BAB II

TINJAUAN PUSTAKA

BAB II

TINJAUAN PUSTAKA

II.1 Machine Learning Dalam Industri Otomotif

Machine learning telah mengubah lanskap industri otomotif dengan memberikan solusi yang inovatif untuk berbagai tantangan yang dihadapi oleh pemain di industri ini (Theissler, Pérez-Velázquez, Kettelgerdes, & Elger, 2021). Dalam konteks penilaian kendaraan bekas, teknologi machine learning digunakan untuk membuat model prediksi yang mampu menilai nilai dan kelayakan kendaraan secara akurat berdasarkan berbagai faktor seperti usia, kondisi fisik, riwayat perawatan, dan karakteristik lainnya (Lin & Hashim, 2023). Model ini membantu dalam menentukan harga yang wajar dan kompetitif untuk kendaraan bekas, baik bagi penjual maupun pembeli (Dutulescu et al., 2023).

Machine learning juga digunakan dalam prediksi harga kendaraan, di mana algoritma dan model prediktif diterapkan untuk memperkirakan harga jual suatu kendaraan berdasarkan tren pasar, faktor ekonomi, dan karakteristik kendaraan itu sendiri (Kashyap & Singh, 2022). Prediksi harga yang akurat sangat penting dalam memastikan keberhasilan transaksi jual-beli kendaraan, baik secara individu maupun dalam skala industri (Nandan & Ghosh, 2023).

Teknologi machine learning juga digunakan dalam analisis pasar otomotif untuk memahami tren konsumen, preferensi pasar, dan perilaku

pembeli (Chen, Dewi, Huang, & Caraka, 2020). Dengan memanfaatkan data besar (big data) yang dihasilkan dari berbagai sumber seperti platform online, media sosial, dan sensor kendaraan, analisis machine learning dapat menghasilkan wawasan yang berharga bagi produsen, dealer, dan penyedia layanan otomotif dalam mengambil keputusan strategis seperti pengembangan produk, pemasaran, dan manajemen rantai pasok (Maheshwari, Gautam, & Jaggi, 2021).

Secara keseluruhan, penggunaan teknologi machine learning dalam industri otomotif telah membawa dampak yang signifikan dalam meningkatkan efisiensi, akurasi, dan keunggulan kompetitif, serta membuka peluang baru untuk inovasi dan pengembangan di masa depan (Bertolini, Mezzogori, Neroni, & Zammori, 2021).

II.2 Proses Lelang Kendaraan Bekas

Proses lelang kendaraan bekas merupakan proses yang kompleks yang melibatkan berbagai tahapan, faktor, dan tantangan yang mempengaruhi hasil akhir lelang. Tahapan utama dalam proses lelang kendaraan bekas meliputi (Baumann, Zavolokina, & Schwabe, 2021):

1. Pendaftaran Kendaraan

   Pada tahap ini, pemilik kendaraan mendaftarkan kendaraannya untuk dilelang. Informasi yang diperlukan biasanya mencakup detail kendaraan seperti merek, model, tahun pembuatan, kondisi, dan dokumen kepemilikan.

2. Pemeriksaan dan Penilaian

   Kendaraan yang didaftarkan kemudian akan diperiksa secara fisik oleh petugas lelang untuk menilai kondisi kendaraan, termasuk keausan, kerusakan, dan kekurangan lainnya. Penilaian ini penting untuk menentukan nilai awal kendaraan dan memastikan keaslian serta kelayakan untuk dilelang.

3. Pendaftaran Peserta Lelang

   Calon pembeli atau penawar yang ingin mengikuti lelang harus mendaftar dan mendapatkan nomor peserta lelang. Langkah ini merupakan persyaratan standar untuk memastikan transaksi lelang berjalan secara teratur dan terorganisir.

4. Penawaran dan Penetapan Harga

   Proses lelang dimulai dengan penawaran awal, di mana harga awal kendaraan ditetapkan dan para peserta lelang mulai mengajukan penawaran. Penawaran berlangsung hingga penawar tertinggi ditentukan, dan kendaraan dijual kepada penawar dengan penawaran tertinggi.

