

RESEARCH ARTICLE

Hybrid Recommender System Berbasis Alternating Least Squares dan CatBoost (Studi Kasus Anime pada MyAnimeList)

Abdul Madjid Marwan, Dadang Setiawan* and Muhammad Johan Alibasa

Fakultas Informatika, Universitas Telkom, Bandung, 40257, Jawa Barat, Indonesia

*Corresponding author: dadangset@telkomuniversity.ac.id

Received on 21 April 2024; accepted on 19 May 2024

Abstrak

Platform di internet berurusan dengan data yang sangat besar dan terus berkembang membutuhkan *Recommender System* (RS) untuk memungkinkan penggunaannya menemukan informasi yang relevan dalam banyaknya volume data. RS bekerja dengan melakukan prediksi dan memperlihatkan informasi yang diinginkan oleh pengguna menggunakan berbagai data seperti riwayat ulasan dari pengguna tersebut terhadap suatu item. Seiring dengan perkembangan waktu, teknologi RS menjadi semakin akurat dalam memberikan prediksi. Walaupun hal demikian dapat menciptakan permasalahan baru yaitu kurangnya keragaman dari hasil yang diberikan RS yang berdampak buruk bagi pengguna dan pemilik platform. Penelitian ini merancang Hybrid RS menggunakan algoritma *Alternating Least Squares* (ALS) serta *Categorical Boosting* (CatBoost) untuk memberikan hasil rekomendasi yang memiliki nilai metrik akurasi tinggi serta memeriksa perubahan yang akan terjadi pada nilai metrik keragaman menggunakan studi kasus anime pada MyAnimeList. Hasil akhir dari penelitian akan diuji menggunakan metrik *Root Mean Square Error* (RMSE) untuk mengukur akurasi serta *Cosine Similarity* untuk mengukur keragaman. Hasil akhir RMSE yang didapatkan beserta dengan *Cosine Similarity* dari rekomendasi anime Hybrid RS memiliki nilai yang lebih baik ketika dibandingkan dengan hasil dari algoritma tersendiri.

Key words: *hybrid recommender, alternating least squares, catboost, keragaman, anime*

Pendahuluan

Pendahuluan

Recommender System (RS) merupakan komponen perangkat lunak yang menentukan item yang sebaiknya ditunjukkan kepada pengguna tertentu berdasarkan statistik dan model machine learning [1]. Item merupakan sebutan umum terhadap sesuatu yang ditunjukkan kepada pengguna pada platform atau konteks tertentu. Netflix menunjukkan film kepada pengguna, sedangkan Amazon menunjukkan produk [2]. RS sudah umum digunakan pada berbagai domain di internet, mulai dari e-commerce, sosial media, hingga situs berita. Keberadaan RS memudahkan pengguna platform untuk menemukan informasi yang relevan secara cepat dalam volume informasi yang terus berkembang.

Tujuan utama dari rekomendasi item yang ditunjukkan kepada pengguna adalah untuk berisikan pemahaman akurat mengenai kebutuhan pengguna tersebut yang dapat diukur menggunakan berbagai metrik akurasi. Walaupun akurasi digunakan sebagai metrik utama, seiring waktu dibutuhkan juga metrik lainnya untuk mendapatkan gambaran yang lebih baik mengenai manfaat suatu rekomendasi item. Salah satu metrik lainnya adalah keragaman. Keragaman pada umumnya didefinisikan sebagai variasi yang terdapat dalam rekomendasi.

Selain menunjukkan rekomendasi item sesuai dengan kebutuhan atau preferensi pengguna, penambahan item yang bervariasi dinilai

dapat bermanfaat bagi suatu RS karena kebutuhan atau preferensi pengguna dapat berubah seiring waktu [1]. Pengguna dapat merasa tidak puas apabila sudah mengetahui rekomendasi item dan merasa tidak mendapat manfaat dari rekomendasi tersebut. Pemilik bisnis pada platform juga dapat merasakan kerugian apabila distribusi item kurang adil dalam rekomendasi item kepada pengguna platform [3].

