

Feature Selection untuk Prediksi Stok Obat Harian

Pendekatan Raw Features dan Analisis Konsensus

Kelompok 16:

- Muhammad Yusuf - 122140193
- Cornelius Linux - 122140079
- Chandra Budi Wijaya - 122140093

Pertanyaan Studi Kasus UTS: "Bagaimana kamu menentukan fitur mana yang relevan untuk model prediksi stok obat harian?"

Presentasi ini menjelaskan metodologi komprehensif untuk memilih fitur terbaik dari data transaksional farmasi mentah, menggunakan enam teknik feature selection yang berbeda untuk mengidentifikasi prediktor stok yang paling signifikan.

Data dan Metodologi Penelitian

Karakteristik Dataset

Penelitian ini menganalisis data transaksi farmasi dari sumber yang autentik dan representatif. Dataset terdiri dari 359 produk unik dengan informasi lengkap tentang pergerakan stok harian. Kami berfokus pada transaksi terakhir dari setiap produk untuk menangkap tren dan pola terkini dalam dinamika inventaris obat.

Target Variabel: Stok_Aktual (stok real di gudang pada momen transaksi terakhir).

Fitur Kandidat (9 Fitur Mentah)

- Qty_Masuk**
Kuantitas penerimaan stok
- Nilai_Masuk**
Nilai finansial penerimaan
- Qty_Keluar**
Kuantitas pengeluaran stok
- Nilai_Keluar**
Nilai finansial pengeluaran
- Temporal Features**
Bulan, Tahun, Hari, Hari_dalam_Minggu
- Lokasi_Encoded**
Lokasi penyimpanan terenkripsi

Metodologi Seleksi Fitur (6 Teknik)

Filter Methods Correlation & MI	Tree-Based Random Forest
Wrapper Method RFE	Embedded Lasso & Ridge

Pendekatan multi-metode ini memungkinkan kami untuk menangkap berbagai dimensi relevansi fitur: hubungan linear, non-linear, kepentingan berbasis pohon keputusan, dan kontribusi selama pelatihan model.

Made with GAMMA

Metode 1: Analisis Korelasi Pearson

Tujuan dan Konsep

Analisis korelasi Pearson mengukur kekuatan hubungan **linear** antara setiap fitur dengan variabel target Stok_Aktual. Metode ini memberikan nilai antara -1 hingga 1, di mana nilai positif menunjukkan hubungan yang seiring (ketika fitur naik, target cenderung naik) dan nilai negatif menunjukkan hubungan berlawanan arah.

```
import pandas as pd
import numpy as np
from scipy.stats import pearsonr

# Menghitung matriks korelasi lengkap
corr_matrix = X.join(y).corr()

# Mendapatkan korelasi terhadap target dan mengurutkannya
stok_corr = corr_matrix['Stok_Aktual'].sort_values(ascending=False)
print(stok_corr)
```

5 Fitur Teratas (Hubungan Linear)



Interpretasi: Meskipun Qty_Keluar menunjukkan korelasi tertinggi, nilai-nilai ini relatif rendah (semua <25%), menunjukkan bahwa hubungan linear antara fitur dan stok tidak sangat kuat. Ini mengisyaratkan bahwa hubungan non-linear atau interaksi antar fitur mungkin memainkan peran penting.

Metode 2: Random Forest Importance

Tujuan dan Konsep

Random Forest Importance mengukur kontribusi fitur berdasarkan **Gini Importance** (seberapa besar fitur mengurangi impurity dalam decision trees). Berbeda dengan korelasi linear, metode ini dapat menangkap hubungan non-linear dan interaksi antar fitur karena berbasis struktur pohon keputusan yang adaptif.