5. Pembayaran dan Pengambilan Kendaraan

   Setelah lelang selesai, pembeli yang menang harus segera membayar kendaraan dan melakukan proses administrasi pembelian. Setelah pembayaran dilakukan, pembeli dapat mengambil kendaraan yang dibelinya.

II.3 Support Vector Machine

Support Vector Machine (SVM) adalah salah satu algoritma pembelajaran mesin yang digunakan untuk pemisahan kelas atau regresi. Tujuannya adalah untuk menemukan hyperplane terbaik yang memisahkan dua kelas dalam ruang fitur yang diberikan. Hyperplane adalah sebuah konsep dalam ruang multidimensional yang dapat digunakan untuk melakukan pemisahan antara dua kelas data (Triandi, Mawengkang, & Efendi, 2022).

Konsep dasar dari SVM adalah mencari hyperplane yang memiliki margin terbesar, yaitu jarak antara hyperplane dan titik-titik terdekat dari kedua kelas data (yang disebut support vectors). Margin yang lebih besar menunjukkan bahwa model SVM lebih dapat dipercaya dan lebih mampu menggeneralisasi pola dari data yang diberikan (Fahmi Limas, Rosnelly, & Nursie, 2023).

SVM dapat digunakan untuk pemisahan kelas linear dan non-linear, tergantung pada jenis kernel yang digunakan. Kernel adalah fungsi yang digunakan untuk mengubah data ke ruang fitur yang lebih tinggi dimensi, di mana data menjadi lebih mudah dipisahkan secara linear. Kernel yang umum digunakan termasuk kernel linier, kernel polinomial, kernel radial basis function (RBF), dan kernel sigmoid (ALTAN & KARASU, 2019).

SVM memiliki beberapa keunggulan, seperti kemampuannya untuk menangani data berdimensi tinggi dan kemampuannya untuk menangani

jumlah fitur yang lebih besar dari jumlah sampel data. Namun, SVM juga memiliki beberapa kelemahan, seperti sensitivitas terhadap skala data dan kebutuhan akan parameter yang tepat (seperti parameter C untuk penalti kesalahan klasifikasi dan parameter gamma untuk kernel RBF) (Gaye, Zhang, & Wulamu, 2021).

Secara keseluruhan, SVM merupakan algoritma pembelajaran mesin yang kuat dan serbaguna, yang telah digunakan dalam berbagai aplikasi seperti klasifikasi, regresi, deteksi anomali, dan pemrosesan citra.

II.3.1 Proses Pelatihan SVM

Proses pelatihan Support Vector Machine (SVM) melibatkan beberapa langkah utama, yang melibatkan penentuan hyperplane terbaik untuk memisahkan kelas data. Berikut adalah langkah-langkah umum dalam proses pelatihan SVM (Nalepa & Kawulok, 2019):

1. Pemilihan Kernel

   Langkah pertama adalah memilih kernel yang sesuai dengan dataset. Kernel ini akan mengubah data ke dalam ruang fitur yang lebih tinggi dimensi, di mana data menjadi lebih mudah dipisahkan secara linear.

2. Pembentukan Model

   Setelah memilih kernel, model SVM dibentuk dengan menentukan hyperplane yang terbaik memisahkan kelas data. Hyperplane ini dipilih sedemikian rupa sehingga memiliki margin terbesar, yaitu jarak antara hyperplane dan titik-titik terdekat dari kedua kelas data (support

vectors).

3. Penyesuaian Parameter

   Beberapa parameter SVM perlu disesuaikan selama proses pelatihan. Parameter utama termasuk parameter C untuk penalti kesalahan klasifikasi dan parameter gamma untuk kernel non-linier seperti RBF. Penyesuaian parameter ini dapat dilakukan menggunakan teknik penyetelan parameter seperti cross-validation.

4. Optimasi Margin

   Setelah hyperplane awal dibentuk, proses optimasi dilakukan untuk memaksimalkan margin antara hyperplane dan support vectors. Hal ini dilakukan dengan meminimalkan fungsi tujuan yang mencakup margin dan penalti untuk kesalahan klasifikasi.