Penelitian ini akan membangun Hybrid RS menggunakan algoritma Alternating Least Squares (ALS) dan Categorical Boosting (CatBoost) untuk menghasilkan rekomendasi item kepada pengguna yang akan diukur menggunakan metrik akurasi Root Mean Squared Error (RMSE) serta metrik keragaman Cosine Similarity. Hasil dari pengukuran akan dibandingkan dengan hasil tersendiri dari algoritma yang membangun Hybrid RS. Penelitian dilakukan untuk meningkatkan hasil dari metrik akurasi dan mengukur perubahan yang terjadi pada metrik keragaman.

MyAnimeList dipilih sebagai platform studi kasus dengan dasar perkembangan stabil yang dialami oleh industri anime pada beberapa tahun terakhir yang juga dapat dilihat dari peningkatan penerapan gaya anime pada iklan di Indonesia [4]. Selain peningkatan industri, anime juga memiliki kemiripan dengan medium film yang sering digunakan pada penelitian RS, contohnya adalah MovieLens [5]. Walaupun memiliki medium yang serupa, tetapi belum terdapat banyak penelitian yang dilakukan menggunakan dataset anime pada konteks RS [6].

Tinjauan Pustaka

Collaborative Filtering

Collaborative Filtering (CF) merupakan suatu tipe RS yang mencocokkan ulasan target pengguna dengan pengguna lainnya mengenai suatu item untuk menemukan neighbors atau kelompok pengguna yang memiliki preferensi serupa. CF memiliki dua cara untuk memperlihatkan hasil akhir dari rekomendasi. Cara pertama memperlihatkan preferensi dari target pengguna melalui nilai numerik, sedangkan cara kedua memperlihatkan preferensi tersebut melalui daftar item [7].

Content-Based

Content-Based (CB) merupakan tipe RS lainnya yang membangun suatu profil pengguna berdasarkan item yang sudah diulas oleh target pengguna tersebut. Sistem akan menggunakan profil pengguna sebagai preferensi yang berisikan fitur dari item yang sudah diulas untuk dicocokkan dengan fitur item lainnya yang serupa dalam proses rekomendasi [8]. Proses rekomendasi yang dilakukan oleh CB tersebut terbagi menjadi 3 yaitu Content Analyzer, Profile Learner, dan Filtering [9].

Hybrid Recommender System

Hybrid merupakan tipe RS yang menggabungkan berbagai tipe RS lainnya seperti CF dan CB untuk menggunakan kelebihan dari suatu tipe dan mengurangi kekurangan dari tipe lainnya. Salah satu penelitian menggabungkan beberapa algoritma RS yang dapat memenuhi kriteria akurasi, kebaruan, dan keragaman lebih baik ketika dibandingkan oleh algoritma masing-masing. Hybrid juga dipilih sebagai tipe RS dalam penelitian ini dengan dasar bahwa belum banyak penelitian yang menggunakan tipe RS tersebut dalam konteks permasalahan keragaman [10]. Penelitian ini akan membangun Hybrid RS menggunakan metode Weighted yang memberikan bobot sesuai terhadap algoritma yang digunakan untuk membangun RS tersebut.

