1. Hold-out vs K-Fold
    a. Hold-out validation adalah metode validasi yang memisahkan dataset menjadi dua bagian yang tidak overlap. Dataset dibagi menjadi training set dan test set, dimana model hanya dilatih pada training set dan kemudian dievaluasi melalui test set. Proporsi pembagian training set dan test set biasanya mengikuti rule of thumb 80/20.
    b. K-Fold Cross validation adalah metode validasi yang membagi dataset menjadi K bagian yang ukurannya sama. Proses training kemudian diulang sebanyak K kali dan pada setiap putaran, satu fold digunakan sebagai test set dan sisanya digunakan sebagai training set.
2. Kapan Hold-out lebih baik dari K-Fold, dan sebaliknya
    a. Hold-out validation lebih baik ketika memiliki dataset yang sangat besar, karena dalam satu kali pembagian (misalnya 80/20) sudah cukup untuk menghasilkan training set dan test set yang representatif. Selain itu, menjalankan K-Fold pada dataset yang sangat besar membutuhkan computational resource yang besar.
    b. K-Fold cross validation lebih baik pada dataset yang relatif kecil / terbatas. K-Fold menggunakan seluruh data untuk pelatihan dan pengujian serta dilakukan secara berulang kali sehingga lebih andal dan tidak terlalu terpengaruh oleh randomness dari pembagian data.
3. Data leakage adalah kesalahan dalam pembuatan model dimana data dari luar (test set), tercampur di dalam data training set. Hal ini dapat menjadi noise pada model yang pada akhirnya akan mempengaruhi output dari model. Contohnya, apabila ada data dari test set yang tercampur di dalam proses training, hal ini dapat mengakibatkan overfitting pada model yang pada akhirnya membuat model memiliki performa baik pada data test, tetapi buruk dalam memprediksi data lainnya.
4. Dampak dari data leakage adalah overfitting pada model sehingga kinerja model terlihat sangat baik saat dievaluasi. Model akan terlihat sangat akurat pada test set, tetapi akan memiliki akurasi yang buruk ketika dihadapkan dengan data baru. Hal ini disebabkan karena model tidak benar-benar mempelajari pola umum dari data, melainkan mempelajari pola dari data yang akan diuji.
5. Solusi mengatasi data leakage
    a. Pisahkan data terlebih dahulu dengan menggunakan train_test_split sebelum melakukan scaling, imputation, atau feature engineering.

b. Lakukan transform pada objek preprocessor seperti Scaler dan Imputer secara terpisah.
c. Gunakan pipeline untuk memastikan urutan yang benar dan terstruktur.