BAB I

PENJELASAN METODOLOGI

A. Langkah Pengerjaan

Langkah-langkah pengerjaan pengembangan solusi analisis big data Fesmaro DataCo adalah sebagai berikut.

a. Pertama, kami menentukan topik analisis terhadap sumber data yang ada. Kolom "late_delivery_risk" memiliki tipe data biner yang cocok untuk diprediksi berdasarkan atribut-atribut lain seperti harga dan kuantitas pemesanan, lama pengantaran, dan lain-lain. Oleh karena itu, kami memutuskan untuk melakukan prediksi resiko *late delivery* dari DataCo Supply Chain.

b. Kami melakukan rancangan solusi berdasarkan sumber data yang sudah dipahami secara sekilas dari luar.

   i. Sumber data memiliki banyak kolom yang tidak relevan dalam pemodelan, seperti data pelanggan dan deskripsi produk. Oleh karena itu, diperlukan *pre-processing* untuk menyingkirkan kolom-kolom yang tidak berguna.

   ii. Kemudian, *exploratory data analysis* (EDA) dilakukan terhadap beberapa atribut utama untuk melihat apakah distribusi data tersebar merata atau tidak menggunakan berbagai macam graf seperti *bar chart* dan histogram.

   iii. Data diolah lebih lanjut melalui proses *feature engineering*: a) dilakukan *parsing* terhadap data bertipe *datetime* berdasarkan tahun, bulan, tanggal, hari, dan jam; b) dilakukan *categorical encoding* untuk memisahkan setiap kelas atribut kategorikal menjadi kolom terpisah; c) *Outlier* data dicari menggunakan *z-score* karena ukuran data yang besar dan diganti dengan nilai median setiap kolom *float* untuk menghindari bias; d) Normalisasi min-max untuk semua kolom *float* untuk meningkatkan akurasi prediksi model.

iv. Pelatihan dan perbandingan evaluasi menggunakan berbagai macam model menggunakan dataset yang sudah diolah.

c. Kami mengembangkan solusi yang telah dirancang sebelumnya menggunakan bahasa pemrograman Python di *notebook* ".ipynb". Kami menjalankan kode menggunakan *website* Kaggle dengan GPU P100 untuk mempercepat waktu pelatihan.

d. Kami menguji performa beberapa model *machine learning* dan mengevaluasi skor performa masing-masing model melalui akurasi. Lalu, kami melihat skor log loss dari model *machine learning* dengan akurasi paling tinggi.

B. Alat dan Teknologi

Bahasa pemrograman yang digunakan adalah Python. *Library* yang dimanfaatkan termasuk: a) Pandas, numpy, scipy, dan math untuk pengolahan dan kalkulasi *outlier* data; b) Seaborn dan matplotlib untuk visualisasi data melalui grafik; c) Sklearn untuk membantu proses pelatihan dan evaluasi model; d) Autologging_ML untuk melatih beberapa model ML dan membandingkan hasil evaluasi dari setiap model tersebut.

C. Sumber Data

Data diperoleh dari *website* Kaggle yang berisi data *supply chain* dari perusahaan DataCo Global. Dataset yang digunakan memiliki format ".csv", dengan total 180519 baris data dan 53 kolom data. Pertama-tama, kolom yang digunakan untuk pelatihan model dipisahkan dari dataset mentah. Kolom-kolom yang digunakan adalah:

Nama Kolom	**Deskripsi**
Type (kategorikal)	Tipe transaksi: CASH, DEBIT, PAYMENT, TRANSFER
Days for shipping (real)	Hari pengiriman sebenarnya dari produk yang dibeli
Days for shipment (scheduled)	Hari pengiriman terjadwal dari produk yang dibeli
Benefit per order	Pendapatan per pesanan yang ditempatkan
Sales per customer	Total penjualan per pelanggan yang dibuat per pelanggan
Late_delivery_risk	Variabel kategorikal yang menunjukkan jika

	pengiriman terlambat (1), maka pengiriman tersebut tidak terlambat (0).
order date (DateOrders)	Tanggal saat pesanan dibuat
Order Item Discount	Nilai diskon pemesanan
Order Item Discount Rate	Persentase nilai diskon pemesanan
Order Item Product Price	Harga produk tanpa diskon
Order Item Profit Ratio	Rasio Keuntungan Item Pesanan
Order Item Quantity	Jumlah produk per pesanan
Sales	Nilai dalam penjualan
Order Item Total	Jumlah total per pesanan
Order Profit Per Order	Keuntungan Per Pesanan
Product Price	Harga produk
shipping date (DateOrders)	Tanggal dan waktu pengiriman yang tepat
Shipping Mode (kategorikal)	Mode pengiriman: Standard Class , First Class , Second Class , Same Day