Keragaman pada Recommender System

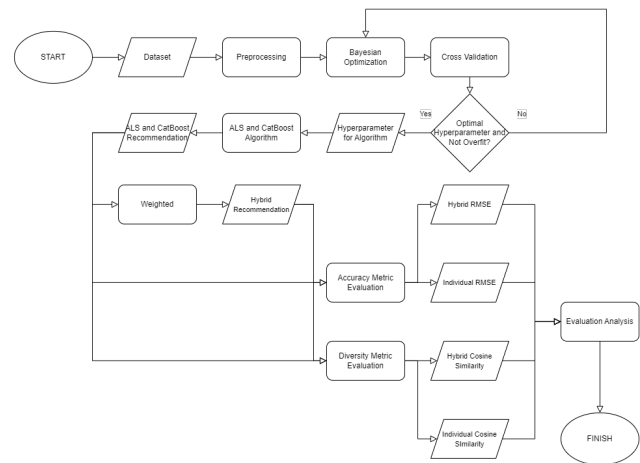
Rekomendasi item membutuhkan tingkat keragaman tertentu untuk memberikan manfaat kepada pengguna karena pada umumnya di awal pengguna menggunakan rekomendasi sebagai alat pencarian untuk menjelajahi item yang baru dan beragam. Keragaman pada penelitian ini adalah rata-rata ketidakmiripan dari cakupan genre yang tidak berulang pada daftar item yang ditunjukkan kepada suatu pengguna. Genre suatu anime dan variasinya akan digunakan untuk membandingkan anime tersebut dengan anime lainnya pada suatu daftar rekomendasi item.

Alternating Least Squares

Alternating Least Squares (ALS) merupakan suatu algoritma Matrix Factorization (MF) yang menginterpretasikan matriks antara data pengguna dan item sebagai suatu matriks yang jarang terisi dan berusaha untuk membangun ulang matriks tersebut dengan dua matriks yang berdimensi lebih kecil. Matriks X dan matriks Y akan dikalikan untuk membangun ulang matriks awal. Proses ini diharapkan mengurangi RMSE di antara ulasan awal yang tersedia pada matriks dengan matriks yang telah dibangun melalui perkalian tersebut [11, 12].

Categorical Boosting

Catboost berbeda dengan algoritma GBDT lain seperti XGBoost dan LightGBM yang memiliki sisi pertumbuhan pohon secara asimetris. Pertumbuhan pohon yang simetris pada CatBoost memiliki beberapa kelebihan seperti dapat mengatasi permasalahan overfitting dan implementasi pada CPUs yang dicapai dengan menggunakan kondisi sama pada setiap percabangan pada simpul [13].



Gambar 1. DataFrame Hybrid Recommender System

Root Mean Squared Error

Root Mean Squared Error (RMSE) merupakan salah satu metrik akurasi yang populer digunakan dalam konteks RS [14]. Akurasi sendiri merupakan metrik utama untuk mengevaluasi performa dari suatu RS menggunakan dataset ulasan.

Cosine Similarity

Cosine Similarity merupakan suatu metrik yang mengukur persamaan dari dua vektor dimensional berdasarkan sudut diantara kedua vektor tersebut. Nilai yang mungkin didapat pada Cosine Similarity berada di jarak antara nol dan satu, dimana nilai mendekati satu menandakan kemiripan yang kuat [15].

Metodologi Penelitian

RS yang akan dibangun pada penelitian ini akan memberikan rekomendasi anime dalam bentuk daftar kepada pengguna acak. Pembangunan membutuhkan beberapa langkah yaitu Preprocessing, Bayesian Optimization, Cross Validation, ALS and CatBoost Algorithm, Weighted, Accuracy Metric Evaluation, Diversity Metric Evaluation, dan *Evaluation Analysis*. Langkah-langkah tersebut beserta alurnya dapat dilihat pada gambar berikut.

Preprocessing

Preprocessing dilakukan untuk memastikan data yang akan digunakan sebagai input sudah memiliki jumlah, struktur, dan format yang sesuai untuk tugas yang ingin dilakukan karena pada umumnya data dari publik memiliki banyak kekurangan seperti data yang tidak konsisten ataupun jumlah dan dimensi yang terlalu besar [16]. Pertama, nama dari kolom akan disesuaikan untuk menyalurkan penulisan pada seluruh kolom dan juga memastikan tidak ada kolom yang memiliki nama sama. Setelah itu, tipe data dari kolom akan diubah agar baris dari kolom dapat digunakan pada pelatihan model ALS dan CatBoost. Terakhir, jumlah dari baris dikurangi untuk mengurangi tingkat komputasi dan waktu yang dibutuhkan oleh RS. Dataset disimpan pada Google Drive dan dipanggil ke dalam program sebagai DataFrame menggunakan library PySpark. Pada tahapan tertentu, struktur DataFrame akan diubah menggunakan library Panda untuk memudahkan pelaksanaan operasi dan fungsi pada DataFrame tersebut. Proses ini juga menggunakan asumsi bahwa pemberian rekomendasi anime terhadap pengguna hanya dilakukan dengan rujukan terhadap ulasan yang sudah dilakukan oleh pengguna tersebut.

