

Perbandingan Random Search dan Algoritma Genetika dalam Penyetelan Hyperparameter XGBoost pada Retail Sales Forecasting

Sheren Afryan Tiastama¹, Indra Budi²

sheren.afryan@ui.ac.id¹, indra.budi@ui.ac.id²

^{1,2} Universitas Indonesia

Informasi Artikel

Diterima : 8 Jul 2024
Direvisi : 29 Jul 2024
Disetujui : 8 Agu 2024

Kata Kunci

Algoritma Genetika,
Random Search, Retail,
Sales Forecasting,
XGBoost

Abstrak

Sales merupakan bagian dari faktor penting yang mempengaruhi sebuah perusahaan dalam menentukan 2 hal, yaitu keuntungan dan kerugian pada perusahaan. Strategi yang tepat untuk menentukan jumlah *sales* dapat dilakukan melalui *forecasting*. Oleh karena itu, *sales forecasting* membutuhkan teknik yang tepat untuk memproduksi hasil yang akurat. *Machine learning* telah dibuktikan dapat membantu mengatasi *sales forecasting*, salah satunya yaitu XGBoost. Namun, XGBoost memiliki banyak *hyperparameter* yang mempengaruhi performanya, hal ini memerlukan metode penyetelan *hyperparameter* untuk menghasilkan *hyperparameter* yang optimal. Random search dan algoritma genetika merupakan metode optimasi untuk menemukan *hyperparameter* yang optimal pada XGBoost. Kedua metode optimasi tersebut dibandingkan pada penelitian ini dengan pengukuran performa RMSE dalam melakukan *retail sales forecasting* pada data penjualan perusahaan *retail* yaitu Rossmann Store yang berasal dari situs Kaggle. Penelitian ini memperoleh hasil random search lebih unggul dari algoritma genetika dengan nilai RMSE pada proses latih dan proses uji sebesar 0.123 dan 0.122. Sementara itu, nilai RMSE yang dihasilkan oleh algoritma genetika pada proses latih dan proses uji yaitu sebesar 0.333 dan 0.332.

Keywords

Genetic Algorithm, Random
Search, Retail, Sales
Forecasting, XGBoost

Abstract

Sales is part of the important factor that influences a company in determining two things, namely profits and losses on the company. The right strategy to determine the amount of sales can be done through forecasting. Therefore, sales forecasting requires the right technique to produce accurate results. Machine learning has been proven to help overcome sales forecasting, one of which is XGBoost. However, XGBoost has many hyperparameters that affect its performance, it requires a hyperparameter setting method to produce an optimal hyperparameter. Random searches and genetic algorithms are optimized methods to find the optimal hyperparameter on XGBoost. The two methods of optimization were compared in this study with the measurement of RMSE performance in doing retail sales forecasting on the sales data of the retail company Rossmann Store which comes from the Kaggle site. The research obtained random search results superior to the genetic algorithm with RMSE values on the training process and the testing process are 0.123 and 0.122. Meanwhile, the RMSE values generated by genetic algorithms in the training and testing process are 0.333 and 0.332.

A. Pendahuluan

Sales merupakan bagian dari faktor penting yang mempengaruhi sebuah perusahaan dalam menentukan 2 hal, yaitu keuntungan dan kerugian pada perusahaan[1]. Hal ini disebabkan oleh pernyataan bahwa *sales* diposisikan sebagai sumber penghasilan dari sebuah perusahaan[2]. *Sales* dapat dipengaruhi oleh beberapa faktor-faktor eksternal, yaitu cuaca, tren musiman yang terjadi pada daerah lokasi toko, persaingan dari perusahaan/toko retail yang lain, *online shopping*[3], dan pandemi[4]. Kasus pandemi yang telah terjadi seperti pandemi Covid-19 mempengaruhi jumlah *sales* pada banyak industri, seperti pada industri retail, berdasarkan hasil survey yang dinyatakan oleh Bank Indonesia pada bulan Agustus tahun 2020, penurunan jumlah *sales* pada perusahaan di bidang *retail* atau *retail sales* terjadi dalam angka sekitar 196.6 atau -9.2% (yoy)[4]. Kondisi ini menyebabkan siklus perekonomian membeku[4]. Faktor-faktor eksternal pada *sales* tersebut memicu perusahaan untuk melakukan aksi internal seperti menerapkan diskon, melakukan promosi, dan menetapkan harga yang tepat pada setiap produk barang[3]. Hal ini menunjukkan bahwa perusahaan membutuhkan strategi yang tepat dalam menentukan jumlah *sales*[2] untuk menjaga grafik *sales* agar tetap dalam kondisi yang baik di setiap tahun[1]. Strategi tersebut dapat ditempuh melalui *forecasting* pada *sales* yang akurat dan tepat[1].

