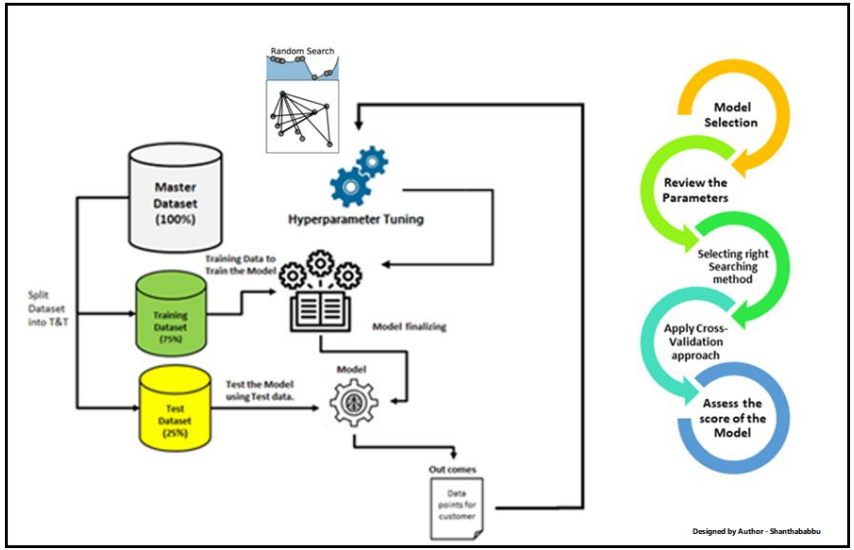

01
BUSINESS UNDERSTANDING
Ask relevant questions and define objectives for the problem that needs to be tackled.
02
DATA MINING
Gather and scrape the data necessary for the project.
03
DATA CLEANING
Fix the inconsistencies within the data and handle the missing values.
04
DATA EXPLORATION
Form hypotheses about your defined problem by visually analyzing the data.
05
FEATURE ENGINEERING
Select important features and construct more meaningful ones using the raw data that you have.
06
PREDICTIVE MODELING
Train machine learning models, evaluate their performance, and use them to make predictions.
07
DATA VISUALIZATION
Communicate the findings with key stakeholders using plots and interactive visualizations.
DATA SCIENCE LIFECYCLE
sudeep.co

Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the **optimal hyperparameters** for your model.

For example:

- In **KNN**, what is the ideal number of neighbors to consider?
- In a **Decision Tree**, what is the maximum depth?
- In **Random Forest**, what is the number of estimators?