5. Penentuan Hyperplane

   Setelah proses optimasi selesai, hyperplane terakhir ditentukan. Hyperplane ini merupakan solusi akhir yang memisahkan dua kelas data dengan margin terbesar.

6. Evaluasi Model

   Setelah pelatihan selesai, model SVM dievaluasi menggunakan data pengujian yang independen. Evaluasi dilakukan untuk mengukur kinerja model dalam melakukan klasifikasi atau regresi, menggunakan metrik seperti akurasi, presisi, recall, atau metrik evaluasi lainnya yang sesuai dengan jenis masalah yang dihadapi.

Proses pelatihan SVM ini berulang dan iteratif, dengan penyesuaian

parameter dan optimasi yang terus-menerus untuk memperbaiki kinerja model. Kesuksesan proses pelatihan sangat tergantung pada pemilihan kernel yang tepat, penyetelan parameter yang akurat, dan kualitas data yang digunakan.

II.3.2 Proses Klasifikasi SVM

Proses klasifikasi menggunakan algoritma Support Vector Machine (SVM) melibatkan langkah-langkah berikut (Chaganti, Nanda, Pandi, Gnrsn Prudhvith, & Kumar, 2020):

1. Praproses Data

   Langkah pertama dalam proses klasifikasi adalah praproses data input. Ini termasuk normalisasi atau penskalaan fitur, dan dalam beberapa kasus, pengurangan dimensi untuk mempercepat proses komputasi.

2. Transformasi Data

   Data input kemudian diubah ke dalam ruang fitur yang sesuai dengan kernel yang telah dipilih selama pelatihan. Kernel ini membantu memetakan data input ke dalam dimensi yang lebih tinggi, di mana kelas data menjadi lebih mudah dipisahkan secara linear atau non-linear.

3. Prediksi

   Setelah transformasi data, langkah selanjutnya adalah melakukan prediksi menggunakan model SVM yang telah dilatih sebelumnya. Ini melibatkan menempatkan data input ke dalam ruang fitur yang sesuai,

kemudian memproyeksikannya ke hyperplane yang telah ditentukan selama pelatihan.

4. Penentuan Kelas

   Setelah proyeksi, SVM menentukan kelas data dengan melihat di mana sisi hyperplane yang diproyeksikan data input berada. Jika data berada di satu sisi hyperplane, itu akan diklasifikasikan ke kelas satu, dan jika berada di sisi lain, itu akan diklasifikasikan ke kelas lain.

5. Evaluasi

   Setelah klasifikasi dilakukan untuk semua data input, hasilnya dievaluasi menggunakan metrik kinerja seperti akurasi, presisi, recall, atau metrik evaluasi lainnya yang sesuai dengan jenis masalah yang dihadapi.

Proses klasifikasi SVM ini dapat dilakukan dengan cepat dan efisien, terutama untuk dataset yang relatif kecil hingga sedang. Namun, untuk dataset yang sangat besar atau kompleks, proses klasifikasi dapat menjadi lebih lambat atau membutuhkan sumber daya komputasi yang lebih besar.

II.4 K-Nearest Neighbors

Algoritma K-Nearest Neighbors (KNN) adalah algoritma klasifikasi dan regresi non-parametrik yang populer dalam machine learning. Algoritma ini bekerja dengan cara mengklasifikasikan data baru berdasarkan kedekatannya dengan data

latih yang telah diberi label. KNN mengasumsikan bahwa data yang serupa

akan terletak berdekatan dalam ruang fitur (Sinaga, Hayadi, & Situmorang, 2022).

Distance metric (metrik jarak) adalah fungsi matematika yang digunakan untuk mengukur kedekatan antara dua titik dalam ruang fitur. Dalam algoritma K- Nearest Neighbors (KNN), distance metric memainkan peran penting dalam menentukan tetangga terdekat data baru. Pemilihan distance metric yang tepat sangat penting untuk performa KNN. Distance metric yang berbeda akan menghasilkan tetangga terdekat yang berbeda, dan ini dapat berakibat pada perbedaan akurasi klasifikasi atau regresi (Abu Alfeilat et al., 2019).