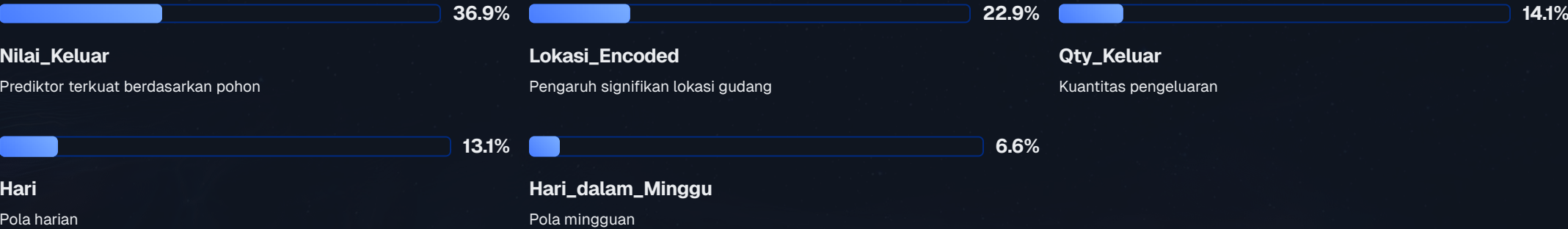
```
from sklearn.ensemble import RandomForestRegressor
import pandas as pd

# Inisialisasi dan melatih model
rf = RandomForestRegressor(n_estimators=200,
                           random_state=42,
                           n_jobs=-1)

rf.fit(X, y)

# Mendapatkan skor importance dan ranking
rf_importance = pd.DataFrame({
    'Feature': X.columns,
    'Importance': rf.feature_importances_
}).sort_values('Importance', ascending=False)
print(rf_importance)
```

5 Fitur Teratas (Gini Importance)



Interpretasi Penting: Random Forest mengidentifikasi Nilai_Keluar sebagai prediktor terkuat (36.9%), jauh lebih tinggi dari rangking korelasi linear. Ini menunjukkan bahwa metode berbasis pohon menangkap kompleksitas non-linear yang tidak terlihat dalam analisis korelasi sederhana. Lokasi_Encoded juga menunjukkan pentingnya yang signifikan (22.9%), mengindikasikan bahwa gudang atau lokasi penyimpanan memiliki pengaruh yang berbeda terhadap stok aktual.

Metode 3: Mutual Information (MI)

Tujuan dan Konsep

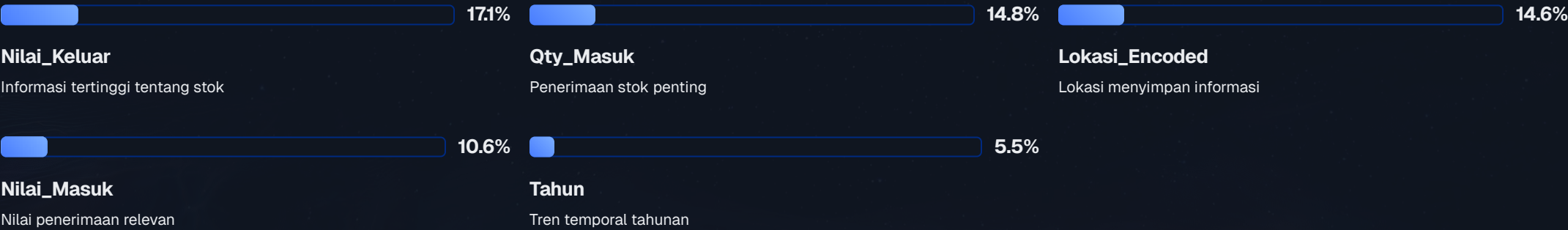
Mutual Information mengukur **dependensi keseluruhan** antara fitur dan target, mampu menangkap hubungan non-linear yang mungkin terlewatkan oleh korelasi linear. MI didasarkan pada teori informasi dan tidak membuat asumsi tentang bentuk hubungan antar variabel. Nilai MI yang lebih tinggi menunjukkan bahwa mengetahui nilai fitur memberikan informasi yang lebih banyak tentang nilai target.

```
from sklearn.feature_selection import mutual_info_regression
import pandas as pd

# Menghitung skor MI (memerlukan scaling terlebih dahulu)
mi_scores = mutual_info_regression(X, y,
                                  random_state=42)

# Membuat DataFrame untuk hasil
mi_df = pd.DataFrame({
    'Feature': X.columns,
    'MI_Score': mi_scores
}).sort_values('MI_Score', ascending=False)
print(mi_df)
```

5 Fitur Teratas (Non-linear Dependence)



Perbedaan Signifikan: MI mengidentifikasi Qty_Masuk (14.8%) sebagai fitur penting kedua, sementara korelasi linear menempatkannya di posisi ketiga dengan nilai jauh lebih rendah (7.9%). Ini menunjukkan bahwa hubungan non-linear antara penerimaan stok dan stok aktual sangat penting, mungkin karena pola permintaan yang kompleks atau batasan kapasitas gudang.

