Laporan Prediksi "classe" Berdasarkan Data Latihan

1. Pemahaman Data dan Tujuan

Dataset ini berisi data sensor dari berbagai bagian tubuh (belt, arm, dumbbell, forearm) yang dikumpulkan dari pengguna yang melakukan latihan. Tujuan kita adalah untuk membangun model klasifikasi yang dapat memprediksi variabel target "classe" (kemungkinan jenis gerakan atau kualitas eksekusi) berdasarkan variabel-variabel sensor lainnya. Variabel "classe" adalah variabel kategorikal yang akan menjadi target prediksi kita.

2. Pemilihan Model

Mengingat sifat tugas ini adalah klasifikasi (memprediksi kategori), beberapa algoritma yang relevan untuk dicoba meliputi:

- **Random Forest:** Algoritma ensemble yang kuat dan seringkali memberikan kinerja yang baik pada berbagai jenis data. Relatif tahan terhadap *overfitting* dan dapat menangani sejumlah besar fitur.
- **Gradient Boosting (misalnya, XGBoost, LightGBM):** Algoritma ensemble lain yang sangat efektif dan seringkali mencapai akurasi tertinggi dalam tugas klasifikasi. Namun, perlu hati-hati dalam penyesuaian *hyperparameter* untuk menghindari *overfitting*.
- **Support Vector Machines (SVM):** Efektif dalam ruang fitur berdimensi tinggi dan dapat menangani kasus di mana jumlah fitur lebih besar dari jumlah sampel.
- **K-Nearest Neighbors (KNN):** Algoritma berbasis instans yang sederhana dan intuitif. Kinerjanya sangat bergantung pada pemilihan *k* dan metrik jarak.

Mengingat kompleksitas data dengan banyak fitur dan kemungkinan adanya hubungan non-linear, saya akan memulai dengan **Random Forest** sebagai model dasar. Random Forest cenderung memberikan kinerja yang baik secara *out-of-the-box* dan relatif mudah diinterpretasikan. Jika hasilnya kurang memuaskan, saya akan mempertimbangkan **Gradient Boosting** untuk potensi akurasi yang lebih tinggi, dengan catatan perlu lebih berhati-hati dalam proses *tuning*.

Alasan Pemilihan Awal Random Forest:

- Kuat dan akurat.
- Dapat menangani data dengan banyak fitur.
- Relatif tahan terhadap *overfitting*.
- Memberikan informasi pentingnya fitur.

3. Pemrosesan Awal Data (Pre-processing)

Sebelum melatih model, langkah-langkah pemrosesan awal data penting untuk memastikan kualitas data dan kinerja model yang optimal. Langkah-langkah yang akan saya lakukan meliputi:

- **Penanganan Nilai yang Hilang (Missing Values):** Dataset yang diberikan memiliki banyak nilai NA. Strategi penanganan nilai yang hilang dapat berupa:
    - **Penghapusan Kolom dengan Proporsi Nilai Hilang Tinggi:** Jika suatu fitur memiliki terlalu banyak nilai yang hilang, mungkin lebih baik untuk menghapusnya karena tidak memberikan banyak informasi yang berguna. Batas proporsi nilai hilang untuk penghapusan perlu ditentukan (misalnya, di atas 50%).
    - **Imputasi:** Mengisi nilai yang hilang dengan nilai statistik seperti mean, median, atau modus dari kolom tersebut. Untuk data sensor, median mungkin lebih robust terhadap *outlier*.
    - **Imputasi Berdasarkan Fitur Lain:** Dalam kasus yang lebih kompleks, nilai yang hilang dapat diprediksi menggunakan fitur-fitur lain.
    - **Mengingat banyaknya nilai NA pada beberapa kolom dalam data contoh, saya akan cenderung untuk menghapus kolom-kolom dengan proporsi nilai hilang yang sangat tinggi untuk menghindari bias yang signifikan akibat imputasi yang tidak akurat.**
- **Penghapusan Fitur yang Tidak Relevan:** Kolom seperti "", "user_name", "raw_timestamp_part_1", "raw_timestamp_part_2", "cvtd_timestamp", dan "problem_id" kemungkinan tidak memberikan informasi prediktif langsung tentang "classe" (kecuali mungkin ada pola implisit terkait pengguna, tetapi untuk menjaga fokus pada data sensor, saya akan menghapusnya). "new_window" dan "num_window" mungkin perlu dianalisis lebih lanjut, tetapi untuk awal, saya akan mempertimbangkan untuk menghapusnya kecuali ada justifikasi yang kuat untuk mempertahankannya.
- **Penanganan Fitur Kategorikal (jika ada setelah penghapusan):** Jika ada fitur kategorikal yang tersisa setelah penghapusan, mereka perlu diubah menjadi format numerik yang dapat dipahami oleh sebagian besar algoritma *machine learning* (misalnya, menggunakan *one-hot encoding*). Dalam data contoh, sebagian besar fitur tampak numerik.
- **Penskalaan Fitur (Feature Scaling):** Beberapa algoritma (seperti SVM dan KNN) sensitif terhadap skala fitur. Penskalaan fitur (misalnya, menggunakan *StandardScaler* atau *MinMaxScaler*) dapat membantu meningkatkan kinerja model. Untuk Random Forest dan Gradient Boosting, penskalaan biasanya tidak terlalu krusial.

