a. Jelaskan apa yang dimaksud dengan *hold-out validation* dan *k-fold cross-validation*!

Hold-Out Validation adalah metode validasi model di mana dataset dibagi menjadi dua subset: subset pelatihan (training set) dan subset pengujian (testing set). Model dilatih pada subset pelatihan dan kemudian diuji pada subset pengujian. Biasanya, pembagian data dilakukan dalam proporsi tertentu, misalnya 80% untuk pelatihan dan 20% untuk pengujian.

K-Fold Cross-Validation adalah metode validasi model di mana dataset dibagi menjadi *k* bagian (folds) yang sama besar. Model dilatih dan diuji sebanyak *k* kali, di mana pada setiap iterasi, satu fold digunakan sebagai data pengujian dan *k-1* fold lainnya digunakan sebagai data pelatihan. Hasil akhirnya adalah rata-rata dari semua metrik evaluasi yang dihitung di setiap iterasi.

b. Jelaskan kondisi yang membuat *hold-out validation* lebih baik dibandingkan dengan *k-fold cross-validation*, dan jelaskan pula kasus sebaliknya!

Kondisi di mana Hold-Out Validation lebih baik:

- Dataset Sangat Besar

  Jika dataset yang tersedia sangat besar, maka menggunakan hold-out validation sudah cukup untuk mendapatkan gambaran kinerja model yang representatif. Pembagian sederhana menjadi training dan testing set bisa memberikan hasil yang memadai tanpa memerlukan pembagian lebih lanjut seperti pada k-fold cross-validation, sehingga menghemat waktu komputasi.

Kondisi di mana K-Fold Cross-Validation lebih baik:

- Dataset Terbatas atau Kecil

  K-fold cross-validation lebih baik digunakan karena model diuji pada berbagai subset data, sehingga memberikan estimasi kinerja yang lebih robust dan dapat mengurangi risiko hasil yang terlalu dipengaruhi oleh cara data dipecah.

- Mengukur Model Lebih Mendalam

  K-fold memberikan penilaian yang lebih mendalam terhadap model karena menggunakan semua data untuk pelatihan dan pengujian dalam berbagai kombinasi. Ini membantu dalam mencegah overfitting dan memberikan gambaran performa model yang lebih baik.

c. Apa yang dimaksud dengan *data leakage*?

Data Leakage adalah situasi di mana informasi dari data pengujian/*test set* bocor ke proses pelatihan model, sehingga model secara tidak sengaja "melihat" informasi yang seharusnya tidak diketahui selama pelatihan. Ini bisa terjadi jika variabel

yang seharusnya tidak tersedia selama pelatihan model dimasukkan dalam data pelatihan atau jika data dari masa depan digunakan secara tidak sengaja dalam pelatihan.

d. Bagaimana dampak *data leakage* terhadap kinerja dari model?

Data leakage dapat memberikan kesan palsu bahwa model memiliki kinerja yang sangat baik karena model "mempelajari" informasi yang sebenarnya tidak akan tersedia dalam situasi nyata. Ini menyebabkan model memiliki akurasi tinggi pada data pengujian, tetapi gagal ketika diterapkan pada data baru yang sebenarnya, karena prediksi model didasarkan pada informasi yang tidak tersedia dalam proses *training*.

e. Berikanlah solusi untuk mengatasi permasalahan *data leakage*!

Terdapat beberapa Solusi, di antaranya:

1. Memastikan Pemisahan yang Tepat

   Pastikan bahwa data yang digunakan untuk pelatihan dan pengujian benar-benar terpisah dan tidak ada overlap/irisan informasi. Proses ini bisa dilakukan dengan hati-hati saat melakukan pemisahan data.

2. Menghindari Penggunaan Data Masa Depan atau Data Test

   Pastikan bahwa hanya informasi yang digunakan pada saat pelatihan adalah yang tersedia saat ini dan tidak menggunakan data dari test set sama sekali. Misalnya, jika prediksi didasarkan pada data masa lalu, jangan gunakan data dari masa depan.

3. Feature Engineering dengan Cermat

   Saat melakukan feature engineering, pastikan bahwa fitur yang dibuat hanya menggunakan data dari train set atau data yang ada saat ini.

4. Penggunaan Pipeline

   Dalam banyak tools machine learning seperti scikit-learn, menggunakan pipeline membantu memastikan bahwa transformasi data dilakukan dengan cara yang benar, hanya pada data pelatihan, sehingga tidak ada informasi dari data pengujian yang bocor ke dalam model.

5. Review dan Uji Coba

   Secara berkala, lakukan review terhadap proses pengembangan model untuk memastikan tidak ada kesalahan yang menyebabkan data leakage. Juga, lakukan uji coba dengan data baru untuk memastikan model bisa melakukan generalisasi dengan baik tanpa adanya kebocoran data.