

BAB III METODOLOGI

3.1 Pendahuluan

Bab ini dimulai dengan memperkenalkan alat yang digunakan dalam penelitian ini yaitu menggunakan python. Selain itu, dijelaskan pula mengenai dataset yang digunakan dalam penelitian ini. Setelah itu, bab ini akan menjelaskan secara rinci metodologi penelitian yang diikuti guna mencapai tujuan penelitian dari studi ini. Metodologi yang digunakan dalam penelitian ini adalah menggunakan ML untuk pemodelan prediktif, yang diadopsi dari Raschka [29]. Metodologi ini telah terbukti efektif dan dapat diandalkan dalam penelitian sejenis.

3.2 Alat dan Bahan

Dalam penelitian ini, Python dipilih sebagai alat utama untuk melakukan pemodelan prediktif dan analisis data. Python menawarkan berbagai pustaka dan alat open-source yang kuat, seperti scikit-learn, TensorFlow, dan Keras, yang memungkinkan peneliti untuk membangun, menguji, dan mengevaluasi berbagai model machine learning (ML) dengan fleksibilitas yang tinggi. Dengan Python, peneliti memiliki kontrol penuh terhadap proses pemodelan, mulai dari pre-processing data hingga pembangunan model, evaluasi, dan penyetelan parameter.

Scikit-learn adalah pustaka Python yang sangat populer untuk machine learning dan analisis data. Dalam penelitian ini, scikit-learn digunakan untuk membangun berbagai model ML, seperti regresi, klasifikasi, dan klustering. Selain itu, scikit-

learn juga menyediakan berbagai metrik evaluasi, seperti akurasi, presisi, dan recall, yang memungkinkan peneliti untuk mengukur kinerja model secara komprehensif.

Selain untuk pemodelan, Python juga digunakan untuk melakukan analisis eksploratif data dan visualisasi. Pustaka seperti Pandas digunakan untuk manipulasi dan pembersihan data, sementara Matplotlib digunakan untuk membuat visualisasi data yang informatif, seperti grafik, plot, dan diagram, yang memudahkan pemahaman dan interpretasi terhadap data.

Google Colab dipilih sebagai lingkungan pengembangan utama dalam penelitian ini. Google Colab menawarkan infrastruktur yang kuat dan gratis untuk menjalankan kode Python dalam lingkungan cloud dengan menggunakan sumber daya komputasi yang tersedia dari Google. Dengan Google Colab, peneliti dapat menjalankan kode Python secara interaktif, membuat dan menyimpan notebook secara online, serta berkolaborasi dengan rekan penelitian dalam waktu nyata.

3.3 Dataset

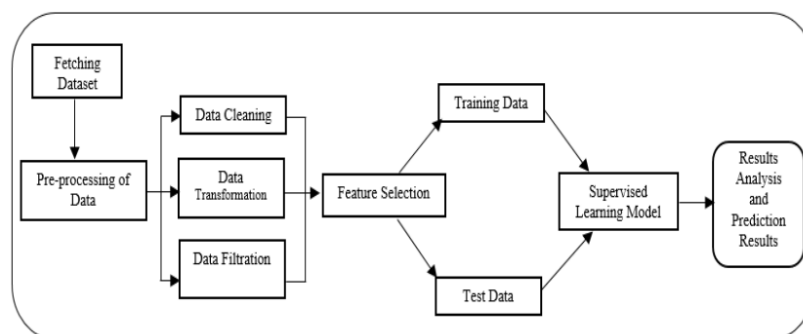
Pada Penelitian ini menggunakan data sekunder yang bersumber dari Badan Pusat Statistik (BPS) berupa data mentah Survei Sosial Ekonomi Nasional (Susenas) Maret dari tahun 2020 hingga tahun 2022 baik KOR maupun Modul Konsumsi/Pengeluaran Kabupaten Pesawaran yang dapat diunduh di website BPS Silastik.bps.go.id.