Bayesian Optimization

Bayesian Optimization (BO) digunakan untuk mengoptimasi hiperparameter untuk mendapatkan nilai metrik akurasi terbaik dalam iterasi yang sedikit mungkin. Penelitian ini menerapkan BO pada algoritma ALS dan CatBoost untuk mendapatkan nilai hiperparameter terbaik terhadap algoritma secara masing-masing sebelum melakukan pelatihan model RS dan menghasilkan prediksi dan rekomendasi anime. Sebelum menggunakan BO, *objective_function* harus didefinisikan yang berisikan hiperparameter yang ingin dioptimalkan sebagai input beserta dengan metrik evaluasi yang ingin dimaksimalkan atau diminimalisir nilainya sebagai output. Sebagai contoh, BO pada ALS akan menggunakan input berupa *maxIter*, *regParam*, serta *Rank* untuk menghasilkan output berupa RMSE. BO menggunakan *Search Space* sebagai jangkauan untuk mendapatkan sampel yang akan digunakan oleh *objective_function*. Pencarian sampel harus menyeimbangkan antara eksplorasi atau eksploitasi suatu titik di dalam *Search Space* [17, 18].

Cross-Validation

Pada umumnya *paramGrid* menggunakan beberapa nilai dari hiperparameter yang ingin diperiksa tetapi pada penelitian ini ALS akan menerapkan cross-validation pada nilai optimal yang sebelumnya sudah didapatkan pada BO. Proses ini akan menggunakan input berupa nilai hiperparameter yang sebelumnya sudah didapatkan pada proses BO ALS dan CatBoost. CatBoost menggunakan teknik *k-fold* yang membagi data yang digunakan dalam pelatihan model (dalam penelitian ini yaitu *trainData*) menjadi beberapa bagian dan dilanjutkan dengan memeriksa data yang sudah terbagi tersebut. Berbeda dengan validasi pada ALS yang menggunakan data berbeda yaitu dari *validData*. *paramGrid* dalam *GridSearch* juga sedikit berbeda, dimana *paramGrid* berisi informasi lebih dan juga menggunakan jangkauan data.

Alternating Least Squares

ALS akan menggunakan fungsi yang membutuhkan beberapa parameter untuk mendefinisikan matriks yang akan dibangun ataupun sebagai hiperparameter. Selain itu, terdapat juga hiperparameter yang digunakan yaitu *Max Iteration*, *Regularization Parameter*, dan *Rank*. *Max Iteration* merupakan angka maksimal dari iterasi yang akan dilakukan oleh ALS sebelum algoritma tersebut berhenti tanpa melihat apakah sudah mencapai titik konvergen atau belum. *Regularization Parameter* merupakan nilai yang digunakan untuk mengontrol tingkat penalti yang diberikan terhadap *loss function* atau jarak antara nilai prediksi dengan nilai sesungguhnya. *Rank* merupakan nilai yang digunakan sebagai faktor laten dalam pembuatan matriks interaksi antara pengguna dan item. Proses ini akan menggunakan *trainData* untuk pelatihan model ALS dengan nilai hiperparameter yang sudah didapatkan dan diperiksa pada proses sebelumnya sebagai input.