Forecasting berfungsi sebagai dasar dari proses perencanaan dan pelaksanaan aktifitas pada *supply chain* yaitu *sourcing*, pembuatan produk, dan distribusi produk kepada pelanggan[5]. Manajemen *supply chain* dan inventaris melibatkan berbagai masalah yang cukup kompleks, dimana seseorang yang berperan sebagai pembuat keputusan (produsen, manajer toko, distributor) perlu melakukan pertimbangan terhadap aspek-aspek terkait, seperti tenaga kerja, transportasi, biaya, tren penawaran dan permintaan, tingkat inventaris, serta risiko dan keuntungan[6]. Seiring dengan berjalannya waktu, *forecasting* menjadi lebih kompleks dan perannya menjadi lebih kritis, dimana hal ini disebabkan oleh tren globalisasi *supply chain*, tingginya tingkat keanekaragaman produk, siklus hidup produk yang pendek, dan persaingan pasar yang semakin kompetitif[5]. Kunci dari permasalahan ini adalah *product sales forecasting* yang mempengaruhi proses pembuatan keputusan[6]. Jika *sales forecasting* yang dihasilkan tidak akurat, ketika nilainya terlalu besar, maka tingkat inventaris pada perusahaan berada dalam posisi tidak aman, sebaliknya, ketika nilainya terlalu kecil, maka kesempatan perusahaan dalam upaya meraih keuntungan akan hilang[7]. Oleh karena itu, *sales forecasting* membutuhkan teknik yang tepat untuk memproduksi hasil yang akurat.

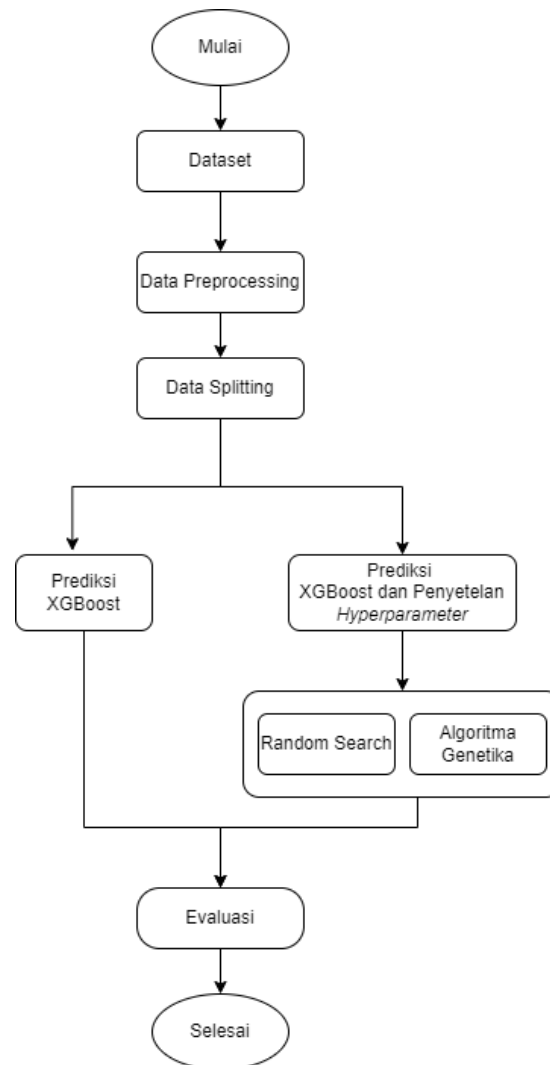
Data *sales* termasuk dalam *time series* data[8]. Sehingga, *sales forecasting* memerlukan metode yang mampu memproses data *time series*. Algoritma *machine learning* mampu menemukan pola dalam data *time series* dengan kemampuan generalisasi dan pemetaan yang kuat[8]. Pola tersebut dapat bersifat pola yang rumit pada dinamika data penjualan, namun bisa ditemukan dengan menggunakan metode algoritma *machine learning*[8]. Hal ini dibuktikan pada penelitian sebelumnya[9] yang melakukan *sales forecasting* menggunakan algoritma *machine learning Extreme Gradient Boosting* (XGBoost) dengan tingkat akurasi yang baik yaitu nilai *Mean Squared Error* sebesar 0.092. Penelitian lainnya[10], dilakukan perbandingan beberapa algoritma *machine learning* dalam melakukan *sales forecasting* yaitu, *Naive Bayes*, *Logistic Regression*, *Decision Tree*, *Random Forest*, dan

