

Classification of Software Requirements

Reza Ali Nirwansyah¹, Aiza Fuji Sari², Andrya Muhammad Naufal³, Cornelia Andaru⁴

*Teknik Komputer, Institut Teknologi Sepuluh Nopember
Kampus ITS Sukolilo, Surabaya 60111.*

¹rezaalinirwansyah@gmail.com

²aizafuji@gmail.com

³andryanaufal@gmail.com

⁴cornelia.andaru@gmail.com

Abstrak— Machine learning telah menjadi pendekatan yang efektif dalam pengklasifikasian kebutuhan perangkat lunak untuk meningkatkan proses pengembangan perangkat lunak. Dalam konteks ini, penelitian ini fokus pada penerapan teknik machine learning untuk klasifikasi kebutuhan perangkat lunak. Metode ini melibatkan penggunaan algoritma pembelajaran mesin yang dikembangkan untuk mengenali pola dan relasi dalam kumpulan data kebutuhan perangkat lunak. Setelah mengumpulkan data kebutuhan perangkat lunak, langkah pertama melibatkan normalisasi teks menggunakan library NLTK. Normalisasi ini mencakup transformasi teks menjadi huruf kecil, penghilangan karakter khusus dan angka, tokenisasi, penghapusan kata stopwords, dan konversi ke kata dasar. Selanjutnya, vektorisasi teks dilakukan menggunakan teknik TF-IDF, mengubah teks menjadi representasi numerik untuk memfasilitasi pemrosesan oleh algoritma machine learning. Berbagai algoritma machine learning, seperti linear regression, naive bayes, decision tree, MLP, SVM kernel linear, dan SVC, diterapkan untuk klasifikasi kebutuhan perangkat lunak. Hasil eksperimen ini adalah memahami performa dari masing-masing metode dan menentukan metode dengan hasil algoritma terbaik.

Kata kunci— Machine Learning, Software Requirement, Algoritma

I. PENDAHULUAN

Pengembangan perangkat lunak merupakan tantangan kompleks yang memerlukan pemahaman mendalam tentang kebutuhan pengguna. Salah satu langkah penting dalam pengembangan perangkat lunak adalah mengidentifikasi dan mengkategorikan kebutuhan perangkat lunak. Dengan meningkatnya kompleksitas proyek dan beragamnya kebutuhan pengguna, kebutuhan akan pendekatan yang lebih canggih dan efisien dalam mengelola persyaratan perangkat lunak menjadi semakin penting.

Dalam beberapa tahun terakhir, machine learning telah muncul sebagai alat yang sangat berguna untuk mengatasi tantangan ini. Penerapan teknik machine learning pada klasifikasi kebutuhan perangkat lunak diharapkan dapat meningkatkan efisiensi proses identifikasi dan pengelompokan kebutuhan, sehingga mempercepat siklus pengembangan perangkat lunak.

Pendekatan ini memanfaatkan kekuatan algoritma pembelajaran mesin untuk mengenali pola kompleks dan hubungan antar persyaratan, sehingga memungkinkan sistem secara otomatis mengkategorikan dan mengatur persyaratan tertentu.

Dalam konteks ini, penelitian ini berfokus pada eksplorasi dan penerapan teknik pembelajaran mesin untuk mengklasifikasikan kebutuhan perangkat lunak. Tujuannya adalah untuk menemukan solusi yang dapat meningkatkan akurasi dan kecepatan manajemen kebutuhan perangkat lunak dengan menggunakan algoritma pembelajaran mesin.

Penelitian ini diharapkan dapat memberikan manfaat terhadap pengembangan perangkat lunak modern dengan menghadirkan pendekatan inovatif dalam pengelolaan kebutuhan, yang dapat mengoptimalkan proses pengembangan dan menghasilkan produk yang lebih responsif terhadap kebutuhan pengguna. Selain itu, penggunaan pembelajaran mesin dalam konteks ini juga dapat membuka pintu untuk penelitian lebih lanjut dalam menghadapi dinamika perubahan kebutuhan perangkat lunak yang terus berkembang.

