1. Jelaskan apa yang dimaksud dengan hold-out validation dan k-fold cross-validation!
Jawab:

- Hold-out validation adalah metode evaluasi model di mana dataset dibagi menjadi dua bagian utama: training set dan test set. Training set digunakan untuk melatih model, sementara test set digunakan untuk mengevaluasi performa model pada data yang belum pernah dilihat sebelumnya. Biasanya, dataset dibagi dengan rasio tertentu, misalnya 70% untuk training dan 30% untuk testing. Kelebihan utama dari hold-out validation adalah kesederhanaannya dan kecepatan eksekusinya, namun hasilnya bisa sangat bergantung pada bagaimana data dipecah.
- K-Fold Cross-Validation adalah metode evaluasi model yang lebih komprehensif di mana dataset dibagi menjadi *k* subset atau fold. Model dilatih *k* kali, setiap kali menggunakan *k-1* fold sebagai training set dan satu fold yang tersisa sebagai test set. Proses ini diulang sebanyak *k* kali sehingga setiap fold digunakan sebagai test set satu kali. Hasil evaluasi biasanya berupa rata-rata dari performa model pada semua fold, memberikan gambaran yang lebih robust tentang kinerja model. Kelebihan utamanya adalah memberikan evaluasi yang lebih stabil dan dapat diandalkan, terutama pada dataset kecil.

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

- Hold-Out Validation lebih baik jika:
    - Dataset sangat besar, karena lebih cepat dijalankan dan hasilnya sudah cukup stabil.
    - Waktu dan sumber daya terbatas, karena memungkinkan untuk evaluasi cepat tanpa memerlukan proses berulang seperti pada k-fold.
    - Pengembangan cepat, karena model harus dievaluasi dan iterasi dilakukan dengan cepat, maka hold-out validation bisa lebih praktis.
- K-Fold Cross-Validation lebih baik jika:
    - Dataset kecil, karena memberikan estimasi performa yang lebih andal karena semua data digunakan untuk pelatihan dan pengujian dalam berbagai konfigurasi.
    - Variabilitas tinggi, karena membantu dalam memberikan estimasi kinerja yang lebih konsisten dibandingkan hold-out.
    - Model tuning, untuk memastikan model tidak overfitting pada training set tertentu.

3. Apa yang dimaksud dengan data leakage?
Jawab:
Data leakage adalah situasi di mana informasi dari luar training dataset bocor ke dalam proses pelatihan, sehingga model mendapatkan akses ke informasi yang seharusnya tidak tersedia saat pelatihan. Data leakage dapat terjadi dalam berbagai bentuk, seperti saat data test secara tidak sengaja digunakan dalam pelatihan atau saat fitur yang digunakan dalam pelatihan memiliki informasi yang berasal dari masa depan (yang tidak akan tersedia pada saat prediksi

nyata). Data leakage mengakibatkan performa model yang terlalu optimis pada data pelatihan tetapi gagal pada data nyata.

4. Bagaimana dampak data leakage terhadap kinerja dari model?
Jawab:

- Overestimation of Performance: Model mungkin menunjukkan kinerja yang sangat tinggi pada data pelatihan karena telah melihat informasi yang seharusnya tidak ada sehingga memberi kesan palsu bahwa model sangat akurat.
- Poor Generalization: Model yang terpengaruh oleh data leakage cenderung tidak mampu menggeneralisasi pada data baru atau nyata, karena telah mempelajari pola yang tidak representatif.
- False Confidence: Data leakage menyebabkan model seolah-olah lebih baik dari yang sebenarnya, yang bisa berbahaya dalam aplikasi dunia nyata karena membuat keputusan yang salah berdasarkan model yang salah.

5. Berikanlah solusi untuk mengatasi permasalahan data leakage!
Jawab:

- Pastikan pemisahan yang ketat antara training, validation, dan test set. Test set harus benar-benar independen dan tidak digunakan sama sekali dalam proses pelatihan atau validasi model.
- Jika menggunakan fitur berbasis waktu (temporal features), pastikan bahwa data dari masa depan tidak bocor ke masa lalu dalam proses pelatihan.
- Tinjau dan cek fitur-fitur yang digunakan dalam model untuk memastikan bahwa tidak ada fitur yang secara tidak sengaja mengandung informasi dari target atau test set.
- Saat menggunakan cross-validation, pastikan bahwa data test tidak digunakan selama pelatihan, termasuk dalam preprocessing (misalnya, normalisasi atau imputasi) yang dilakukan setelah split data.
- Gunakan pipeline yang benar di mana semua langkah preprocessing dan pelatihan dilakukan dalam urutan yang benar, memastikan bahwa informasi dari test set tidak bocor kembali ke training set.