XGBoost yang memperoleh hasil XGBoost sebagai algoritma paling unggul dari lainnya dengan nilai akurasi sebesar 0.95. Penelitian lainnya[11], dilakukan *sales forecasting* menggunakan algoritma *machine learning Linear Regression* yang memperoleh nilai *Mean Absolute Percentage Error* (MAPE) sebesar 1% pada salah satu produk *shampoo*. Penelitian lainnya[12], dilakukan *sales forecasting* menggunakan algoritma *machine learning* XGBoost, *Decision Tree*, dan *Random Forest* yang memperoleh hasil XGBoost menduduki posisi paling tinggi di antara algoritma *Decision Tree* dan *Random Forest* dengan nilai akurasi sebesar 77.82%. Berdasarkan penelitian-penelitian tersebut, dapat ditarik kesimpulan bahwa algoritma XGBoost memiliki performa yang lebih unggul dari algoritma lainnya dalam melakukan *sales forecasting*. Namun, algoritma XGBoost memiliki banyak *hyperparameter* yang mempengaruhi peformanya, seperti *n_estimators*, *max_depth*, *min_child_weight*, *gamma*, *subsample*, *colsample_bytree*, *alpha*, *lambda*, dan *eta* (*learning rate*)[12]. Masalah ini memerlukan penentuan nilai setiap *hyperparameter* yang optimal untuk dapat menghasilkan performa terbaik, sehingga mampu menghasilkan *sales forecasting* yang akurat.

Pada penelitian ini, dilakukan percobaan *sales forecasting* menggunakan algoritma XGBoost dengan membandingkan penggunaan metode *random search* dan algoritma genetika dalam penyetelan *hyperparameter* pada XGBoost melalui perhitungan *Root Mean Squared Error* (RMSE). *Random search* merupakan metode yang digunakan untuk melakukan optimasi dalam penyetelan *hyperparameter* dengan memilih setiap kandidat *hyperparameter* secara *random*. Algoritma Genetika merupakan metode heuristik yang dapat melakukan pencarian *hyperparameter* yang optimal melalui penerapan teori evolusi di dalamnya. Oleh karena itu, pada penelitian ini, dilakukan *sales forecasting* menggunakan XGBoost dan metode optimasi pada penyetelan *hyperparameter*-nya yaitu, metode *random search* dan algoritma genetika. Hal ini bertujuan untuk memberikan sudut pandang baru dalam menemukan teknik terbaik untuk melakukan *sales forecasting*.

B. Metode Penelitian

Pada penelitian ini, penerapan algoritma XGBoost menggunakan metode optimasi penyetelan *hyperparameter*, *random search* dan algoritma genetika dilakukan melalui beberapa tahapan seperti pada Gambar 1 di bawah ini.



Gambar 1. Tahapan Penelitian

1. Dataset *Retail Sales*

Data yang digunakan pada penelitian ini merupakan dataset publik yang berasal dari situs (<https://www.kaggle.com/c/rossmann-store-sales/data>). Data tersebut merupakan data penjualan pada Rossmann Store, sebuah perusahaan ritel pada bidang farmasi yang mengoperasikan lebih dari 3,000 toko pada 7 negara di Eropa. Dataset Rossmann Store terdiri dari 844,338 data pelatihan dan 168,868 data pengujian. Fitur-fitur yang mempengaruhi jumlah penjualan pada perusahaan Rossmann berjumlah 15 fitur dan 1 fitur target yang telah diuraikan pada Tabel 1.