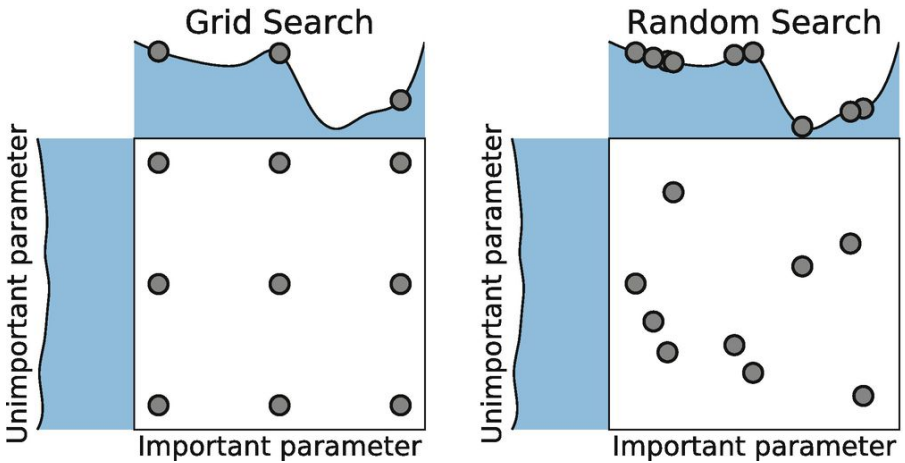


Hyperparameter Tuning

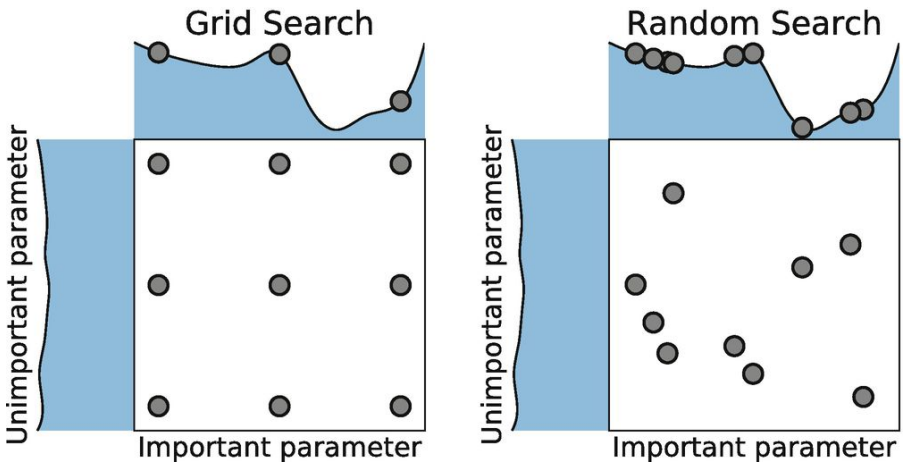
We are going to Grid Search and Random Search.

- **Grid Search -** we define a grid of hyperparameter values we want to try. Grid Search tries **all possible combinations.**

  It can be computationally expensive, especially when dealing with a large number of hyperparameters.



Hyperparameter Tuning



We are going to Grid Search and Random Search.

- **Random Search -** we define probability distributions for each hyperparameter, from which random values are sampled.
  It's up to the **researcher to set the maximum number of combinations.**

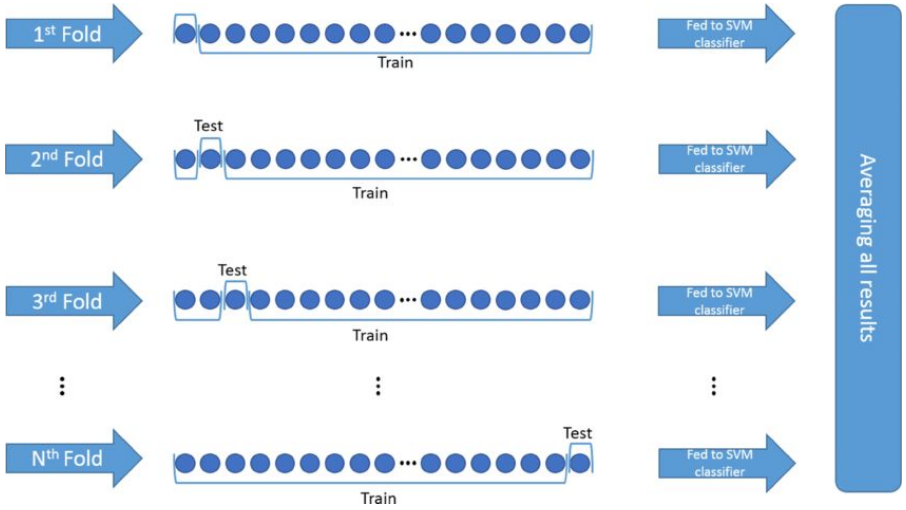
Hyperparameter Tuning

In both Grid and Random Search, there is no guarantee that we will discover the optimal values for the hyperparameters. Instead, we can only identify the best values among those we have tried.

Cross Validation

Cross Validation is a technique used to assess the performance of our model on different partitions of the data.

CV is the process of performing several different test-train splits, ensuring that we're not leaving meaningful data points in the test data.



Cross Validation

There are many types of CV:

- **KFold**: Splits data into k equally sized folds for cv.
- **GroupKFold:** Groups data based on unique groups to ensure no groups overlap between train and test sets in cv.
- **ShuffleSplit:** Randomly shuffles and splits data into train-test sets for multiple iterations.
- **StratifiedKFold:** Preserves class distribution in each fold for balanced cv.
- **GroupShuffleSplit:** Groups data and shuffles within groups for randomized train-test splits.
- **StratifiedShuffleSplit:** Randomly shuffles data while preserving class distribution in train-test splits.