Categorical Boosting

Algoritma CatBoost membutuhkan preprocessing tambahan untuk merubah dataset agar dapat digunakan. DataFrame Panda tersebut akan dibagi menggunakan metode *train.test.split*. Berbeda dengan *randomSplit*, *train.test.split* akan mengacak nilai dari dataset sebelum memisahkan informasi yang berisikan input fitur (sering ditulis sebagai *X*) yang akan digunakan dalam pelatihan model seperti *user_id* dan *anime_id* dengan informasi yang berisikan target (sering ditulis sebagai *y*) yang ingin diprediksi oleh model seperti rating [19]. Berbeda dengan algoritma sebelumnya, fungsi CatBoost hanya membutuhkan masukan dalam bentuk hiperparameter. Pelatihan model menggunakan *datasetTrain* dan *datasetTest* sebagai input yang merupakan *trainData* dan *testData* yang sudah menjalani proses perubahan data tambahan.

Weighted

Weighted menggunakan hasil prediksi dari model ALS dan CatBoost menggunakan *testData* lalu dikombinasikan menjadi satu DataFrame. Weighted pada awalnya memberikan nilai atau bobot yang sama terhadap algoritma yang digunakan, tetapi seiring waktu akan menyesuaikan dengan hasil prediksi dari pengguna [20]. Kedua kolom baru tersebut akan digunakan dalam proses pemberatan. Nilai dari kedua prediksi akan dipindahkan dari bentuk nilai didalam kolom DataFrame menjadi kumpulan nilai pada array. Proses akan menggunakan fungsi *optimize_hybrid_weight* yang mengambil nilai dari kedua array sebagai input dan melakukan kalkulasi dengan berat yang diberikan.

Accuracy Metric Evaluation

Evaluasi terhadap metrik akurasi dilakukan dengan menggunakan daftar rekomendasi anime dan prediksi nilainya sebagai input. Output dari proses ini akan berupa nilai RMSE dari rekomendasi anime tersebut yang didapatkan berdasarkan *testData*. Nilai RMSE yang mendekati 0 memberikan indikasi bahwa prediksi nilai pada rekomendasi anime akurat dengan nilai sesungguhnya dan sebaliknya untuk nilai RMSE yang menjauhi nilai 0.

Diversity Metric Evaluation

Evaluasi terhadap metrik keragaman dilakukan dengan menggunakan daftar rekomendasi anime dan prediksi nilainya sebagai input. Output dari proses ini akan berupa nilai rata-rata Cosine Similarity dari rekomendasi anime tersebut yang didapatkan berdasarkan *testData*. DataFrame yang digunakan pada proses ini akan berbeda dengan tahap sebelumnya, dimana pada tahap ini DataFrame dari hasil rekomendasi akan digabungkan dengan beberapa kolom dari *new.anime* yaitu *name* dan *genres*.

Dengan melakukan evaluasi terhadap kedua metrik tersebut, dapat memberikan gambaran yang lebih komprehensif mengenai kualitas rekomendasi yang diberikan oleh sistem.

Evaluation Analysis

Setelah melakukan evaluasi terhadap metrik akurasi dan keragaman, langkah terakhir adalah melakukan analisis terhadap hasil evaluasi tersebut. Analisis ini bertujuan untuk memahami kinerja sistem secara keseluruhan, mengidentifikasi kekuatan dan kelemahan, serta memberikan rekomendasi untuk perbaikan atau peningkatan di masa mendatang.

Hasil dan Pembahasan

Hasil Hybrid Recommender System

Hybrid RS yang dibangun berdasarkan algoritma ALS dan CatBoost akan memberikan hasil rekomendasi anime dalam bentuk daftar beserta dengan prediksi nilai ulasan. Hasil rekomendasi ditujukan kepada pengguna acak. DataFrame menampilkan nilai prediksi yang didapatkan oleh algoritma tersendiri, algoritma Hybrid melalui proses Weighted, ataupun nilai ulasan sesungguhnya. Hal tersebut dilakukan untuk memberikan gambaran mengenai proses yang terjadi dalam pemberian rekomendasi anime kepada pengguna yang juga berdasarkan nilai-nilai tersebut. Berikut merupakan salah satu contoh dari DataFrame berisikan hasil rekomendasi anime.