Dari kolom-kolom yang terpilih, kolom “shipping date” dan “order date” di-*parsing* sehingga masing-masing kolom tersebut memiliki kolom tahun, bulan, tanggal, hari, dan jam. Lalu, dilakukan *categorical encoding* pada kolom kategorikal “Type” dan “Shipping Mode” untuk memisahkan setiap kelas atribut kategorikal menjadi kolom terpisah, sebelum nilai data diubah menjadi biner 0 atau 1. Kemudian, dideteksi nilai *outlier* di kolom *float* dengan menggunakan *z-score* untuk kemudian digantikan dengan nilai median dari kolom tersebut. Berikut adalah banyak nilai *outlier* dari setiap kolom *float*:

Nama Kolom	**Banyak baris data *outlier***
'Benefit per order'	3608
'Sales per customer'	477
'Order Item Discount'	2106
'Order Item Discount Rate'	0

'Order Item Product Price'	488
'Order Item Profit Ratio'	6013
'Sales'	467
'Order Item Total'	477
'Order Profit Per Order'	3608
'Product Price'	488

Setelah itu, dilakukan normalisasi *min-max* dari *range* 0-1 pada kolom *float* untuk meningkatkan akurasi model menggunakan rumus:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Seluruh kolom data setelah dilakukan pengolahan bersifat numerik.

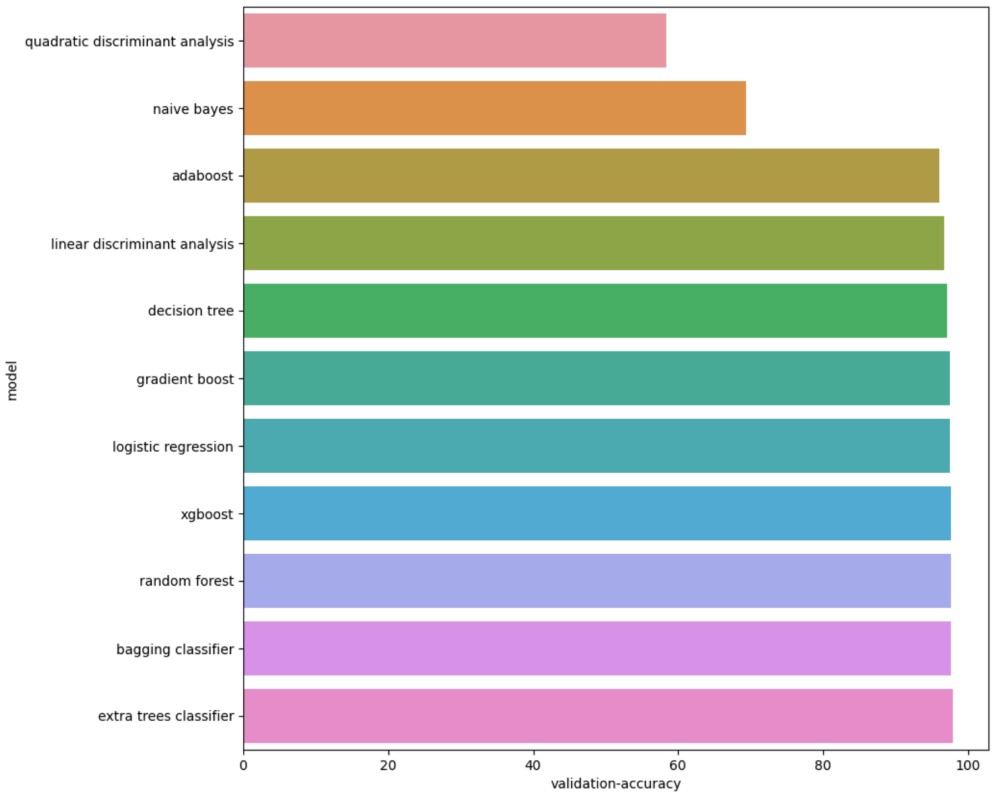
BAB II

ANALISIS HASIL DAN EVALUASI

A. Metrik Evaluasi

Metrik evaluasi yang digunakan untuk membandingkan beberapa model ML adalah *validation accuracy*, yaitu akurasi yang didasari oleh data validasi. Berikut adalah tabel dan grafik perbandingan skor *validation accuracy* dari setiap model yang digunakan.

	model	validation-accuracy
10	quadratic discriminant analysis	58.353645
0	naive bayes	69.407822
3	adaboost	96.061378
9	linear discriminant analysis	96.742743
1	decision tree	97.080656
4	gradient boost	97.485043
6	logistic regression	97.485043
5	xgboost	97.573676
2	random forest	97.598604
7	bagging classifier	97.659539
8	extra trees classifier	97.889431



Hasil perbandingan menunjukkan bahwa model *extra trees classifier* memiliki skor *validation accuracy* paling tinggi, yakni sebesar 97,8894. Model *extra trees classifier* kemudian dikalkulasikan skor *log loss* oleh kakas Sklearn dan diperoleh skor *log loss* 0,0821.