Tabel 1. Data Retail Sales

No	Fitur	Keterangan	Type
1	ID	Id yang merepresentasikan (toko, tanggal)	Integer
2	Store	Id toko	Integer
3	Sales	Jumlah omset penjualan per hari	Integer
4	Customers	Jumlah customer per hari	Integer
5	Open	Kondisi toko (buka=1, tutup=0)	Integer
6	StateHoliday	Mengindikasikan hari libur (a= libur nasional, b=libur paskah, c=libur natal, 0=None). Normalnya, semua toko tutup pada hari libur yang ditentukan oleh negara. Lalu, seluruh sekolah tutup pada hari libur nasional dan <i>weekends</i> .	String
7	SchoolHoliday	Mengindikasi (toko, tanggal) jika terkena dampak dari tutupnya sekolah umum karena libur	Integer
8	StoreType	Tipe toko: a, b, c, dan d	String
9	Assortment	Deskripsi macam-macam level, a = basic, b = extra, c = extended.	String
10	CompetitionDistance	Jarak antara toko dan toko kompetitor (meter)	Integer
11	CompetitionOpenSince[Month/Year]	Perkiraan tahun dan bulan bukanya toko kompetitor	Integer
12	Promo	Mengindikasi adanya promo pada toko di suatu hari tertentu	Integer
13	Promo2	Indikasi partisipasi toko dalam melakukan promo secara berkelanjutan: 0 = toko tidak berpartisipasi, 1 = toko berpartisipasi	Integer
14	Promo2Since[Year/Week]	Mendeskripsikan tahun dan minggu Ketika toko memulai untuk berpartisipasi pada Promo2	Integer
15	PromoInterval	Interval dimulainya penerapan Promo2, Contoh: "Feb,May,Aug,Nov" berarti dimulai setiap bulan Februari, Mei, Agustus, November pada tahun yang diberikan pada toko.	Integer
16	Sale per Customer	Jumlah penjualan setiap customer (target)	Integer

2. Dataset Preprocessing

Tahap selanjutnya yaitu pengolahan data atau *data preprocessing*. Tahap ini bertujuan untuk membuat data yang digunakan lebih akurat. Tahap ini terdiri dari proses *handling missing value* dan *encoding*. Pada proses *handling missing value*, dilakukan pengecekan terhadap data yang *record*-nya tidak terisi atau bernilai NaN, kemudian data tersebut diisi dengan nilai 0,1 atau median, proses ini dilakukan pada fitur CompetitionDistance,

CompetitionOpenSince[Mon/Year], Promo2Since[Mon/Year], PromoInterval, dan Open. Setelah itu, dilakukan proses *encoding* atau perubahan tipe data objek menjadi tipe data numerik pada fitur Assortment, StateHoliday, dan StoreType.

3. Data Splitting

Tahap selanjutnya yaitu pembagian jumlah data atau *data splitting*. Data proses pelatihan yang berjumlah 844,338 dibagi pada rasio 70% sebagai data latih dan 30% data validasi.

4. XGBoost

XGBoost merupakan algoritma yang memanfaatkan *decision tree* dan dibentuk dengan pendekatan *gradient descent*[13]. XGBoost terdiri dari 2 komponen, yaitu pendekatan *gradient descent* berfungsi untuk melakukan optimasi pada *loss function* dan regularisasi parameter untuk mencegah *overfitting*[14]. Konsep utama dari XGBoost adalah meminimalisir fungsi objektif yang terdiri dari *loss function* dan fungsi regularisasi seperti pada rumus di bawah ini[14]:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + (ft(x_i))) + \Omega(ft) \quad (1)$$

Dimana:

l = *loss function* (representasi dari *error* antara nilai aktual y_i dan nilai prediksi $\hat{y}_i$)

ft = model pada pohon ke- t

t = indeks iterasi selama proses optimisasi

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda ||\omega||^2 \quad (2)$$

Dimana:

T = Jumlah *leaf*

ω = Bobot *leaf*

γ dan λ = Koefisien dengan nilai *default* $\lambda = 1$ dan $\gamma = 0$

5. XGBoost dan Penyetelan Hyperparameter

Pada penelitian ini dilakukan *retail sales forecasting* menggunakan XGBoost dengan nilai *hyperparameter default* dan XGBoost dengan optimasi penyetelan *hyperparameter* menggunakan *Random Search* dan Algoritma Genetika. XGBoost memiliki banyak *hyperparameter* yang dapat meningkatkan performanya, *hyperparameter* tersebut diantaranya adalah[12,15,16]:

- a) eta (*learning rate*): penyusutan koefisien setiap *tree* untuk mencegah *overfitting*.
- b) gamma: pengurangan nilai minimum *loss function* untuk membuat *tree-split* yang baru. Semakin besar gamma, maka algoritma akan semakin konservatif.
- c) max_depth: nilai maksimum dari kedalaman setiap *tree*. Semakin besar nilai max_depth, maka model akan semakin kompleks dan dapat menyebabkan *overfitting*.

- d) *n_estimators* : jumlah estimator/kalkulasi iteratif.
- e) *min_child_weight*: nilai minimum dari bobot-bobot setiap *child*. Semakin besar nilai *min_child_weight*, maka algoritma akan menjadi semakin konservatif.
- f) *subsample*: rasio *subsample* dari sampel pelatihan yang digunakan untuk mencegah *overfitting*.
- g) *colsample_bytree*: rasio *subsample* dari kolom-kolom pada setiap *tree* yang digunakan untuk mencegah *overfitting*.
- h) *lambda*: regularisasi L2 pada bobot-bobot *trew*. Dimana, apabila nilai *lambda* semakin besar, maka model akan semakin konservatif.
- i) *alpha*: regularisasi L1 pada bobot-bobot *tree*. Dimana, apabila nilai *alpha* semakin besar, maka model akan semakin konservatif.

Pada penelitian ini, *random search* dan algoritma genetika menggunakan *search interval hyperparameter* XGBoost sebagai berikut[12]:

Tabel 2. *Interval Search Hyperparameter* XGBoost

<i>Hyperparameter</i>	Batas Bawah	Batas Atas
<i>n_estimators</i>	900	2000
<i>max_depth</i>	3	9
<i>min_child_weight</i>	1	5
<i>gamma</i>	0.1	0.5
<i>subsample</i>	0.6	1
<i>colsample_bytree</i>	0.6	1
<i>reg_alpha</i>	0.0001	100
<i>reg_lambda</i>	0.05	3
<i>learning_rate</i>	0.01	0.1

6. *Random Search*

Random search adalah metode optimasi *hyperparameter* dengan pencarian kandidat *hyperparameter* secara acak/*random*. Proses pencarian acak kandidat *hyperparameter* akan berhenti ketika perulangan telah mencapai jumlah iterasi yang ditetapkan[17].

7. Algoritma Genetika

Algoritma Genetika merupakan sebuah kelas pencarian dan algoritma optimasi yang terinspirasi dari proses genetika dan seleksi alam. Algoritma ini beroperasi pada prinsip seleksi alam, *crossover*, dan mutasi untuk menyelesaikan permasalahan optimisasi yang kompleks. Prinsip tersebut mendorong keanekaragaman genetik dihasilkan oleh algoritma gentika, hal ini menyebabkan algoritma genetika menjadi efisien untuk melakukan eksplorasi terhadap ruang solusi dan menemukan solusi yang optimal pada variasi dari permasalahan optimasi[18]. Alur kerja algoritma genetika yaitu sebagai berikut[18]:

1. Inisialisasi sebuah populasi yang terdiri dari solusi berpotensi. Solusi yang dimaksud dapat berupa deret angka binary atau struktur data lainnya.