3.4 Pemodelan Prediktif

Raschka [29] melakukan penelitian dengan menggunakan metode Machine Learning (ML) dalam pemodelan prediktif untuk mengklasifikasikan variabel target rumah tangga sebagai "miskin" atau "tidak miskin". Oleh karena itu, proses prediksi ML dimulai dengan tahap pra-pemrosesan data (i), yang kemudian diikuti oleh tahap pembelajaran (ii), yang melibatkan pelatihan dan pemilihan model prediktif. Tahap terakhir adalah evaluasi model dan prediksi data (iii).

Dalam penelitian ini, pengklasifikasi yang digunakan dalam pipeline ML juga telah diterapkan oleh Fitzpatrick dan rekan-rekannya [13] untuk memprediksi tingkat kemiskinan dengan menggunakan data survei. Namun, beberapa modifikasi telah dilakukan pada langkah-langkah yang dilakukan oleh Fitzpatrick untuk meningkatkan kinerja model prediktif dan untuk menjawab pertanyaan penelitian dalam studi ini. Modifikasi-modifikasi tersebut akan dijelaskan secara rinci pada langkah-langkah yang relevan dalam penelitian ini.

Berdasarkan penjelasan di atas, kerangka pikir yang digunakan dalam penelitian ini adalah sebagai berikut :



Gambar 3. 1 Kerangka Pikir Machine Learning

3.4.1 Pra-Processing Data

Tahap pertama dalam pemodelan pada penelitian ini adalah pra-pemrosesan data. Pra-pemrosesan data merupakan proses awal yang dilakukan sebelum data dapat digunakan untuk melatih dan menguji model prediktif. Tujuan dari tahap ini adalah untuk membersihkan, mengubah, dan mempersiapkan data agar sesuai dengan kebutuhan analisis. Tahap pra-pemrosesan data sangat penting untuk memastikan bahwa data yang digunakan dalam pemodelan prediktif berkualitas, terstandarisasi, dan siap untuk dianalisis. Dengan melakukan pra-pemrosesan data yang baik, penelitian ini dapat meminimalkan kesalahan dan bias yang mungkin timbul serta memaksimalkan kemampuan model prediktif untuk memberikan hasil yang akurat dan bermakna.

3.4.1.1 Ekstraksi Data

Tahap awal pra-pemrosesan data adalah data extraction atau ekstraksi data. Pada tahap ini, data yang relevan dan diperlukan untuk penelitian dikumpulkan dari sumber-sumber yang tersedia. Data extraction melibatkan proses mengidentifikasi, memperoleh, dan mengumpulkan data yang akan digunakan dalam analisis.

Data extraction merupakan tahap awal yang penting dalam pra-pemrosesan data, karena keberhasilan penelitian sangat tergantung pada kualitas dan kecukupan data yang digunakan. Dengan melakukan ekstraksi data yang baik, peneliti dapat memastikan bahwa data yang diperoleh relevan dengan pertanyaan penelitian dan siap untuk diproses dan dianalisis dalam tahap pra-pemrosesan selanjutnya.

3.4.1.2 Mengatasi *Missing Data*

Setelah memilih atribut dan melabelkannya, menghapus sebuah missing value juga sangat penting dalam pengolahan data, salah satunya menggunakan operator Replace Missing Values, dengan params average atau rata-rata untuk mengganti nilai yang noise.

3.4.1.3 Dropping Features

Sering sekali, dalam Machine Learning (ML), kita dihadapkan dengan data yang memiliki banyak fitur. Hal ini menjadi tantangan karena meningkatkan risiko adanya korelasi tinggi antara fitur-fitur tersebut, dan juga meningkatkan kebutuhan akan daya komputasi yang lebih tinggi. Dalam hal ini, penciptaan fitur numerik dapat memberikan kontribusi lebih lanjut pada masalah ini [29].

Langkah berikutnya adalah menghapus fitur-fitur yang dianggap tidak berguna untuk klasifikasi. Hal ini dilakukan agar hanya fitur-fitur yang memiliki nilai prediktif yang tinggi yang tetap dipertahankan untuk melatih model ML. Dengan demikian, hanya fitur-fitur yang paling relevan yang digunakan dalam proses pembelajaran model, sehingga meningkatkan efisiensi dan keakuratan dalam klasifikasi.