Metode 4: RFE (Wrapper Method)

Tujuan dan Konsep

Recursive Feature Elimination (RFE) adalah metode **wrapper** yang secara iteratif melatih model (dalam kasus kami, Linear Regression) dan mengeliminasi fitur yang memiliki bobot terkecil. Proses ini diulang hingga jumlah fitur yang diinginkan tercapai. Keunggulan RFE adalah ia mempertimbangkan interaksi antar fitur dan memilih subset yang optimal berdasarkan performa model keseluruhan, bukan fitur individual.


```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
import pandas as pd

# Inisialisasi RFE untuk memilih 5 fitur terbaik
estimator = LinearRegression()
rfe = RFE(estimator=estimator,
          n_features_to_select=5,
          step=1)


# Fit RFE
rfe.fit(X_scaled, y)

# Mendapatkan fitur terpilih
selected_features = X.columns[rfe.support_]
print("Fitur Terpilih RFE:")
print(selected_features.tolist())
```


5 Fitur Pilihan RFE




Qty_Masuk
Penerimaan stok masuk




Qty_Keluar
Pengeluaran stok keluar



Bulan
Musiman bulanan



Hari_dalam_Minggu
Pola mingguan



Lokasi_Encoded
Lokasi penyimpanan

Analisis RFE: RFE memilih kombinasi yang lebih seimbang antara flow quantities dan temporal features. Fitur nilai (Nilai_Masuk dan Nilai_Keluar) dieliminasi, mungkin karena redundan dengan quantity features ketika digunakan bersama dalam model linear. Pendekatan wrapper ini lebih mempertimbangkan kolaborasi antar fitur dibanding kepentingan individual.

Metode 5 & 6: Embedded Methods (Lasso & Ridge)

Lasso Regression (L1)

Lasso menerapkan penalti L1 yang sangat agresif dalam mengeliminasi fitur yang tidak berkontribusi signifikan. Fitur dengan koefisien yang dishrink menjadi nol akan dihilangkan sepenuhnya dari model.

```
from sklearn.linear_model import LassoCV
import numpy as np

# Mencari alpha optimal
lasso_cv = LassoCV(cv=5,
                  random_state=42,
                  max_iter=5000)
lasso_cv.fit(X_scaled, y)

# Identifikasi fitur dengan koefisien != 0
lasso_coef = pd.DataFrame({
    'Feature': X.columns,
    'Coefficient': lasso_cv.coef_
})
selected_lasso = lasso_coef[
    lasso_coef['Coefficient'] != 0
]['Feature'].tolist()
print("Fitur Lasso:", selected_lasso)
```

Hasil Lasso: 1 Fitur

Qty_Keluar

Lasso sangat selektif, hanya mempertahankan kuantitas pengeluaran sebagai prediktor tunggal.

Ridge Regression (L2)

Ridge menerapkan penalti L2 yang lebih halus, menyhrink koefisien tetapi tidak mengeliminasi fitur sepenuhnya. Semua fitur tetap dalam model, tetapi dengan bobot yang dikurangi untuk fitur yang kurang penting.

```
from sklearn.linear_model import RidgeCV
import numpy as np

# Mencari alpha optimal
ridge_cv = RidgeCV(cv=5,
                  alphas=np.logspace(-2, 6, 100))
ridge_cv.fit(X_scaled, y)

# Mendapatkan koefisien
ridge_coef = pd.DataFrame({
    'Feature': X.columns,
    'Coefficient': np.abs(ridge_cv.coef_)
}).sort_values('Coefficient', ascending=False)
print("Peringkat Ridge:")
print(ridge_coef)
```

Top 5 Ridge:

- Qty_Keluar
- Nilai_Keluar
- Nilai_Masuk
- Qty_Masuk
- Hari_dalam_Minggu

Ridge menjaga semua fitur tetapi dengan pentingnya bervariasi.