2. Evaluasi dilakukan melalui perhitungan fungsi objektif yang direpresentasikan oleh $f(x) = y$. Proses ini melakukan evaluasi pada setiap performa individu terkait kriteria yang ditentukan. Setiap anggota pada populasi mengalami evaluasi berdasarkan *fitness function* yang menentukan jarak dari solusi yang diberikan dengan solusi optimal pada permasalahan yang akan diselesaikan.
3. Seleksi dilakukan dengan memilih individu tau solusi berdasarkan nilai *fitness* masing-masing. Semakin baik sebuah solusi, maka semakin besar kesempatan untuk terpilih mengikuti reproduksi.
4. Mutasi dilakukan untuk memperkenalkan perubahan acak pada susunan genetic keturunannya (*offspring*) dalam mempertahankan keberagaman genetic pada populasi. Selain itu, mutasi dilakukan untuk mencegah algoritma terjebak pada lokal optimum.
5. Pemeriksaan kriteria terminasi dilakukan untuk menentukan apakah algoritma sudah mencapai kriteria terminasinya atau belum. Jika sudah mencapai kriteria terminasi, maka algoritma genetika berhenti bekerja dan nilai solusi terbaik dikembalikan.

8. Evaluasi

Tahap terakhir dari penelitian ini yaitu evaluasi sebagai alat ukur dalam melakukan perbandingan performa setiap model yang digunakan pada setiap algoritma *machine learning*. Evaluasi dilakukan untuk menghindari model yang kurang tepat untuk digunakan. Pengukuran performa model memerlukan pemilihan penggunaan matriks evaluasi yang tepat seperti *root mean squared error* (RMSE)[19].

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - \hat{p}_i)^2} \quad (3)$$

Keterangan:

n = jumlah data

p_i = nilai aktual pada data ke- i

$\hat{p}_i$ = nilai prediksi pada data ke- i

C. Hasil dan Pembahasan

1. Data Preprocessing

Pada penjelasan sebelumnya, data *preprocessing* yang dilakukan pada penelitian ini yaitu *handling missing value* dan *data encoding*. Hasil dari proses *handling missing value* dapat dilihat pada Tabel 3 di bawah ini.

Tabel 3. Hasil *Handling Missing Value*

Fitur	Metode <i>Handling Missing Value</i>	Jumlah <i>Missing Value</i> (Sebelum)	Jumlah <i>Missing Value</i> (Sesudah)
CompetitionDistance	Median	3	0
CompetitionOpenScienceMonth	Nilai 0	354	0
CompetitionOpenSinceYear	Nilai 0	354	0
Promo2SinceWeek	Nilai 0	544	0
Promo2SinceYear	Nilai 0	544	0
PromoInterval	Nilai 0	544	0
Open	Nilai 1	11	0

Hasil dari proses *data encoding* yang melakukan perubahan nilai berupa huruf menjadi numerik dapat dilihat pada Tabel 4 di bawah ini.

Tabel 4. Hasil *Data Encoding*

Fitur	Nilai Sebelum <i>Encoding</i>	Nilai Sesudah <i>Encoding</i>
Assortment	[a,b,c]	[1,2,3]
StoreType	[a,b,c,d]	[1,2,3,4]
StateHoliday	[a,b,c,0]	[1,2,3,0]

2. *Data Splitting*

Hasil dari pembagian data pada data latih yang sudah ada menjadi data latih baru dan data validasi dengan rasio 70% dan 30% yaitu berjumlah 675,470 dan 168,868 data.

3. *Retail Sales Forecasting Menggunakan XGBoost*

Data yang sudah melalui proses *data preprocessing* dan *data splitting* digunakan untuk melakukan *sales forecasting* terhadap produk barang dari perusahaan *retail* (Rossmann Store) atau dapat disebut *retail sales forecasting* menggunakan XGBoost dengan nilai *hyperparameter default*. Nilai akurasi berdasarkan nilai RMSE sebagai pengukur performa XGBoost dengan *hyperparameter default* dalam melakukan *retail sales forecasting* dapat dilihat pada Tabel 5 di bawah ini yang menunjukkan performa yang baik dengan RMSE mendekati 0.