Beberapa distance metric yang umum digunakan dalam KNN adalah (Ali, Neagu, & Trundle, 2019):

1. Jarak Euclidean

   Distance metric ini mengukur jarak lurus antara dua titik dalam ruang fitur.

2. Jarak Manhattan

   Distance metric ini mengukur total jarak horizontal dan vertikal antara dua titik dalam ruang fitur.

3. Jarak Minkowski

   Distance metric ini adalah generalisasi dari jarak Euclidean dan Manhattan, dan memungkinkan untuk mengontrol bobot jarak pada setiap dimensi.

Proses klasifikasi menggunakan algoritma K-NN melibatkan langkah-langkah berikut (Bentéjac, Csörgő, & Martínez-Muñoz, 2021):

1. Persiapan Data
    a. Memuat data latih dan data uji yang ingin diklasifikasikan.
    b. Melakukan pra-pemrosesan data yang diperlukan, seperti normalisasi data, handling missing values, dan encoding data kategorikal.
2. Pemilihan Nilai K

    Nilai K adalah jumlah tetangga terdekat yang akan digunakan untuk klasifikasi. Nilai K yang optimal perlu ditentukan melalui eksperimen. Ada beberapa metode yang dapat digunakan untuk menentukan nilai K, seperti:

    a. Cross-validation: Melakukan cross-validation untuk mengevaluasi performa KNN dengan nilai K yang berbeda.
    b. Elbow method: Memilih nilai K yang menghasilkan kurva error terendah.
    c. Heuristic: Menggunakan aturan umum, seperti memilih K yang sama dengan akar kuadrat dari jumlah data latih.
3. Klasifikasi Data Baru
    a. Menghitung jarak antara data baru dengan semua data latih menggunakan metrik jarak, seperti jarak Euclidean, Manhattan, atau Minkowski.
    b. Mengidentifikasi K data latih terdekat dengan data baru

berdasarkan jarak yang telah dihitung.

c. Memprediksi label

d. Label mayoritas dari K tetangga terdekat akan menjadi label yang diprediksikan untuk data baru.

e. Nilai rata-rata dari K tetangga terdekat akan menjadi label yang diprediksikan untuk data baru (hanya untuk regresi).

4. Evaluasi Kinerja

Menghitung metrik akurasi, seperti accuracy, precision, recall, dan F1-score, untuk mengevaluasi performa KNN.

Algoritma KNN adalah algoritma klasifikasi yang sederhana dan mudah dipahami. Algoritma ini dapat digunakan untuk berbagai jenis data dan dapat mencapai hasil yang baik dengan pemilihan nilai K dan metrik jarak yang tepat.

II.5 Cross Validation

Cross-validation adalah salah satu teknik evaluasi kinerja model yang umum digunakan dalam pembelajaran mesin. Tujuannya adalah untuk mengevaluasi seberapa baik model yang dibangun dapat melakukan generalisasi pada data baru yang tidak terlihat selama pelatihan. Proses cross-validation umumnya melibatkan langkah-langkah berikut (Ramezan, Warner, & Maxwell, 2019):

1. Pembagian Data

Data tersedia dibagi menjadi subset yang disebut lipatan (folds).

Biasanya, data dibagi menjadi k subset yang sama besar.

2. Pemilihan Subset

   Salah satu lipatan dipilih sebagai data validasi, sementara lipatan lainnya digunakan sebagai data pelatihan. Proses ini diulangi k kali, sehingga setiap lipatan digunakan sebagai data validasi sekali dan data pelatihan k-1 kali.

3. Pelatihan Model

   Model dilatih menggunakan data pelatihan yang terpisah.

4. Evaluasi Model

   Model dievaluasi menggunakan data validasi yang terpisah. Metrik kinerja seperti akurasi, presisi, recall, atau F1-score dihitung untuk mengevaluasi kinerja model.

5. Perhitungan Kinerja

   Langkah 3 dan 4 diulangi k kali, dan hasil evaluasi dari setiap lipatan digabungkan untuk menghasilkan estimasi kinerja model secara keseluruhan. Keuntungan dari cross-validation adalah bahwa ini memberikan perkiraan kinerja yang lebih andal daripada hanya menggunakan satu set data validasi. Ini membantu dalam mengukur kemampuan model untuk generalisasi pada data baru dan mengidentifikasi apakah model cenderung overfitting atau underfitting

   (Gronau & Wagenmakers, 2019).