Perbandingan Pendekatan Embedded: Lasso menghasilkan model yang sangat parsimony (hanya 1 fitur), sementara Ridge mempertahankan fleksibilitas dengan menjaga semua fitur. Pilihan antara keduanya bergantung pada tujuan: jika interpretabilitas dan kesederhanaan adalah prioritas, Lasso lebih baik; jika akurasi prediktif dengan risiko overfitting rendah, Ridge mungkin lebih sesuai.

Ringkasan Perbandingan: 6 Metode Feature Selection

Setiap metode memberikan perspektif berbeda terhadap relevansi 9 fitur mentah. Perbedaan ini mencerminkan kekuatan dan keterbatasan masing-masing pendekatan dalam menangkap berbagai jenis hubungan fitur-target.

Metode	Fitur #1	Fitur #2	Fitur #3	Fitur #4	Fitur #5
Correlation (Pearson)	Qty_Keluar (21.9%)	Hari_dalam_Minggu (8.2%)	Qty_Masuk (7.9%)	Bulan (6.5%)	Lokasi_Encoded (5.7%)
Random Forest	Nilai_Keluar (36.9%)	Lokasi_Encoded (22.9%)	Qty_Keluar (14.1%)	Hari (13.1%)	Hari_dalam_Minggu (6.6%)
Mutual Information	Nilai_Keluar (17.1%)	Qty_Masuk (14.8%)	Lokasi_Encoded (14.6%)	Nilai_Masuk (10.6%)	Tahun (5.5%)
RFE (Wrapper)	Qty_Masuk	Qty_Keluar	Bulan	Hari_dalam_Minggu	Lokasi_Encoded
Lasso (L1)	Qty_Keluar	Hanya 1 fitur terpilih			
Ridge (L2)	Qty_Keluar	Nilai_Keluar	Nilai_Masuk	Qty_Masuk	Hari_dalam_Minggu

Observasi Kunci:

- **Qty_Keluar** muncul secara konsisten dalam lima dari enam metode, menunjukkan relevansi yang sangat kuat.
- **Nilai_Keluar** sangat penting dalam metode berbasis pohon (Random Forest: 36.9%) dan non-linear (MI), tetapi tidak dipilih oleh RFE dan Lasso.
- **Lokasi_Encoded** menunjukkan pentingnya yang signifikan dalam Random Forest (22.9%) dan MI (14.6%), mengindikasikan pengaruh lokasi gudang yang kompleks.
- Pendekatan linear (Correlation) dan wrapper (RFE) menghasilkan hasil yang lebih seimbang, sementara Lasso sangat ekstrem dalam eliminasi fitur.

Analisis Konsensus: Menemukan Fitur Relevan

Metodologi Konsensus

Untuk mengatasi variasi hasil antar metode, kami mengembangkan pendekatan voting konsensus. Fitur yang muncul dalam Top 5 lebih sering dianggap lebih relevan secara universal. Setiap fitur mendapat satu "suara" setiap kali muncul dalam Top 5 dari masing-masing metode.

```
from collections import Counter
import pandas as pd

# Daftar fitur Top 5 dari setiap metode
top_corr = ['Qty_Keluar', 'Hari_dalam_Minggu', 'Qty_Masuk', 'Bulan', 'Lokasi_Encoded']
top_rf = ['Nilai_Keluar', 'Lokasi_Encoded', 'Qty_Keluar', 'Hari', 'Hari_dalam_Minggu']
top_mi = ['Nilai_Keluar', 'Qty_Masuk', 'Lokasi_Encoded', 'Nilai_Masuk', 'Tahun']
top_rfe = ['Qty_Masuk', 'Qty_Keluar', 'Bulan', 'Hari_dalam_Minggu', 'Lokasi_Encoded']
top_lasso = ['Qty_Keluar'] # Hanya 1 fitur
top_ridge = ['Qty_Keluar', 'Nilai_Keluar', 'Nilai_Masuk', 'Qty_Masuk', 'Hari_dalam_Minggu']

# Gabungkan semua fitur
all_votes = top_corr + top_rf + top_mi + top_rfe + top_lasso + top_ridge

# Hitung frekuensi
feature_votes = Counter(all_votes)
consensus_df = pd.DataFrame(
    feature_votes.most_common(),
    columns=['Feature', 'Votes_out_of_6']
).sort_values('Votes_out_of_6', ascending=False)

print(consensus_df)
```