4. Penggunaan Validasi Silang (Cross-Validation)

Validasi silang adalah teknik penting untuk mengevaluasi kinerja model secara lebih robust dan mendapatkan perkiraan yang lebih baik tentang kesalahan di luar sampel (generalization error). Saya akan menggunakan **k-fold cross-validation**.

- **Proses k-Fold:** Dataset latihan akan dibagi menjadi *k* bagian (lipatan atau *folds*) yang sama besar. Untuk setiap iterasi validasi, satu lipatan akan digunakan sebagai set validasi, dan *k-1* lipatan lainnya akan digunakan sebagai set pelatihan. Model akan

dilatih pada set pelatihan dan dievaluasi pada set validasi. Proses ini diulang *k* kali, dengan setiap lipatan digunakan sebagai set validasi sekali.

- **Metrik Evaluasi:** Metrik evaluasi yang relevan untuk klasifikasi meliputi:
    - **Akurasi (Accuracy):** Proporsi prediksi yang benar.
    - **Presisi (Precision):** Proporsi prediksi positif yang sebenarnya positif.
    - **Recall (Sensitivity atau True Positive Rate):** Proporsi aktual positif yang diprediksi dengan benar.
    - **F1-score:** Rata-rata harmonik presisi dan recall. Ini berguna ketika ada ketidakseimbangan kelas.
    - **Matriks Konfusi (Confusion Matrix):** Tabel yang menunjukkan jumlah prediksi benar dan salah untuk setiap kelas.
- **Pemilihan *k*:** Nilai *k* yang umum digunakan adalah 5 atau 10. Saya akan memulai dengan **k=5**. Ini memberikan keseimbangan antara bias dan varians dalam perkiraan kinerja model.
- **Tujuan Validasi Silang:**
    - Mendapatkan perkiraan yang lebih andal tentang kinerja model di data yang belum dilihat.
    - Membantu dalam pemilihan model dan *hyperparameter* yang tepat.
    - Mengidentifikasi potensi masalah seperti *overfitting* (kinerja baik pada data latihan tetapi buruk pada data validasi).

5. Pemikiran tentang Kesalahan di Luar Sampel yang Diharapkan

Kesalahan di luar sampel (generalization error) adalah kinerja model pada data baru yang belum pernah dilihat selama pelatihan. Beberapa faktor yang dapat mempengaruhi kesalahan di luar sampel yang diharapkan meliputi:

- **Kompleksitas Model:** Model yang terlalu kompleks (*high variance*) dapat *overfit* pada data latihan dan berkinerja buruk pada data baru. Model yang terlalu sederhana (*high bias*) mungkin tidak dapat menangkap pola yang kompleks dalam data.
- **Kualitas dan Representasi Data Latihan:** Jika data latihan tidak representatif dari data sebenarnya yang akan diprediksi, kinerja model di luar sampel akan buruk.
- **Jumlah Data Latihan:** Semakin banyak data latihan yang relevan dan berkualitas, semakin baik model dapat belajar dan melakukan generalisasi.
- **Noise dalam Data:** Data yang berisik (kesalahan pengukuran, *outlier* yang tidak informatif) dapat mempersulit model untuk mempelajari pola yang sebenarnya.
- **Ketidakseimbangan Kelas:** Jika jumlah sampel untuk setiap kelas dalam variabel target "classe" sangat tidak seimbang, model mungkin cenderung memprediksi kelas mayoritas dan berkinerja buruk pada kelas minoritas.