Salah satu bentuk cross-validation yang paling umum adalah K-Fold

Cross- Validation, di mana data dibagi menjadi k subset yang sama besar dan proses pelatihan dan evaluasi diulangi k kali menggunakan subset yang berbeda sebagai data validasi (Nti, Nyarko-Boateng, & Aning, 2021).

II.6 Penelitian Terdahulu

Sebagai referensi dalam pengembangan penelitian ini, digunakan penelitian yang berhubungan dengan topik permasalahan penelitian, yaitu seputar prediksi penjualan kendaraan bekas menggunakan algoritma SVM dan KNN. Tabel 2.1 menunjukkan penelitian terdahulu yang penulis gunakan sebagai referensi dalam penelitian ini.

Tabel II.1 Penelitian Terdahulu

Penulis	Judul	Metode Penelitian	Hasil Penelitian
(Yennimar et al., 2022)	Comparison Analysis of SVM Algorithm with Linear Regression in Predicting used Car Prices	Pertama, penelitian dimulai dengan pencarian referensi terkait topik dari studi sebelumnya. Kemudian, masalah ditentukan, dan batasan masalah ditetapkan untuk memfokuskan penelitian. Data tentang mobil dan harga	Hasil penelitian ini menunjukkan analisis komparatif antara metode SVM (Support Vector Machine) dan regresi linier dalam memprediksi harga mobil. Hasil pelatihan menunjukkan bahwa metode SVM memiliki

		dikumpulkan dari internet. Selanjutnya, data dianalisis untuk memilih fitur yang mempengaruhi penjualan. Data yang telah dikumpulkan diuji menggunakan algoritma SVM (Support Vector Machine) dan regresi linier. SVM menggunakan fungsi linier dalam ruang fitur berdimensi tinggi, sementara regresi linier memodelkan hubungan antara variabel independen dan variabel respon.	akurasi yang lebih baik dibandingkan regresi linier, dengan SVM mampu menangani data non-linear melalui teknik kernel-trick, yang tidak dapat dilakukan oleh regresi linier. Kesimpulannya, metode SVM lebih efektif dalam memprediksi harga mobil dibandingkan regresi linier.
(Samruddhi & Ashok Kumar, 2020)	Used Car Price Prediction using K- Nearest Neighbor Based	Penelitian ini menggunakan dataset mobil bekas yang dikumpulkan dari situs	Model yang dilatih menggunakan algoritma K Nearest Neighbor (KNN) untuk

	Model	Kaggle dalam format CSV, yang terdiri dari 14 variabel, termasuk harga, lokasi, dan jenis bahan bakar. Proses pengolahan data dilakukan untuk menghapus data yang tidak perlu dan menyiapkan dataset untuk analisis. Tahapan preprocessing mencakup penghapusan elemen non-numerik dari fitur numerik, konversi nilai kategorikal menjadi numerik menggunakan Label Encoder, dan pemisahan variabel target, yaitu harga. Model dibangun menggunakan algoritma	memprediksi harga mobil bekas mencapai akurasi 85%, dengan Root-Mean Squared Error (RMSE) sebesar 4.01 dan Mean Absolute Error (MAE) sebesar 2.01, khususnya pada nilai k=4. Selain itu, model juga diuji menggunakan metode k-fold cross- validation, dengan hasil akurasi 82% dan RMSE 4.73 pada 10 fold. Hasil ini menunjukkan bahwa model KNN lebih unggul dibandingkan dengan regresi linier yang hanya mencapai akurasi 71%. Penelitian ini