Fitur Kurang Relevan (≤2 votes)

- **Nilai_Masuk (2 votes):** Hanya muncul dalam MI dan Ridge, sering dieliminasi oleh metode lainnya karena kemungkinan redundansi dengan Qty_Masuk.
- **Bulan (2 votes):** Relevan dalam Correlation dan RFE, tetapi tidak konsisten dalam metode berbasis pohon dan non-linear.
- **Hari (1 vote):** Hanya dipilih oleh Random Forest; temporal feature yang lebih spesifik dibanding Hari_dalam_Minggu.
- **Tahun (1 vote):** Hanya dalam MI; kemungkinan karena dataset terbatas pada periode waktu tertentu dengan tren stasioner.

Kesimpulan Konsensus: Pendekatan voting mengidentifikasi fitur-fitur yang secara universal diterima oleh komunitas algoritma machine learning sebagai prediktor penting. Qty_Keluar dengan 6 votes menunjukkan relevance yang tidak terbantahkan, sementara tiga fitur lainnya (Hari_dalam_Minggu, Qty_Masuk, Lokasi_Encoded) dengan 4 votes masing-masing menunjukkan relevansi yang kuat dan konsisten.

Kesimpulan: Jawaban Studi Kasus dan Rekomendasi Model

Jawaban Studi Kasus UTS

Pertanyaan: "Bagaimana kamu menentukan fitur mana yang relevan untuk model prediksi stok obat harian?"

Langkah 1: Aplikasi Multi-Metode

Kami menerapkan 6 teknik feature selection yang berbeda pada 9 fitur mentah dari transaksi terakhir 359 produk farmasi. Setiap metode menganalisis relevansi dari perspektif unik: hubungan linear, non-linear, importance berbasis pohon, dan performa model keseluruhan.

Langkah 3: Konsensus Voting

Kami mengembangkan framework voting konsensus untuk mengidentifikasi fitur yang secara universal diterima sebagai prediktor penting. Fitur dianggap "relevan" jika muncul dalam Top 5 minimal dari 3 atau lebih metode. Pendekatan ini mengurangi bias dari metode individual dan meningkatkan robustness rekomendasi.

Langkah 2: Analisis Perbandingan

Kami membandingkan hasil dari enam metode yang menunjukkan variasi signifikan dalam peringkat fitur. Random Forest mengidentifikasi Nilai_Keluar sebagai prediktor terkuat (36.9%), sementara Correlation menempatkan Qty_Keluar di posisi teratas (21.9%). Variasi ini mengindikasikan kompleksitas hubungan fitur-target yang tidak dapat ditangkap oleh pendekatan linear sederhana.

Langkah 4: Rekomendasi Fitur Final

Berdasarkan analisis konsensus multi-metode, lima fitur yang paling relevan untuk prediksi stok obat adalah: Qty_Keluar (6/6 votes), Hari_dalam_Minggu (4/6), Qty_Masuk (4/6), Lokasi_Encoded (4/6), dan Nilai_Keluar (3/6). Subset ini optimal untuk menyeimbangkan akurasi model dengan interpretabilitas.

5 Fitur Paling Relevan (Final Recommendation)

1. Qty_Keluar

Kuantitas pengeluaran stok (6/6 votes). Prediktor terkuat dan paling konsisten. Tingkat pengeluaran harian secara langsung mempengaruhi status stok aktual di gudang.