II. DASAR TEORI

A. Software Requirements

Requirement merupakan pandangan pengguna terhadap sistem eksternal dan pandangan pengembang terhadap karakteristik internal [1]. Software requirement dapat diartikan sebagai hal-hal yang diharapkan dari suatu perangkat lunak yang akan dikembangkan. Berikut jenis-jenis dari software requirement.

i. Functional Requirements

Functional Requirements berperan dalam menentukan fungsi yang harus bisa dilakukan oleh sistem ataupun komponen sistem dalam perangkat lunak [2].

ii. Non-Functional Requirements.

Non-Functional Requirements adalah requirement yang berkaitan dengan kinerja, keandalan, keamanan,

pemeliharaan, akurasi, penanganan kesalahan, kapasitas, dan sejenisnya [2].

B. Normalisasi Teks

Normalisasi Teks merupakan komponen penting untuk aplikasi pemrosesan bahasa dan pengucapan. Persyaratan dalam normalisasi teks sangat bervariasi tergantung apa yang dibutuhkan [4]. Beberapa teknik yang termasuk bagian normalisasi adalah tokenisasi teks, merubah semua huruf menjadi huruf kecil, menghapus kata-kata yang tidak relevan, dan melakukan pengubahan kata menjadi bentuk dasar. Tokenisasi merupakan proses untuk membagi teks menjadi daftar token, dimana daftar token berupa kata-kata individual [5].

C. Vektorisasi Teks

Dalam analisis memahami suatu bahasa atau text perlu dilakukan vektorisasi sebagai langkah awal. Vektorisasi merupakan proses mengkodekan dokumen atau teks ke dalam domain angka [5]. Terdapat beberapa teknik untuk melakukan vektorisasi, dalam hal ini menggunakan metode TF-IDF.

Term Frequency—Inverse Document Frequency (TF-IDF) merupakan salah satu algoritma pengukuran kesamaan teks dan menggambarkan teks menjadi vektor yang terdiri dari n kata kunci yang diboboti dan muncul dalam teks meliputi pengamatan empiris [6]. TF-IDF menggabungkan dua metrik yaitu nilai frekuensi mentah dari sebuah istilah dalam dokumen atau teks tertentu dan invers frekuensi dokumen atau teks [5].

D. Machine Learning Algorithms

Mesin learning merupakan ilmu pemrograman komputer sehingga mereka dapat belajar dari data yang ada. Pada dasarnya mesin learning melibatkan pembuatan model matematika dalam memahami data [3].

III. METODE

Pada pembahasan ini akan membandingkan metode algoritma mesin learning mana yang memiliki performa terbaik dalam melakukan klasifikasi pada Software Requirements. Adapun metode - metode algoritma mesin learning yang digunakan adalah

i. Linear Regression

Linear Regression adalah salah satu metode untuk memodelkan hubungan antara satu variabel terikat dengan satu atau lebih variabel bebas. Tujuan penggunaan Linear Regression adalah untuk menemukan *the best-fitting line* untuk mendeskripsikan data. Metode ini biasanya digunakan untuk memprediksi hasil dan memahami hubungan antar variabel. Bentuk dasar Linear Regression

dituliskan dalam persamaan $Y = a + bX + e$, dimana Y adalah variabel terikat, X adalah variabel bebas, a adalah titik potongnya, b adalah kemiringan, dan e adalah suku kesalahan [7].

ii. DecisionTree

Decision Tree merupakan salah satu metode *machine learning* yang digunakan untuk tugas klasifikasi dan regresi. Metode ini membagi data menjadi beberapa subset berdasarkan atribut paling signifikan di setiap node. Kelebihan dari Decision Tree sendiri adalah mudah diinterpretasikan dan dapat menangani data numerik dan kategorikal. Namun, metode ini juga rentan terhadap overfitting, yang dapat diatasi dengan menggunakan teknik seperti pemangkasan dan metode lain seperti *random forests* [8].

iii. Naive Bayes

Naive Bayes adalah metode klasifikasi probabilistik berdasarkan teorema Bayes dengan asumsi independensi antar fitur yang “naif”. Metode ini biasanya digunakan untuk klasifikasi teks, pemfilteran spam, dan sistem rekomendasi. Meskipun sederhana, Naive Bayes seringkali memiliki kinerja yang baik dalam praktiknya dan efisien untuk data berskala besar. [8][9].