Perkiraan Kesalahan di Luar Sampel:

Dengan menggunakan validasi silang, kita akan mendapatkan perkiraan kinerja model pada data yang belum dilihat. Rata-rata skor validasi silang (misalnya, akurasi rata-rata dari 5 lipatan) akan menjadi perkiraan kesalahan di luar sampel yang diharapkan.

- Jika model (misalnya, Random Forest awal) memberikan akurasi validasi silang yang tinggi (misalnya, di atas 90%), kita dapat berharap kesalahan di luar sampel relatif rendah (di bawah 10%).
- Jika akurasi validasi silang lebih rendah, kesalahan di luar sampel yang diharapkan akan lebih tinggi.
- Perbedaan yang signifikan antara kinerja pada data latihan dan kinerja validasi silang dapat mengindikasikan *overfitting*.

6. Pemilihan Fitur (Feature Selection)

Setelah pemrosesan awal, kita mungkin memiliki banyak fitur. Pemilihan fitur dapat membantu:

- Menyederhanakan model.
- Mengurangi *overfitting*.
- Meningkatkan kinerja (dalam beberapa kasus).
- Meningkatkan interpretasi model.

Beberapa metode pemilihan fitur yang dapat dipertimbangkan:

- **Metode Berbasis Pohon (Tree-based methods):** Random Forest dan Gradient Boosting dapat memberikan skor pentingnya fitur, yang dapat digunakan untuk memilih fitur-fitur terpenting.
- **Metode Statistik:** Uji statistik (misalnya, ANOVA untuk fitur numerik dan variabel kategorikal) dapat digunakan untuk menilai hubungan antara fitur dan variabel target.
- **Pemilihan Fitur Berbasis Model (Model-based feature selection):** Model lain (misalnya, Logistic Regression dengan L1 regularization) dapat digunakan untuk memilih fitur.
- **Pemilihan Fitur Iteratif:** Metode seperti *Recursive Feature Elimination* (RFE) dapat digunakan untuk memilih subset fitur terbaik.

Mengingat sifat eksploratif awal, saya akan cenderung untuk melatih model pertama dengan semua fitur yang tersisa setelah penanganan nilai hilang dan penghapusan fitur yang jelas tidak relevan. Kemudian, berdasarkan pentingnya fitur yang diberikan oleh Random Forest, saya akan mempertimbangkan untuk mengurangi jumlah fitur jika diperlukan untuk meningkatkan kinerja atau mengurangi kompleksitas model.

7. Pelatihan Model dan Prediksi

Setelah pemrosesan awal data dan (mungkin) pemilihan fitur, model Random Forest (atau model terpilih lainnya) akan dilatih menggunakan seluruh data latihan. Kemudian, model yang terlatih ini akan digunakan untuk memprediksi variabel "classe" untuk 20 kasus uji yang diberikan.

Kesimpulan Sementara:

Strategi awal ini melibatkan penggunaan Random Forest sebagai model klasifikasi, penanganan nilai hilang dengan penghapusan kolom dengan proporsi hilang tinggi dan baris yang tersisa, penghapusan fitur yang tidak relevan, penskalaan fitur, dan evaluasi model menggunakan 5-fold cross-validation untuk memperkirakan kesalahan di luar sampel. Hasil validasi silang akan memberikan indikasi awal tentang seberapa baik model diharapkan dapat melakukan generalisasi pada data baru. Langkah selanjutnya adalah menganalisis hasil validasi silang, mempertimbangkan algoritma lain (seperti Gradient Boosting), melakukan *tuning hyperparameter*, dan menganalisis pentingnya fitur untuk potensi peningkatan model.