Tabel 5. Hasil XGBoost *Default Hyperparameter*

Model	RMSE	
	Proses Latih	Proses Uji
XGBoost <i>Default</i>	0.152	0.152

4. *Retail Sales Forecasting Menggunakan XGBoost dan Penyetelan Hyperparameter*

Data yang sudah melalui proses *data preprocessing* dan *data splitting* digunakan untuk melakukan *sales forecasting* terhadap produk barang dari perusahaan *retail* (Rossmann Store) atau dapat disebut *retail sales forecasting* menggunakan XGBoost dengan *random search* dan algoritma genetika dalam penyetelan *hyperparameter* pada XGBoost. Proses ini dapat menghasilkan *hyperparameter* XGBoost yang optimal dan menentukan perbandingan dari performa *random search* dan algoritma

genetika dalam melakukan penyetelan *hyperparameter* pada XGBoost berdasarkan nilai RMSE.

Hyperparameter XGBoost yang optimal sebagai hasil dari penerapan *random search* ditunjukkan pada Tabel 6 dan hasil dari penerapan algoritma genetika ditunjukkan pada Tabel 7 di bawah ini.

Tabel 6. *Hyperparameter* Optimal XGBoost Hasil *Random Search*

<i>Hyperparameter</i>	Nilai
n_estimators	1500
max_depth	6
min_child_weight	5
gamma	0.1
subsample	0.7
colsample_bytree	0.6
reg_alpha	0.1
reg_lambda	2
learning_rate (eta)	0.1

Tabel 7. *Hyperparameter* Optimal XGBoost Hasil Algoritma Genetika

<i>Hyperparameter</i>	Nilai
n_estimators	1500
max_depth	6
min_child_weight	5
gamma	0.1
subsample	0.7
colsample_bytree	0.6
reg_alpha	0.1
reg_lambda	2
learning_rate (eta)	0.1

Nilai akurasi berdasarkan nilai RMSE sebagai pengukur performa *random search* dan algoritma genetika dalam melakukan penyetelan *hyperparameter* pada XGBoost untuk *retail sales forecasting* dapat dilihat pada Tabel 8. Pada tabel tersebut ditunjukkan bahwa nilai RMSE antara proses latih dan proses uji dari penerapan *random search* dan algoritma genetika pada XGBoost tidak jauh berbeda, namun jika dipilih yang paling unggul dalam melakukan penyetelan *hyperparameter* pada XGBoost yaitu *random search* dengan nilai RMSE yang lebih rendah dari algoritma genetika yaitu 0.123 pada proses latih dan 0.122 pada proses uji.

Tabel 8 Perbandingan Nilai RMSE Metode Penyetelan *Hyperparameter XGBoost*

Model	RMSE	
	Proses Latih	Proses Uji
XGBoost – Random Search	0.123	0.122
XGBoost – Algoritma Genetika	0.333	0.332

D. Simpulan

Dari hasil penelitian yang telah dilakukan menggunakan *random search* dan algoritma genetika pada XGBoost untuk melakukan penyetelan *hyperparameter* menggunakan dataset perusahaan *retail* Rossmann Store, dapat disimpulkan bahwa metode *random search* memiliki performa yang lebih baik dari algoritma genetika dengan nilai RMSE pada proses latih dan proses uji sebesar 0.123 dan 0.122. Sementara itu, nilai RMSE algoritma genetika pada proses latih dan proses uji yaitu 0.333 dan 0.332. Perbedaan nilai RMSE tersebut tidak signifikan, artinya baik *random search* maupun algoritma genetika dapat digunakan untuk menjadi solusi dalam menemukan *hyperparameter* yang optimal pada XGBoost untuk melakukan *retail sales forecasting*.

E. Ucapan Terima Kasih

Publikasi ini didukung oleh Universitas Indonesia melalui Hibah Pascasarjana Tahun 2023 No NKB-021/UN2.RST/HKP.05.00/2023.

