Modeling

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

Hold-out validation adalah teknik validasi model di mana dataset dibagi menjadi dua atau tiga bagian, yaitu training set, validation set (opsional), dan test set. Umumnya, data dibagi menjadi sekitar 70-80% untuk training set dan 20-30% untuk test set. Model dilatih menggunakan training set dan dievaluasi menggunakan test set. Teknik ini sederhana dan cepat karena model hanya perlu dilatih dan diuji satu kali.

K-fold cross-validation adalah teknik validasi yang membagi dataset menjadi k bagian (folds) yang hampir sama ukurannya. Model dilatih dan diuji k kali, di mana setiap kali satu fold digunakan sebagai test set dan sisa k-1 folds digunakan sebagai training set. Hasil evaluasi diperoleh dengan menghitung rata-rata performa model dari k kali pengujian. Teknik ini memberikan estimasi yang lebih akurat terhadap kinerja model karena menggunakan seluruh data untuk pelatihan dan pengujian.

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

Hold-out validation lebih baik ketika:

- **Simplicity and Speed**: Hold-out validation lebih sederhana dan lebih cepat dibandingkan k-fold cross-validation. Ketika sumber daya komputasi terbatas atau model membutuhkan waktu pelatihan yang sangat lama, hold-out validation mungkin menjadi pilihan yang lebih praktis. Metode ini hanya melibatkan satu kali pelatihan dan evaluasi model, sehingga lebih efisien dalam hal waktu dan sumber daya.
- **Large Datasets**: Pada dataset yang sangat besar, perbedaan antara hasil hold-out validation dan k-fold cross-validation mungkin tidak signifikan. Dalam kasus ini, hold-out validation bisa cukup akurat dan efisien karena dataset yang besar sudah memberikan sampel yang cukup representatif dalam pembagian data pelatihan dan pengujian.
- **Simplicity in Initial Experiments**: Untuk eksperimen awal atau pengujian cepat dari model yang baru dibangun, hold-out validation memberikan cara cepat untuk mendapatkan gambaran kasar tentang performa model tanpa memerlukan perhitungan yang intensif dari k-fold cross-validation.

K-fold cross-validation lebih baik ketika:

- **Small to Medium Datasets**: Pada dataset kecil hingga menengah, k-fold cross-validation memberikan estimasi performa yang lebih stabil

dan andal. Karena setiap data point digunakan untuk pelatihan dan pengujian, hasil dari k-fold cross-validation lebih representatif dari kemampuan model untuk generalisasi. Hal ini membantu mengurangi variabilitas yang disebabkan oleh pembagian data acak yang mungkin tidak representatif dalam hold-out validation.

- **Minimizing Variability**: K-fold cross-validation mengurangi variabilitas yang mungkin terjadi dari pembagian data tunggal dalam hold-out validation. K-fold cross-validation menggunakan setiap subset data untuk pengujian sekali, sehingga memberikan gambaran yang lebih robust tentang performa model yang dapat diharapkan secara keseluruhan.
- **Assessing Model Stability**: Untuk model yang sensitif terhadap fluktuasi dalam data pelatihan, k-fold cross-validation membantu mengevaluasi stabilitas model dengan memberikan hasil dari beberapa pembagian data. Hal ini berguna untuk model yang perlu divalidasi secara menyeluruh untuk memastikan performa yang konsisten di berbagai subset data.

c. Apa yang dimaksud dengan *data leakage*?

Data leakage terjadi ketika informasi dari luar training set (termasuk test set atau data baru) secara tidak sengaja digunakan dalam proses pelatihan model, sehingga model belajar informasi yang seharusnya tidak diketahui saat pelatihan. Data leakage ini dapat terjadi melalui berbagai cara, seperti fitur yang mengandung informasi masa depan, data preprocessing yang salah (misalnya normalisasi yang dilakukan sebelum membagi data), atau kebocoran label.

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

Data leakage dapat menyebabkan optimisme palsu dalam evaluasi kinerja model di mana pengukuran kinerja model menjadi tidak akurat. Evaluasi yang dilakukan dengan data yang telah bocor mungkin menunjukkan hasil yang lebih baik daripada performa model di data yang benar-benar tidak terlihat sebelumnya. Hal ini membuat hasil evaluasi tidak bisa diandalkan untuk menentukan sejauh mana model akan berfungsi pada data baru dan nyata. Data leakage sering kali menyebabkan overfitting di mana model mungkin terlalu menyesuaikan diri dengan data pelatihan tersebut, sehingga performanya dalam situasi nyata atau data baru bisa sangat buruk.