iv. Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP) adalah jenis jaringan saraf tiruan yang dicirikan oleh beberapa lapisan node, termasuk masukan, keluaran, dan lapisan tersembunyi. Metode ini digunakan untuk berbagai tugas pembelajaran mesin, termasuk klasifikasi, regresi, dan pengenalan pola. Multilayer Perceptron (MLP) mampu mempelajari hubungan kompleks dalam data dan dikenal karena kemampuannya memperkirakan berbagai fungsi. Metode ini dilatih menggunakan propagasi mundur dan dapat memanfaatkan teknik seperti regularisasi dan dropout untuk mencegah *overfitting* [8].

v. Support Vector Machine (SVM)

Support Vector Machines (SVM) merupakan salah satu jenis model yang digunakan untuk klasifikasi dan analisis regresi. Cara kerja SVM adalah dengan menemukan *hyperplane* yang terbaik dan memisahkan titik data ke dalam kelas yang berbeda. SVM banyak digunakan untuk *image recognition*, *text classification*, dan *bioinformatics* [8][10].

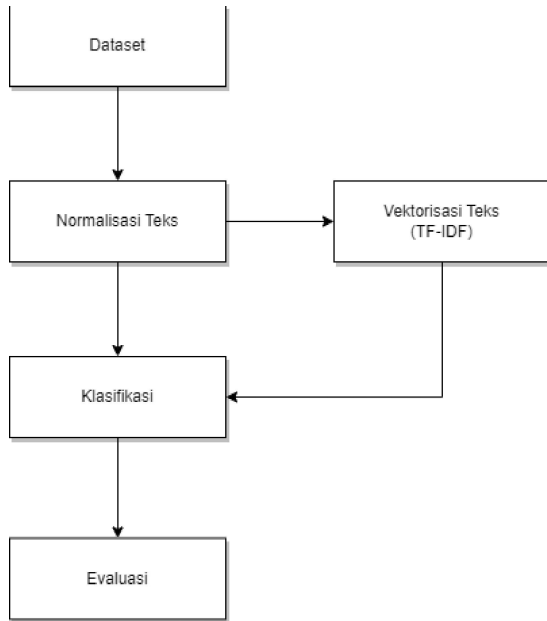
vi. Random Forest

Random Forest adalah metode pembelajaran yang beroperasi dengan membangun banyak pohon keputusan pada waktu pelatihan dan mengeluarkan kelas yang merupakan mode kelas (klasifikasi) atau prediksi rata-rata (regresi) dari masing-masing pohon.

Random forest memitigasi kecenderungan overfitting pada masing-masing pohon keputusan dan tahan terhadap outlier dan noise dalam data. Metode ini banyak digunakan untuk klasifikasi, regresi, dan analisis kepentingan fitur di berbagai domain [8][10]

IV. PROSES EKSPERIMEN

Proses yang dilakukan untuk mencapai hasil performa terbaik dari tiap metode mesin learning seperti yang terlihat pada Gambar 1.



Gambar. 1 Proses Eksperimen Klasifikasi Software Requirement

Pada pembahasan ini menggunakan bahasa pemrograman Python dan dataset dengan format CSV. Kami menggunakan dua buah data set yaitu binary.csv dan multilabel.csv. Pada dataset binary hanya terdapat dua label yaitu 'FR' dan 'NFR', sedangkan pada multilabel terdapat label 'F' dan 'A, L, LF, MN, O, PE, SC, SE, US, FT, PO'.

Langkah selanjutnya yang dilakukan setelah mendapatkan data set adalah melakukan normalisasi text terhadap dataset atau pembersihan dataset. Normalisasi text dengan menggunakan library NLTK. Normalisasi yang dilakukan adalah mengubah teks menjadi huruf kecil, menghilangkan karakter khusus dan angka, melakukan tokenisasi, menghapus kata stopwords dan mengubah menjadi kata dasar.

Setelah melakukan normalisasi teks selanjutnya akan dilakukan vektorisasi teks dengan menggunakan teknik TF - IDF. Vektorisasi teks dibutuhkan agar teks berubah ke domain angka dan dapat diproses oleh algoritma mesin learning.