Hasil Metrik RMSE

Terdapat beberapa tahapan dalam memilih rentang pada setiap hiperparameter. Rentang dimulai dari angka kecil, dan nilainya ditingkatkan berkala. Apabila suatu rentang mendapatkan nilai RMSE terbaik, maka rentang tersebut disimpan dan pemilihan rentang dilakukan

ALS_Prediction	CatBoost_Prediction	Optimal_Hybrid_Prediction	Actual_Rating
8.442508697509766	8.087662198472	8.300748643952513	9
6.470187664031982	7.011407642412676	6.686403115850265	6
8.28976058959961	8.672618993359707	8.442711173041022	8
5.033612251281738	5.918367084792178	5.387068896461823	6
7.5277910232543945	7.0099194981362585	7.320902772807303	7
7.861484527587891	7.921488268508689	7.885455700329437	8
9.18708610534668	8.477563777136837	8.903634212057291	10
7.334250450134277	8.037643991753253	7.61525396968404	8
6.262392997741699	6.8863039401310635	6.511643219597625	7
6.085735321044922	5.776193491177862	5.9620742779232625	6

Gambar 2. DataFrame Hybrid Recommender System

untuk hyperparameter selanjutnya hingga mendapatkan ketiga nilai. Lalu menggunakan 3 rentang hyperparameter tersebut untuk melakukan pencarian lagi dengan meningkatkan jumlah pengambilan sampel. Pada akhirnya, algoritma ALS mendapatkan nilai hyperparameter optimal yaitu 35 untuk *maxIter*, 0.23966237306286117 untuk *regParam*, dan 3 untuk *Rank* untuk menghasilkan RMSE optimal dengan nilai 1.3193561150603346.

Hasil Metrik Cosine Similarity

Cosine Similarity mengalami proses yang berbeda dengan pencarian nilai RMSE. Model dilatih menggunakan hyperparameter yang sebelumnya sudah didapatkan dari proses BO dan cross-validation. Lalu masing-masing algoritma akan mengambil sampel acak dari prediksi yang akan digunakan untuk pembuatan fitur vektor untuk menilai Cosine Similarity. Apabila sudah didapatkan, prediksi awal algoritma ALS akan digabungkan dengan hasil dari algoritma CatBoost untuk mendapatkan prediksi hybrid. Tanpa menggunakan proses weighted, prediksi hybrid akan diperiksa nilai Cosine Similarity untuk melihat apakah ada perubahan nilai atau tidak. Algoritma ALS menghasilkan nilai Cosine Similarity yaitu 0.30045898541681804 sedangkan algoritma CatBoost menghasilkan nilai 0.32942037138523067. Hasil ideal dari Cosine Similarity adalah nilai yang mendekati nol. Setelah mendapatkan prediksi hybrid yang berisikan prediksi ALS dan CatBoost, nilai Cosine Similarity yang didapatkan adalah 0.2734247481534338.

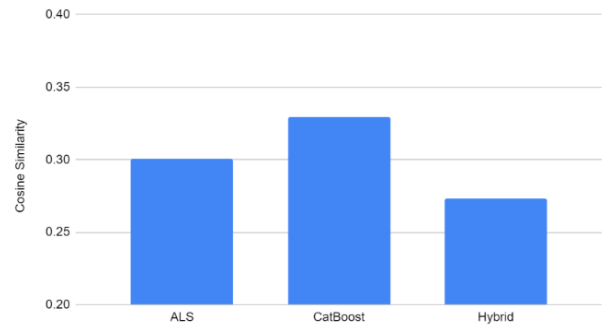
Analisis Hasil Metrik

Evaluasi dari RMSE dan Cosine Similarity memberikan hasil yang serupa, yaitu hasil hybrid yang lebih baik dibandingkan dengan hasil dari algoritma itu tersendiri. Berikut merupakan grafik untuk memperlihatkan perbandingan nilai RMSE dan Cosine Similarity yang didapatkan pada setiap model dan juga pada hybrid RS. Rentang dari nilai diubah untuk dapat memperlihatkan perbedaan pada grafik yang berskala kecil.