F. Referensi

- [1] R. F. Ali, A. Muneer, A. Almaghthawi, A. Alghamdi, S. M. Fati, and E. A. Abdullah Ghaleb, 'BMSP-ML: big mart sales prediction using different machine learning techniques', *IJ-AI*, vol. 12, no. 2, p. 874, Jun. 2023.
- [2] A. S. Miha Djami, N. W. Utami, and A. A. I. I. Paramitha, 'The Prediction Of Product Sales Level Using K-Nearest Neighbor and Naive Bayes Algorithms (Case Study : PT Kotamas Bali)', *pilar*, vol. 19, no. 2, pp. 77–84, Sep. 2023.
- [3] S. Raizada and J. R. Saini, 'Comparative Analysis of Supervised Machine Learning Techniques for Sales Forecasting', *IJACSA*, vol. 12, no. 11, 2021.
- [4] E. Sakapurnama and N. Safitri, 'Impact of Pandemic Covid-19 on Consumer Purchase Intention: Empirical Study from Indonesia', *JIKK*, vol. 16, no. 1, pp. 84–94, Jan. 2023.
- [5] T. Boone, R. Ganeshan, A. Jain, and N. R. Sanders, 'Forecasting sales in the supply chain: Consumer analytics in the big data era', *International Journal of Forecasting*, vol. 35, no. 1, pp. 170–180, Jan. 2019.
- [6] X. Bi, G. Adomavicius, W. Li, and A. Qu, 'Improving Sales Forecasting Accuracy: A Tensor Factorization Approach with Demand Awareness', *INFORMS journal on computing*, Nov. 2020, Accessed: Sep. 20, 2023. [Online]. Available: <http://arxiv.org/abs/2011.03452>
- [7] S. Hwang, G. Yoon, E. Baek, and B.-K. Jeon, 'A Sales Forecasting Model for New-Released and Short-Term Product: A Case Study of Mobile Phones', 2023.

- [8] B. Pavlyshenko, 'Machine-Learning Models for Sales Time Series Forecasting', *Data*, vol. 4, no. 1, p. 15, Jan. 2019.
- [9] A. Massaro, A. Panarese, D. Giannone, and A. Galiano, 'Augmented Data and XGBoost Improvement for Sales Forecasting in the Large-Scale Retail Sector', *Applied Sciences*, 2021.
- [10] F. Branda, F. Marozzo, and D. Talia, 'Ticket Sales Prediction and Dynamic Pricing Strategies in Public Transport', *BDCC*, vol. 4, no. 4, p. 36, Nov. 2020.
- [11] A. Anggrawan, H. Hairani, and N. Azmi, 'Prediksi Penjualan Produk Unilever Menggunakan Metode Regresi Linear', *BITE*, vol. 4, no. 2, pp. 123–132, Dec. 2022.
- [12] X. Zhao and P. Keikhosrokiani, 'Sales Prediction and Product Recommendation Model Through User Behavior Analytics', *Computers, Materials & Continua*, vol. 70, no. 2, pp. 3855–3874, 2022.
- [13] D. Tarwidi, S. R. Pudjaprasetya, D. Adytia, and M. Apri, 'An optimized XGBoost-based machine learning method for predicting wave run-up on a sloping beach', *MethodsX*, vol. 10, p. 102119, 2023.
- [14] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, pp. 785–794.
- [15] S. Ji, X. Wang, W. Zhao, and D. Guo, 'An Application of a Three-Stage XGBoost-Based Model to Sales Forecasting of a Cross-Border E-Commerce Enterprise', *Mathematical Problems in Engineering*, vol. 2019, pp. 1–15, Sep. 2019.
- [16] A. Panarese, G. Settanni, V. Vitti, and A. Galiano, 'Developing and Preliminary Testing of a Machine Learning-Based Platform for Sales Forecasting Using a Gradient Boosting Approach', *Applied Sciences*, vol. 12, no. 21, p. 11054, Oct. 2022.
- [17] M. R. Mubarak and R. Herteno, 'Hyper-parameter Tuning Pada XGBoost Untuk Prediksi Keberlangsungan Hidup Pasien Gagal Jantung', vol. 09, 2022.
- [18] A. Mehdary, A. Chehri, A. Jakimi, and R. Saadane, 'Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection', *Sensors*, vol. 24, no. 4, p. 1230, Feb. 2024.
- [19] Z. Xia, S. Xue, L. Wu, J. Sun, Y. Chen, and R. Zhang, 'ForeXGBoost: passenger car sales prediction based on XGBoost', *Distrib Parallel Databases*, vol. 38, no. 3, pp. 713–738, Sep. 2020.