Langkah selanjutnya adalah melakukan klasifikasi dengan menggunakan algoritma mesin learning. Klasifikasi yang dilakukan untuk mengetahui sebuah requirement termasuk ke dalam label apa. Algoritma mesin learning yang kami gunakan adalah linear regression, naive bayes, decision tree, MLP, SVM kernel linear, dan SVC.

Dari semua metode algoritma mesin learning yang digunakan akan didapatkan nilai dari performa setiap metode yang digunakan. Dari nilai performa yang didapatkan akan terlihat metode mesin learning terbaik dalam melakukan klasifikasi.

V. ANALISIS HASIL

Dalam bagian ini, kami menyajikan hasil eksperimen. Kami melakukan 6 percobaan untuk menentukan model pembelajaran mesin terbaik antara SVM, DT, LR, NB, MP, RF.

Berikut ini merupakan Data yang diperoleh.

Binary Classification			
	Precision	recall	f1-score
LR	0.92	0.53	0.52
DT	0.85	0.85	0.85
NB	0.91	0.70	0.74
MLP	0.90	0.85	0.87
SVM	0.83	0.63	0.66
RF	0.87	0.60	0.59

Berikut ini merupakan data hasil eksperimen untuk klasifikasi Multi Label

Multi Label Classification			
	Precision	recall	f1-score
LR	0.46	0.24	0.28
DT	0.40	0.32	0.34
NB	0.19	0.13	0.11

MLP	0.57	0.46	0.46
SVM	0.42	0.21	0.23
RF	0.44	0.27	0.50

Pada data tersebut terdapat beberapa istilah, yaitu Precision, recall, dan f1-score. Presisi merupakan seberapa banyak dari data yang diklasifikasi sebagai positif oleh model yang sebenarnya merupakan positif. atau sederhanya, bagaimana model dapat benar benar memprediksi suatu kelas.

Recall atau bisa disebut sensitivitas atau true positive rate adalah metrik evaluasi untuk mengukur seberapa banyak dari instance yang sebenarnya positif yang berhasil diidentifikasi oleh model. Singkatnya, recall mengukur kemampuan model untuk menemukan semua instance positif yang ada

F1-score adalah metrik evaluasi yang menggabungkan presisi (precision) dan recall untuk memberikan ukuran keseluruhan kinerja model klasifikasi. F1-score adalah rata-rata harmonik dari presisi dan recall, dan memberikan nilai tunggal yang mencakup informasi tentang sejauh mana model dapat melakukan klasifikasi yang baik pada kelas tertentu.

VI. PEMBAHASAN

Dalam penelitian ini, kami telah melakukan serangkaian eksperimen untuk mengevaluasi performa enam model pembelajaran mesin yang berbeda dalam dua jenis klasifikasi, yaitu Binary dan Multi Label Classification. Model yang diuji melibatkan Support Vector Machine (SVM), Decision Tree (DT), Logistic Regression (LR), Naive Bayes (NB), Multi-layer Perceptron (MLP), dan Random Forest (RF). Evaluasi dilakukan dengan menggunakan tiga metrik utama, yaitu Precision, Recall, dan F1-Score.

Pertama-tama, pada klasifikasi Binary, hasil eksperimen menunjukkan perbedaan yang signifikan antara model-model tersebut. Logistic Regression (LR) menunjukkan nilai tertinggi pada Precision (0.92), yang mengindikasikan kemampuannya dalam mengklasifikasikan dengan akurat seberapa banyak data yang benar-benar positif. Namun, perlu diperhatikan bahwa Recall-nya (0.53) tidak sebaik model Decision Tree (DT) yang mencapai 0.85. Model DT menunjukkan keseimbangan yang baik antara Precision dan Recall, dengan F1-Score mencapai 0.85, menandakan kinerja keseluruhan yang baik.

Pada sisi lain, Naive Bayes (NB) dan Multi-layer Perceptron (MLP) juga menunjukkan hasil yang cukup

baik dalam Binary Classification, dengan NB memiliki nilai Precision sebesar 0.91 dan MLP mencapai 0.90. Namun, perlu dicatat bahwa F1-Score MLP (0.87) lebih unggul daripada NB (0.74), menunjukkan bahwa MLP mungkin lebih cocok untuk situasi di mana seimbang antara Precision dan Recall diinginkan.

