a. Hold-out validation dan k-fold cross-validation

Hold-out validation adalah metode evaluasi model yang membagi dataset menjadi dua bagian, yaitu data latih dan data uji. Umumnya 70-80% data digunakan untuk pelatihan dan sisanya untuk pengujian. Metode ini sederhana dan cepat, namun hasilnya dapat bervariasi tergantung pada pembagian data.

K-fold cross-validation membagi dataset menjadi k bagian. Model dilatih menggunakan k-1 bagian dan diuji pada satu bagian yang tersisa. Proses ini diulang k kali, dengan setiap bagian berperan sebagai data uji sekali. Hasil akhir adalah rata-rata dari semua iterasi. Metode ini memberikan estimasi performa yang lebih stabil.

b. Kondisi yang membuat salah satu metode lebih baik

Hold-out validation lebih baik digunakan ketika:

- Dataset yang tersedia sangat besar
- Komputasi k-fold cross-validation akan terlalu mahal dalam hal waktu dan sumber daya
- Waktu pelatihan terbatas
- Membagi data menjadi dua bagian besar sudah cukup mewakili distribusi data sebenarnya

K-fold cross-validation lebih baik digunakan ketika:

- Dataset yang tersedia relatif kecil
- Kita ingin mendapatkan estimasi kinerja yang lebih stabil dan mengurangi bias
- Variasi dalam data tinggi
- Perlu mengurangi risiko overfitting atau underfitting
- Kita ingin memastikan bahwa model tidak mengandalkan subset data tertentu
- Dibutuhkan generalisasi yang lebih baik

c. Data leakage

Data leakage terjadi ketika informasi dari luar dataset pelatihan tidak sengaja masuk ke dalam proses pelatihan model. Akibatnya, model memiliki "bocoran" informasi tentang data uji. Contohnya adalah ketika terdapat fitur yang secara langsung atau tidak langsung mengandung informasi tentang target label yang ingin diprediksi.

d. Dampak data leakage

Data leakage dapat menyebabkan hasil evaluasi model yang terlalu optimis dan tidak realistis. Model mungkin menunjukkan performa yang sangat baik pada data uji, namun ini disebabkan oleh penggunaan informasi yang seharusnya tidak tersedia saat pelatihan. Akibatnya, ketika

model digunakan pada data baru, kinerjanya dapat menurun signifikan karena ketidakmampuan untuk menggeneralisasi dengan baik.

e. Solusi mengatasi data leakage

1. Memisahkan data dengan tepat, memastikan tidak ada tumpang tindih antara data pelatihan dan pengujian.
2. Melakukan feature engineering dengan hati-hati, memastikan fitur yang digunakan hanya berasal dari data pelatihan.
3. Menerapkan teknik cross-validation yang tepat untuk mencegah kebocoran informasi.
4. Melakukan review dan validasi secara berkala terhadap proses persiapan data dan feature engineering.
5. Menguji model dengan data yang benar-benar baru dan belum pernah digunakan dalam pelatihan.