		K Nearest Neighbor (KNN) untuk memprediksi harga mobil bekas. Dalam penerapan KNN, model dilatih dengan berbagai nilai k (dari 2 hingga 10) untuk menentukan akurasi terbaik, dengan prediksi dilakukan berdasarkan pengukuran jarak menggunakan metrik Euclidean Distance untuk menemukan nilai terdekat.	menyimpulkan bahwa model KNN yang diusulkan merupakan model yang dioptimalkan untuk prediksi harga mobil bekas, dan penulis berencana untuk menerapkan teknik pembelajaran mesin lanjutan di masa depan untuk lebih meningkatkan akurasi model.
(Maincer et al., 2023)	Fault Diagnosis in Robot Manipulators Using SVM and KNN	Penelitian ini menggunakan pendekatan matematis dan algoritma pembelajaran mesin untuk menganalisis dan memprediksi perilaku	Dengan menggunakan 63.000 sampel dari database, penelitian ini menguji tujuh kelas kesalahan yang dihasilkan dari berbagai parameter, termasuk

		robot manipulator, khususnya robot SCARA, dengan menggunakan persamaan dinamik yang diperoleh dari formulasi Lagrangian. Persamaan dinamik tersebut mencakup matriks inersia, serta vektor gaya gravitasi, di mana sensor yang mengalami kesalahan ditangani dengan mengubah pengukuran sudut. Penelitian ini juga menerapkan algoritma K- Nearest Neighbor (KNN) untuk klasifikasi, serta Support Vector Machine (SVM) untuk optimasi parameter, menggunakan algoritma	momen inersia dan panjang link. Hasil menunjukkan bahwa akurasi deteksi kesalahan dengan PSO-SVM mencapai 96,95%, lebih tinggi dibandingkan 94,53% pada KNN. PSO berfungsi untuk mengoptimalkan parameter SVM, meningkatkan keakuratan dan efisiensi diagnosis. Simulasi menunjukkan bahwa pendekatan SVM, terutama yang terintegrasi dengan PSO, memiliki potensi besar untuk mendeteksi kesalahan dalam sistem manipulasi robotik.

		Particle Swarm Optimization (PSO) untuk meningkatkan akurasi diagnosis. Metode validasi silang (cross-validation) juga diterapkan untuk mengevaluasi kinerja model klasifikasi dengan membagi dataset menjadi beberapa lipatan (folds) untuk memastikan ketahanan dan keakuratan model yang dihasilkan.	Penelitian ini menyarankan penggabungan SVM dengan teknik optimasi lain di masa depan untuk hasil yang lebih baik.
(Untoro et al., 2020)	Evaluation of Decision Tree, K-NN, Naive Bayes and SVM with MWMOTE on UCI Dataset	Penelitian ini menggunakan dataset UCI yang tidak seimbang untuk klasifikasi dengan distribusi kelas yang tidak merata, di mana data minoritas sering	Penelitian ini mengevaluasi algoritma Decision Tree, Naive Bayes, K- NN, dan Support Vector Machine (SVM) pada dataset tidak seimbang menggunakan metrik

		kali sulit dipelajari. Untuk mengatasi ketidakseimbangan ini, digunakan metode oversampling bernama Majority Weighted Minority Oversampling Technique (MWMOTE), yang menekankan pemilihan dan pembuatan sampel sintetis yang lebih akurat melalui tiga tahap. Model klasifikasi kemudian dibangun untuk memprediksi tren data masa depan, dengan evaluasi kinerja menggunakan metrik precision, recall, F-measure, dan akurasi untuk menilai sejauh mana model memenuhi	precision, recall, f-measure, dan akurasi. Setelah oversampling dengan MWMOTE, hasil menunjukkan bahwa akurasi rata- rata meningkat 2-4% pada dataset yang seimbang. Algoritma Decision Tree menunjukkan kinerja terbaik dengan akurasi 96,30%, diikuti oleh K-NN dengan 92,95%, sedangkan SVM dan Naive Bayes mencapai masing-masing 82% dan 78,74%. Secara keseluruhan, metode Decision Tree memberikan hasil yang unggul dalam akurasi, precision, recall, dan f-