Secara keseluruhan, hasil SVM dan RF pada Binary Classification menunjukkan nilai yang sedikit lebih rendah dibandingkan model-model lainnya, dengan SVM memiliki Precision 0.83 dan RF dengan 0.87. Keduanya menunjukkan kinerja yang lebih rendah dalam mengklasifikasikan data positif secara akurat.

Selanjutnya, pada klasifikasi Multi Label, perbandingan model-model menunjukkan variasi performa yang berbeda. MLP muncul sebagai model terbaik dengan nilai Precision, Recall, dan F1-Score berturut-turut sebesar 0.57, 0.46, dan 0.46. Namun, perlu dicatat bahwa nilai Recall MLP yang lebih rendah menunjukkan adanya tantangan dalam mengidentifikasi sebagian besar data yang benar-benar positif.

Beberapa faktor yang menyebabkan data pada hasil klasifikasi Multi Label tidak menyentuh angka diatas 0.8 adalah Perubahan Skala: Dalam tugas multilabel, jumlah kelas yang mungkin lebih banyak, dan variasi jumlah label yang dimiliki setiap sampel dapat membuat nilai presisi menjadi lebih jomplang. Dalam tugas biner, terdapat dua kelas saja, sementara dalam multilabel, kelas-kelas yang berbeda bisa memiliki frekuensi yang berbeda-beda.

Tidak Seimbangnya Label: Beberapa label mungkin lebih dominan daripada yang lain, dan ini dapat mempengaruhi metrik presisi secara signifikan. Label-label yang lebih umum mungkin memiliki dampak yang lebih besar pada nilai presisi secara keseluruhan.

Kemungkinan Multi Output: Pada beberapa tugas multilabel, terdapat lebih dari satu output yang harus diprediksi, dan keterkaitan antara output-output tersebut dapat mempengaruhi presisi secara keseluruhan.

Perbedaan dalam Interpretasi: Presisi pada tugas multilabel dapat memiliki interpretasi yang lebih kompleks, karena mungkin ada variasi dalam cara masing-masing label dipertimbangkan sebagai benar atau salah.

VII. KESIMPULAN

Dari hasil ini, dapat disimpulkan bahwa pemilihan model tergantung pada tujuan spesifik dari tugas klasifikasi. Jika fokus pada akurasi dalam mengidentifikasi data positif, Logistic Regression dan

Decision Tree dapat dianggap sebagai pilihan yang baik dalam Binary Classification, sementara untuk Multi Label Classification, MLP mungkin menjadi pilihan yang lebih unggul meskipun perlu memperhatikan keseimbangan antara Precision dan Recall.

REFERENSI

- [1] Wiegers, K. E., Beatty, J. *Software Requirements*. Pearson Education. 2013.
- [2] Brackett, J. W. *Software Requirements*. Software Engineering Institute USA. 1990.
- [3] VanderPlas, J. *Python Data Science Handbook*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016
- [4] Sarkar, D. *Text Analytics with Python*; Apress: New York, NY, USA, 2016.
- [5] Dias Canedo, E., Cordeiro Mendes, B. "Software requirements classification using machine learning algorithms." *Entropy*, 22(9), 1057. MDPI. 2020.
- [6] Zhang, L., Jiang, Y., Sun, L. "An improved TF-IDF text clustering method." *Journal of Jilin University (Science Edition)*, vol. 59, no. 5, pp. 1200–1204. 2021.
- [7] Dietterich, T. G. "What Is Machine Learning?" *Machine Learning and AI for Healthcare*. 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6134427>
- [8] Pranckevičius, T., Marcinkevičius, V. "Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification." *Balt. J. Mod. Comput.*, 5. 2017. [Online]. Available: <https://api.semanticscholar.org/CorpusID:37013688>
- [9] Hoti, A., Hoti, M. H., Hoti, H., Salihu, A. "Identifying Fake News written on Albanian language in social media using Naive Bayes, SVM, Logistic Regression, Decision Tree and Random Forest algorithms." *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 1-6. 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:249928788>
- [10] Das, S., Bhattacharyya, K., Sarkar, S. "Performance Analysis of Logistic Regression, Naive Bayes, KNN, Decision Tree, Random Forest and SVM on Hate Speech Detection from Twitter." *International Research Journal of Innovations in Engineering and Technology*. 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:262185271>