Data leakage juga dapat menyebabkan kesalahan dalam penyesuaian model. Jika model mendapatkan akses ke data yang tidak seharusnya, proses tuning dan pengaturan hiperparameter bisa saja informasi yang tidak

representatif dari data yang sebenarnya akan digunakan dalam model. Hal tersebut dapat mengarah pada keputusan yang tidak optimal dalam memilih parameter model atau algoritma.

Terakhir, data leakage membuat kurangnya generalisasi di mana model yang terkena kebocoran data mungkin bekerja dengan sangat baik pada data yang sudah bocor, tetapi kinerjanya bisa menurun drastis ketika diterapkan pada data yang benar-benar baru dan tidak terpapar.

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

- Pemisahan yang Tepat antara Data Pelatihan dan Pengujian
  - Stratified Sampling:

    Pastikan bahwa data pelatihan dan pengujian dipisahkan dengan benar sejak awal, tanpa adanya informasi yang bocor di antara keduanya. Gunakan teknik seperti stratified sampling untuk menjaga distribusi target yang seimbang antara set pelatihan dan pengujian.
  - Hold-out dan Cross-validation:

    Gunakan metode validasi seperti hold-out validation atau k-fold cross-validation dengan benar untuk memastikan bahwa evaluasi model dilakukan hanya pada data yang belum pernah dilihat oleh model selama pelatihan.
- Penerapan Proses Data yang Konsisten
  - Pra-pemrosesan Terpisah:

    Lakukan pra-pemrosesan data (seperti normalisasi, standaridasi, atau imputasi missing values) secara terpisah untuk data pelatihan dan pengujian. Jangan melakukan pra-pemrosesan pada seluruh dataset sebelum membaginya menjadi data pelatihan dan pengujian.
  - Pipeline Machine Learning:

    Gunakan pipeline untuk memastikan bahwa seluruh langkah pra-pemrosesan, pemilihan fitur, dan pemodelan diterapkan secara konsisten, menghindari kebocoran dari data pengujian ke dalam data pelatihan.
- Feature Engineering yang Hati-hati
  - Pemisahan Data Sebelum Feature Engineering:

    Pastikan bahwa setiap feature engineering dilakukan hanya pada data pelatihan dan tidak menggunakan informasi dari data pengujian. Misalnya, jika membuat fitur berdasarkan data historis, pastikan tidak ada informasi dari masa depan yang bocor ke dalam data pelatihan.

    - Time-series Data Handling:

      Dalam konteks data time-series, pastikan bahwa fitur yang dibuat tidak menggunakan informasi dari periode waktu yang lebih baru yang tidak tersedia selama pelatihan model.
- Monitoring dan Validasi yang Ketat
    - Regular Monitoring:

      Lakukan monitoring yang ketat terhadap proses pembelajaran model, terutama ketika melakukan hyperparameter tuning atau feature selection, untuk memastikan bahwa tidak ada kebocoran data yang tidak disengaja.
    - Validasi Tambahan:

      Setelah model dilatih dan diuji, lakukan validasi tambahan dengan menggunakan dataset baru atau menggunakan metode seperti time-based split (jika applicable) untuk memastikan bahwa model benar-benar dapat generalisasi dengan baik.
- Menghindari Overlapping Data
    - Avoid Data Overlap:

      Jika data dikumpulkan dari sumber yang berbeda, pastikan bahwa tidak ada overlap antara data pelatihan dan data pengujian. Misalnya, pastikan tidak ada individu atau entitas yang muncul baik di data pelatihan maupun di data pengujian.
    - Unique Identifiers Check:

      Gunakan pengidentifikasi unik untuk memastikan bahwa entitas yang sama tidak secara tidak sengaja muncul di set pelatihan dan pengujian.
- Training dan Awareness

  Perlu adanya pemahaman untuk mengenali tanda-tanda dan potensi data leakage. Kesadaran akan pentingnya pemisahan data dan proses yang benar merupakan langkah pencegahan yang penting.