		prediksi dengan nilai aktual.	measure dibandingkan algoritma lainnya, baik pada data awal maupun setelah penyeimbangan.
(Nti et al., 2021)	Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation	Penelitian ini menggunakan lima tahap untuk mengembangkan model prediksi stroke berdasarkan dataset yang diunduh dari UCI Machine Learning Repository. Pertama, data dipersiapkan dan diskalakan agar lebih efisien, dilanjutkan dengan k-fold cross-validation (CV) menggunakan berbagai nilai k untuk mengoptimalkan akurasi model. Empat algoritma machine learning (GBM,	Penelitian ini meneliti pengaruh nilai k pada metode k-fold cross-validation (CV) terhadap performa empat algoritma machine learning (ML) – Gradient Boosting Machine (GBM), Logistic Regression (LR), Decision Tree (DT), dan K-Nearest Neighbors (KNN) – dalam memprediksi kemungkinan seseorang mengalami stroke. Dengan menguji nilai k dari 3 hingga 20,

		Logistic Regression, Decision Tree, dan KNN) dilatih dan diuji kinerjanya berdasarkan akurasi dan area under the curve (AUC) dengan algoritma scikit- learn default. Model dievaluasi untuk menentukan kinerja optimal tiap nilai k, dan semua analisis dilakukan dalam lingkungan Python dengan menggunakan Pandas, Scikit-Learn, dan Matplotlib pada laptop dengan spesifikasi tertentu.	ditemukan bahwa nilai optimal k bergantung pada algoritma yang digunakan, misalnya, k = 3 cocok untuk LR, k = 7 untuk GBM dan KNN, serta k = 15 untuk DT. Meski k = 10 umum digunakan dalam literatur, hasil penelitian ini menunjukkan bahwa k = 7 memberikan akurasi dan AUC lebih baik serta kompleksitas komputasi lebih rendah. LOOCV memberikan akurasi sedikit lebih tinggi dibandingkan k-fold (k = 7 dan k = 10) tetapi membutuhkan waktu komputasi yang jauh lebih besar. Penelitian ini

			menyimpulkan bahwa pemilihan k yang tepat sangat penting karena setiap algoritma memiliki respon berbeda terhadap perubahan nilai k.
(Kaliappan et al., 2023)	Impact of Cross-Validation on Machine Learning Models for Early Detection of Intrauterine Fetal Demise	Penelitian ini menggunakan data Cardiotocography yang memiliki 2126 baris dan 22 kolom dengan label kondisi Normal, Suspect, dan Pathological. Karena data tidak seimbang, metode Random Over Sampler digunakan untuk menyeimbangkan data. Tahapan penelitian meliputi Analisis Data Eksplorasi (EDA),	Penelitian ini mengevaluasi sembilan algoritma pembelajaran mesin pada dataset kardiotokografi janin dengan membagi data menjadi 80% pelatihan dan 20% pengujian serta menerapkan berbagai teknik validasi silang untuk meningkatkan kinerja model. Hasil analisis menunjukkan bahwa teknik validasi silang mampu meningkatkan

		klasifikasi menggunakan algoritma seperti Decision Tree, Random Forest, KNN, Naive Bayes, dan ensemble methods (Voting, AdaBoost, Gradient Boosting, dan Support Vector Classifier) yang dievaluasi menggunakan teknik validasi silang (misalnya, K-fold, Hold-Out, dan Monte Carlo). Selain itu, pendekatan Explainable AI menggunakan LIME dan SHAP diaplikasikan untuk menilai interpretasi model, menunjukkan bahwa fitur seperti Prolonged Deceleration berpengaruh signifikan	akurasi model hingga 2%, dengan Gradient Boosting dan Voting Classifier menghasilkan kinerja terbaik dengan akurasi 0,99, recall 0,98, precision 0,96–0,98, dan F1-score sekitar 0,97. Teknik evaluasi menggunakan LIME dan SHAP menyoroti fitur-fitur yang paling berpengaruh pada performa model. Meski hasil menunjukkan potensi tinggi, penelitian ini mengidentifikasi perlunya dataset yang lebih besar untuk meningkatkan prediksi status janin di masa

		terhadap akurasi prediksi kondisi kesehatan janin.	depan.