B. Hasil Pengujian

Berikut adalah data lengkap pengujian setiap model ML yang dilatih.

	model	training-accuracy	training-precision	training-recall	training-f1	training-confusion matrix	validation-accuracy	validation-precision	validation-recall	validation-f1	validation-confusion matrix	training-classification report	validation-classificatior report
10	quadratic discriminant analysis	58.313887	78.344998	53.859168	43.398329	[[5040, 60195], [6, 79174]]	58.353645	78.377937	53.897640	43.473322	[[1272, 15035], [1, 19796]]	precision recall f1-score ...	precision recall f1-score ...
0	naive bayes	69.817540	73.308507	71.451765	69.511150	[[57652, 7583], [36005, 43175]]	69.407822	72.909809	71.051942	69.089997	[[14360, 1947], [9098, 10699]]	precision recall f1-score ...	precision recall f1-score ...
3	adaboost	96.134058	96.135463	96.056838	96.094413	[[62141, 3094], [2489, 76691]]	96.061378	96.071536	95.973958	96.020120	[[15503, 804], [618, 19179]]	precision recall f1-score ...	precision recall f1-score ...
9	linear discriminant analysis	96.846588	96.969998	96.683271	96.807770	[[61968, 3267], [1287, 77893]]	96.742743	96.875811	96.570400	96.701977	[[15457, 850], [326, 19471]]	precision recall f1-score ...	precision recall f1-score ...
1	decision tree	100.000000	100.000000	100.000000	100.000000	[[65235, 0], [0, 79180]]	97.080656	97.049343	97.057444	97.053374	[[15788, 519], [535, 19262]]	precision recall f1-score ...	precision recall f1-score ...
4	gradient boost	97.567427	97.875878	97.307427	97.531351	[[61722, 3513], [0, 79180]]	97.485043	97.807293	97.215920	97.447162	[[15399, 908], [0, 19797]]	precision recall f1-score ...	precision recall f1-score ...
6	logistic regression	97.566042	97.874720	97.305894	97.529937	[[61720, 3515], [0, 79180]]	97.485043	97.807293	97.215920	97.447162	[[15399, 908], [0, 19797]]	precision recall f1-score ...	precision recall f1-score ...
5	xgboost	98.263338	98.463112	98.078799	98.240545	[[62735, 2500], [8, 79172]]	97.573676	97.849092	97.331334	97.538731	[[15463, 844], [32, 19765]]	precision recall f1-score ...	precision recall f1-score ...
2	random forest	100.000000	100.000000	100.000000	100.000000	[[65235, 0], [0, 79180]]	97.598604	97.894015	97.345957	97.563379	[[15448, 859], [8, 19789]]	precision recall f1-score ...	precision recall f1-score ...
7	bagging classifier	99.922446	99.927713	99.915776	99.921706	[[65135, 100], [12, 79168]]	97.659539	97.822498	97.485844	97.629698	[[15604, 703], [142, 19655]]	precision recall f1-score ...	precision recall f1-score ...
8	extra trees classifier	100.000000	100.000000	100.000000	100.000000	[[65235, 0], [0, 79180]]	97.889431	98.133415	97.671148	97.860140	[[15559, 748], [14, 19783]]	precision recall f1-score ...	precision recall f1-score ...

C. Kesimpulan dan Rekomendasi

Dalam prediksi resiko *late delivery* DataCo Supply Chain, model *extra classifier* memiliki performa yang paling baik dengan skor *validation accuracy* 97,8894 dan *log loss* 0,0821. Untuk pengembangan lebih lanjut, dapat dicari model ML lain atau dilakukan eksperimen terhadap parameter model-model ML untuk memperbaiki skor prediksi, atau dicari atribut-atribut data lain yang cocok untuk dilatih.

Sumber:

- Data: https://www.kaggle.com/datasets/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis
- Repository: https://github.com/Breezy-DR/prediksi-resiko-late-delivery-dataco/tree/main
- Notebook: https://www.kaggle.com/code/farreldr/prediksi-resiko-late-delivery-dataco-supply-chain