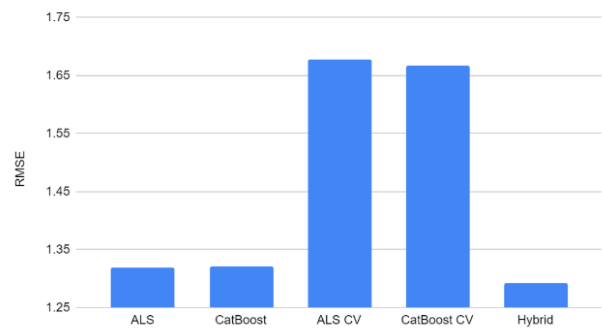
Penerapan Hybrid Recommender System

Penelitian ini menghasilkan hybrid RS yang dapat diterapkan pada suatu platform atau aplikasi melalui integrasi dengan UI/UX dan database. Terdapat beberapa kelebihan dan kekurangan dalam penerapan hybrid RS yang diajukan dalam penelitian ini. Kelebihannya adalah Hybrid RS menggunakan metode yang tidak membutuhkan tingkat komputasional yang tinggi. Hal tersebut akan memudahkan penerapan hybrid RS terhadap suatu platform atau aplikasi yang mungkin dikembangkan selanjutnya. Kekurangan dari hybrid RS yang diajukan adalah hanya menggunakan informasi tertentu untuk menemukan nilai metrik akurasi dan keragaman, padahal bisa menggunakan informasi lebih dari suatu dataset untuk mendapatkan gambaran yang lebih jelas mengenai data tersebut.

Cosine Similarity



RMSE



Gambar 3. Chart Nilai RMSE dan Cosine Similarity

Kesimpulan

Penelitian ini telah membangun hybrid RS dengan metode weighted menggunakan model yang dilatih dengan algoritma ALS dan CatBoost. Data yang digunakan berasal dari Anime Recommendation Database 2020 yang sebelumnya dibersihkan dan disesuaikan formatnya sebelum disimpan menjadi dataset baru dengan nama *new_anime* dan *new_rating*. Dataset lalu dibagi menjadi *trainData* yang akan digunakan untuk melatih model, *validData* untuk mengvalidasi hasil dari pelatihan model menggunakan data baru, dan *testData* untuk menghasilkan rekomendasi anime dan nilai prediksi. Setelah nilai tersebut ditemukan, nilai digunakan dalam cross-validation untuk memastikan bahwa model dapat menghasilkan khususnya nilai metrik akurasi yang baik pada data baru. Setelah mendapatkan perbedaan nilai yang sedikit pada cross-validation, tahap dilanjut dengan melatih model ALS dan CatBoost menggunakan nilai hyperparameter yang sebelumnya sudah didapatkan. Hasil rekomendasi anime dan nilai prediksi menggunakan *testData* pada model yang sudah dilatih oleh *trainData* lalu akan digunakan untuk digabungkan dengan hasil dari model lainnya untuk proses optimasi bobot. Proses weighted menemukan bobot optimal 0.6005032211023404 untuk algoritma ALS dan 0.3994967788976594 untuk algoritma CatBoost untuk mendapatkan nilai metrik akurasi RMSE terbaik dalam iterasi dan kondisi yang sudah ditentukan. Hybrid RS lalu menggunakan model yang sudah dilatih sebelumnya untuk menunjukkan rekomendasi anime dan nilai prediksi yang akan digunakan dalam pemeriksaan Cosine Similarity. Nilai RMSE dan Cosine Similarity dari Hybrid RS mengalami penurunan ketika dibandingkan dengan hasil dari algoritma tersendiri. Hybrid RS dengan proses weighted telah menghasilkan nilai metrik akurasi RMSE yang lebih kecil, dan model yang dilatih dari RS tersebut memberikan nilai metrik keragaman Cosine Similarity yang lebih kecil berdasarkan pengamatan nilai. Hybrid RS memiliki kelebihan dari segi kompleksitas yang lebih sederhana

karena tidak menggunakan nilai kosong pada data yang berpengaruh dalam tingkat komputasional yang lebih rendah dan juga proses pelatihan RS yang lebih cepat. Penelitian selanjutnya dapat dilakukan dengan memilih mesin dengan tingkat komputasional tinggi agar dapat menggunakan metode, algoritma, dan dataset yang lebih kompleks dan banyak serta menyelaraskan proses pencarian metrik RMSE dan Cosine Similarity.