Penelitian oleh (Yennimar et al., 2022) membandingkan efektivitas algoritma Support Vector Machine (SVM) dan regresi linier dalam memprediksi harga mobil bekas. Dalam penelitian ini, data tentang harga mobil dikumpulkan dan dianalisis untuk memilih fitur-fitur yang memengaruhi harga jual. Hasil menunjukkan bahwa SVM memiliki akurasi prediksi lebih tinggi dibandingkan regresi linier, berkat kemampuannya menangani data non-linear melalui teknik kernel-trick. Penelitian ini menyimpulkan bahwa SVM lebih efektif dibandingkan regresi linier untuk memprediksi harga mobil.

(Samruddhi & Ashok Kumar, 2020) menerapkan algoritma K-Nearest Neighbor (KNN) untuk memprediksi harga mobil bekas menggunakan dataset dari

Kaggle. Data diproses melalui berbagai tahap, termasuk penghapusan elemen non- numerik dan konversi data kategorikal. Model KNN diuji dengan nilai k yang berbeda dan menghasilkan akurasi sebesar 85% dengan nilai k=4, serta menunjukkan performa unggul dibandingkan regresi linier. Model ini juga diuji menggunakan k-fold cross-validation, menghasilkan akurasi sebesar 82%, menunjukkan potensi KNN sebagai model prediksi harga mobil bekas yang efisien. (Maincer et al., 2023) menggunakan algoritma SVM dan KNN untuk mendiagnosis kesalahan pada robot

manipulator, khususnya robot SCARA. Dengan pendekatan dinamik Lagrangian, penelitian ini mengaplikasikan Particle Swarm Optimization (PSO) untuk meningkatkan akurasi SVM dalam diagnosis. Hasil menunjukkan bahwa PSO-SVM mencapai akurasi 96,95%, lebih tinggi dibandingkan KNN. Simulasi ini menunjukkan bahwa integrasi SVM dengan PSO memiliki potensi tinggi untuk mendeteksi kesalahan pada robot manipulasi, dengan saran pengembangan menggunakan teknik optimasi lainnya.

Penelitian oleh (Untoro et al., 2020) mengevaluasi algoritma Decision Tree, K-NN, Naive Bayes, dan SVM pada dataset tidak seimbang dari UCI menggunakan metode oversampling MWMOTE untuk menyeimbangkan data. Setelah penyeimbangan, algoritma Decision Tree memberikan hasil akurasi tertinggi (96,30%), diikuti oleh K-NN. Hasil ini menunjukkan bahwa metode Decision Tree unggul dalam akurasi dan performa metrik lainnya, bahkan setelah penyeimbangan data menggunakan MWMOTE, menjadikannya metode terbaik untuk dataset tidak seimbang.

(Nti et al., 2021) mengkaji pengaruh variasi nilai k dalam k-fold cross-validation pada performa empat algoritma pembelajaran mesin, termasuk Gradient

Boosting Machine (GBM) dan K-Nearest Neighbors (KNN). Penelitian ini menemukan bahwa nilai k optimal bervariasi untuk tiap algoritma, misalnya k=7 untuk GBM dan KNN. Hasil menunjukkan bahwa pemilihan nilai k yang tepat penting untuk meningkatkan akurasi dan

mengurangi kompleksitas komputasi, terutama dengan hasil yang lebih baik pada k=7 dibandingkan dengan k=10 atau leave-one-out cross-validation.

(Kaliappan et al., 2023) mengevaluasi sembilan algoritma pembelajaran mesin pada data kardiotokografi untuk mendeteksi dini kematian janin dalam kandungan. Dengan data yang tidak seimbang, metode Random Over Sampler digunakan untuk menyeimbangkan data sebelum klasifikasi dilakukan menggunakan algoritma seperti Decision Tree dan Gradient Boosting. Validasi silang meningkatkan akurasi model hingga 2%, dengan Gradient Boosting dan Voting Classifier menghasilkan performa terbaik. Hasil analisis menggunakan LIME dan SHAP menunjukkan bahwa beberapa fitur, seperti Prolonged Deceleration, sangat memengaruhi prediksi. Penelitian ini menyoroti potensi model pembelajaran mesin untuk prediksi kesehatan janin dan menyarankan penggunaan dataset yang lebih besar untuk hasil yang lebih akurat.