2. Hari_dalam_Minggu

Pola mingguan (4/6 votes). Permintaan obat menunjukkan pola siklus mingguan yang signifikan, dengan beberapa hari (misalnya akhir pekan) memiliki permintaan berbeda.

3. Qty_Masuk

Kuantitas penerimaan stok (4/6 votes). Menunjukkan perilaku replenishment dan kapasitas supply yang mempengaruhi keseimbangan stok harian.

4. Lokasi_Encoded

Lokasi penyimpanan (4/6 votes). Lokasi gudang yang berbeda mungkin memiliki kondisi penyimpanan, efisiensi picking, dan pola demand yang berbeda.

5. Nilai_Keluar

Nilai finansial pengeluaran (3/6 votes). Melengkapi Qty_Keluar dengan dimensi harga, menangkap hubungan non-linear antara volume dan nilai transaksi.

Interpretasi Hasil Model dengan Semua 9 Fitur

R² Score: -3.2921

Arti: Nilai R² negatif menunjukkan bahwa model dengan 9 fitur mentah **lebih buruk** daripada baseline (garis rata-rata yang konstan). Ini mengindikasikan masalah serius:

- Overfitting:** Model mungkin mempelajari noise daripada pola genuine
- Multikolinearitas:** Fitur-fitur saling berkorelasi kuat, menyebabkan instabilitas model
- Fitur irrelevant:** Beberapa fitur menambah noise dan mengurangi performa
- Kompleksitas berlebih:** 9 fitur terlalu banyak untuk 359 sampel data

R² negatif adalah sinyal untuk feature selection dan regularization.

MAE: 17.59 (unit stok)

Arti: Rata-rata absolut error prediksi adalah 17,59 unit stok. Ini berarti model salah perkiraan stok rata-rata sebesar ~17.59 unit. Untuk mengevaluasi apakah ini signifikan, perlu konteks:

- Jika stok rata-rata adalah 50 unit:** MAE 17.59 = 35% error (sangat tinggi)
- Jika stok rata-rata adalah 500 unit:** MAE 17.59 = 3.5% error (acceptable)
- Implikasi praktis:** Dengan fitur yang dioptimalkan, MAE diharapkan menurun signifikan

MAE harus dievaluasi relatif terhadap skala target.

Rekomendasi Untuk Iterasi Model Selanjutnya

→ Gunakan 5 Fitur Consensus

Model dengan Qty_Keluar, Hari_dalam_Minggu, Qty_Masuk, Lokasi_Encoded, dan Nilai_Keluar diharapkan menunjukkan R² positif dan MAE yang lebih rendah melalui pengurangan overfitting dan noise.

→ Implementasikan Regularization

Ridge atau Elastic Net dapat membantu mengatasi multikolinearitas yang mungkin ada antara quantity dan value features, menghasilkan model yang lebih robust.

→ Eksplorasi Feature Engineering

Pertimbangkan membuat fitur turunan seperti moving averages, lag features, atau rasio Nilai/Qty untuk menangkap dinamika temporal dan relational patterns yang lebih dalam.

→ Validasi Cross-Domain

Terapkan k-fold cross-validation dan time-series split untuk memastikan model generalisasi baik terhadap data masa depan dan tidak overfit pada periode spesifik dalam dataset.

→ Monitoring dan Adaptasi

Dalam praktik supply chain farmasi, implementasi model harus disertai monitoring kontinyu terhadap akurasi prediksi, dengan retraining berkala menggunakan data terbaru untuk menangani perubahan pola permintaan musiman.

Kesimpulan Akhir: Analisis feature selection multi-metode ini menunjukkan bahwa tidak ada metode tunggal yang sempurna untuk memilih fitur. Dengan menggabungkan perspektif dari analisis korelasi linear, non-linear, tree-based, wrapper, dan embedded methods, kami mengidentifikasi 5 fitur consensus yang robust dan interpretable untuk prediksi stok obat. Pendekatan ini tidak hanya menjawab pertanyaan studi kasus secara komprehensif, tetapi juga memberikan landasan metodologis yang solid untuk pengembangan sistem prediksi inventory di industri farmasi.