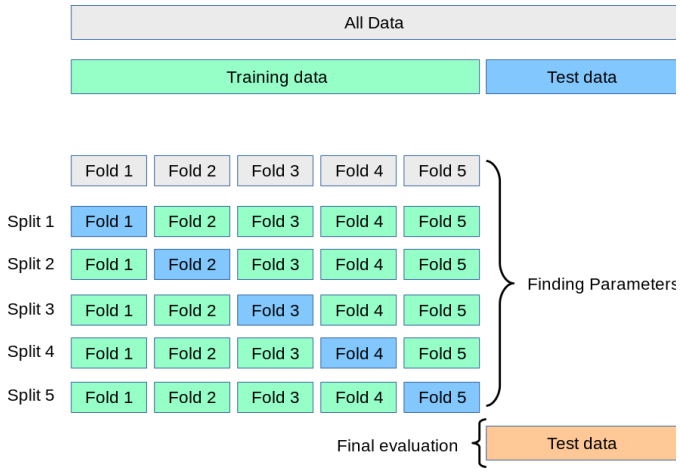
Soal a

Hold-out validation vs k-fold cross-validation

Hold-out validation adalah proses validasi model dengan melakukan train-test split terhadap dataset, melakukan training model menggunakan data train, lalu mengukur performa model dengan melakukan prediksi terhadap data test dan membandingkan hasilnya dengan ground truth. Train-test split adalah proses membagi dataset menjadi data train dan data test dengan perbandingan jumlah data yang umumnya 80-20. Data train dan test ini pun dipisah lagi menjadi X_train, y_train, X_test, dan y_test, di mana X adalah fitur-fitur dari tuple data dan y adalah label dari target prediksi. Model menggunakan X_train dan y_train untuk melatih dirinya dengan cara mencocokkan–menggunakan algoritma yang diimplementasikan pada mode–tuple pada X_train dengan label pada y_train yang berkorespondensi dengan tuple tersebut. Setelah dilatih, model diuji dengan memprediksi target berdasarkan X_test. Hasil prediksi dibandingkan dengan ground truth yakni y_train dan performa diukur menggunakan metrik evaluasi seperti accuracy, precision, recall, f1-score, ROC-AUC, dll.

Dibandingkan hold-out validation yang melakukan train-test split sekali, **k-fold cross-validation** membagi dataset menjadi k bagian (fold) yang sama besar lalu melakukan evaluasi sebanyak k kali dengan menggunakan sebuah fold sebagai data test dan sisanya sebagai data train pada setiap iterasi evaluasi. Metode evaluasi ini memastikan bahwa setiap bagian dari dataset dapat digunakan untuk menjadi bagian dari data test agar dapat menghasilkan evaluasi yang lebih menyeluruh dan mencegah overfit.



Sumber: scikit-learn

Soal b

Kondisi yang membuat hold-out validation lebih baik

Hold-out validation lebih cocok digunakan ketika menggunakan dataset yang sangat besar atau ketika sumber daya komputasi yang digunakan terbatas. Dalam kondisi ini, menggunakan k-fold cross validation akan sangat memakan waktu karena ia melakukan training dan prediksi sebanyak k kali, yang mana semakin besar nilai k, semakin banyak iterasi yang dilakukan. Ketika menggunakan dataset yang cukup besar, hanya melakukan train-test split sekali mungkin lebih praktis dan dapat memberikan hasil yang sudah cukup menggambarkan

performa model daripada harus mengorbankan waktu yang banyak untuk menggunakan k-fold cross validation.

Kondisi yang membuat k-fold cross validation lebih baik
Metode evaluasi ini lebih cocok digunakan ketika ukuran dataset cukup kecil. Dataset yang kecil dapat menyebabkan hasil evaluasi gagal untuk mendeskripsikan performa model secara menyeluruh apabila hanya menggunakan sebagian dari dataset sebagai data test karena sampel data ini mungkin tidak merepresentasikan pola-pola penting pada dataset secara keseluruhan.

Soal c dan d
Data leakage adalah fenomena ketika data dari luar data train digunakan untuk melatih model namun data eksternal ini tidak tersedia pada proses prediksi menggunakan model. Jika terjadi demikian, model akan memberikan performa yang sangat baik saat testing dan validasi, namun kinerjanya menjadi tidak akurat ketika digunakan saat deployment.
Terdapat 2 jenis data leakage, yakni target leakage dan train-test contamination. Target leakage terjadi ketika model dilatih dengan data yang tidak akan ada pada tahap prediksi. Contohnya, pada model untuk memprediksi pemalsuan kartu kredit, informasi mengenai terjadi chargeback atau tidak biasanya muncul setelah kepalsuan tersebut terdeteksi sehingga fitur ini tidak bisa digunakan untuk melatih model. Apabila fitur chargeback digunakan untuk melatih model, yang mana fitur ini memiliki korelasi yang sangat kuat terhadap target, model akan menunjukkan performa yang sangat baik saat evaluasi. Namun, saat deployment, karena informasi chargeback itu tidak hadir, model akan mengalami kesulitan memprediksi pemalsuan kredit. Jenis data leakage berikutnya adalah train-test contamination yang terjadi ketika informasi mengenai data test bocor ke data train akibat proses splitting atau preprocessing yang tidak benar. Salah satu penyebab kontaminasi ini adalah preprocessing seperti scaling atau standardisasi yang dilakukan sebelum splitting. Proses ini secara tidak langsung memasukkan informasi mengenai data test sehingga model seakan-akan sudah mengetahui informasi mengenai data test saat evaluasi dan akan menunjukkan kinerja yang sangat baik pada tahap, membuatnya tidak mampu dalam melakukan generalisasi terhadap data yang baru dan tidak pernah ia lihat.

Soal e
Data leakage dapat dicegah dengan melakukan preprocessing dan data splitting yang baik dan benar. Data splitting seharusnya dilakukan sebelum preprocessing agar dapat menghindari data train dan test diproses secara bersamaan dan menyebabkan train-test contamination. Feature engineering dan selection juga harus dilakukan dengan hati-hati agar fitur yang secara logika bisnis hanya muncul setelah prediksi dilakukan atau fitur yang berhubungan langsung dengan target tidak menimbulkan target leakage.