Daftar Pustaka

1. Castells P, Jannach D. *Recommender Systems: A Primer*. Feb. 2023.
2. Ricci F, Rokach L, Shapira B. *Recommender Systems: Techniques, Applications, and Challenges*. In: *Recommender Systems Handbook*. New York, NY: Springer US; 2022. p. 1-35.
3. Mansoury M, Abdollahpouri H, Pechenizkiy M, Mobasher B, Burke R. FairMatch: A Graph-based Approach for Improving Aggregate Diversity in Recommender Systems. In: *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization*. New York, NY, USA: ACM; 2020. p. 154-62.
4. Parrot Analytics. *Growing Global Demand for Anime*; 2021. (accessed May 04, 2023). <https://www.parrotanalytics.com/insights/growing-global-demand-for-anime-aug-2021/>.
5. Tahmasebi F, Meghdadi M, Ahmadian S, Valiollahi K. A hybrid recommendation system based on profile expansion technique to alleviate cold start problem. *Multimed Tools Appl*. 2021 Jan;80(2):2339-54.
6. Valdivieso H. *Anime Recommendation Database 2020*; 2020.
7. Kumar PP, Vairachilai S, Potluri S, Mohanty SN. *Recommender Systems: Algorithms and Applications*. New York: CRC Press; 2021.
8. Mohanty SN, Chatterjee JM, Jain S, Elngar AA, Gupta P. *Recommender System with Machine Learning and Artificial Intelligence: Practical Tools and Applications in Medical, Agricultural and Other Industries*. Wiley; 2020. Available from: <https://books.google.co.id/books?id=36TqDwAAQBAJ>.
9. Koene A, et al. *Ethics of Personalized Information Filtering*; 2015. p. 123-32.
10. Çano E, Morisio M. *Hybrid Recommender Systems: A Systematic Literature Review*. 2019 Jan.
11. Dangeti P. *Statistics for Machine Learning*. Packt Publishing; 2017. Available from: <https://books.google.co.id/books?id=C-dDDwAAQBAJ>.
12. Pilászy I, Tikk D, Takács G, Németh B, Zibriczky D. *Recommender systems and methods using modified alternating least squares algorithm*; 2014. .
13. Owen L. *Hyperparameter Tuning with Python: Boost your machine learning model's performance via hyperparameter tuning*. Packt Publishing; 2022. Available from: <https://books.google.co.id/books?id=CqF-EAAAQBAJ>.
14. Ricci F, Rokach L, Shapira B, Kantor P. *Recommender Systems Handbook*; 2011.
15. Jannach D, Zanker M, Felfernig A, Friedrich G. *Recommender Systems: An Introduction*. Cambridge University Press; 2010. Available from: <https://books.google.co.id/books?id=eygTJBd.U2cC>.
16. García S, Luengo J, Herrera F. *Data Preprocessing in Data Mining. Intelligent Systems Reference Library*. Springer International Publishing; 2014. Available from: <https://books.google.co.id/books?id=SbFkBAAAQBAJ>.
17. Brochu E, Cora VM, de Freitas N. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*; 2010. .
18. *Bayesian Optimization for hyperparameter Tuning in Random Forests*; 2013. Available from: <https://api.semanticscholar.org/CorpusID:53647007>.
19. Müller AC, Guido S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media; 2016. Available from: <https://books.google.co.id/books?id=1-41DQAAQBAJ>.
20. Sevaslidou J, Eugenia P. *A novel approach for hybrid recommendation systems*. 2021.
21. Kunaver M, Požrl T. *Diversity in recommender systems – A survey*. *Knowl Based Syst*. 2017;123:154-62.