3.4.1.4 Eksplorasi Data

Setelah tahap pra-pemrosesan selesai, dataset digunakan untuk eksplorasi guna mendapatkan pemahaman yang lebih baik tentang hubungan antara fitur-fitur yang ada. Analisis eksplorasi data memainkan peran penting dalam menentukan model Machine Learning (ML) yang akan digunakan dan fitur-fitur mana yang perlu

diberikan fokus. Dalam sebuah artikel [30], dijelaskan bahwa analisis data tidak hanya berguna untuk menguji hipotesis yang sudah ada, tetapi juga untuk menemukan hipotesis baru dengan menggunakan data yang tersedia.

Dalam penelitian ini, hubungan antara fitur-fitur dan variabel target "kemiskinan" dieksplorasi. Data diperiksa dengan memplotkan distribusi fitur-fitur terhadap variabel target "kemiskinan" atau dengan mengukur korelasinya. Langkah pertama yang dilakukan dalam penelitian ini adalah memetakan fitur numerik yang terbatas pada dataset Kabupaten Pesawaran terhadap variabel target "kemiskinan". Dengan melakukan langkah ini, penelitian ini dapat memperoleh wawasan yang lebih baik tentang bagaimana fitur-fitur tertentu berhubungan dengan variabel target dan memberikan landasan untuk analisis lebih lanjut.

3.4.1.5 Seleksi *Feature*

Seleksi Fitur pada penelitian ini menggunakan metode filter. Metode Filter merupakan sebuah pendekatan yang digunakan untuk mengurutkan fitur-fitur dengan melakukan uji statistik untuk menghitung koefisien korelasi antara fitur-fitur input. Pendekatan ini telah digunakan dalam metode klasifikasi yang dikembangkan oleh Fitzpatrick dkk. [13]. Dalam metode Filter, fitur-fitur tersebut kemudian diberi peringkat berdasarkan hasil perkalian antara koefisien korelasi dan deviasi standar dari parameter yang terkait dalam data. Dengan menggunakan metode ini, kita dapat mengidentifikasi fitur-fitur yang memiliki hubungan kuat dengan variabel target, serta mengurutkannya berdasarkan tingkat relevansi dan kepentingannya dalam analisis lebih lanjut.

Seleksi Fitur pada penelitian ini menggunakan empat metode yang berbeda, yaitu regresi logistik, SelectKBest, Recursive Feature Elimination (RFE), dan Mutual Information.

a) Regresi Logistik

Regresi Logistik adalah metode statistik yang digunakan untuk menganalisis hubungan antara satu atau lebih variabel independen (fitur) dengan variabel dependen biner (variabel target) dan memperkirakan probabilitas bahwa variabel dependen akan terjadi. Dalam konteks seleksi fitur, Regresi Logistik digunakan untuk mengevaluasi signifikansi hubungan antara setiap fitur dan variabel target, sehingga fitur-fitur yang memiliki pengaruh signifikan terhadap variabel target dapat dipilih untuk dimasukkan ke dalam model.

Rumus regresi logistik umumnya diwakili oleh fungsi logit (log-odds) yang dihubungkan dengan variabel independen melalui koefisien regresi. Dalam kasus regresi logistik biner, rumusnya adalah sebagai berikut:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

(7)

di mana:

- p adalah probabilitas dari variabel dependen yang bernilai 1,
- x_1, x_2, \dots, x_n adalah nilai-nilai dari variabel independen (fitur),
- $\beta_0, \beta_1, \dots, \beta_n$ adalah koefisien regresi yang akan diestimasi.

Koefisien regresi (β) dalam model regresi logistik menunjukkan perubahan dalam log-odds dari variabel dependen yang dihasilkan oleh satu unit perubahan dalam variabel independen, semuanya tetap pada tingkat variabel independen lainnya.

Dengan demikian, dalam konteks seleksi fitur, fitur-fitur yang memiliki koefisien regresi yang signifikan dalam model regresi logistik akan dipilih untuk dimasukkan ke dalam model, karena mereka memiliki pengaruh yang penting terhadap variabel target yang sedang diprediksi.

b) Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) adalah metode seleksi fitur yang digunakan untuk memilih subset fitur yang paling penting dari sekumpulan fitur yang tersedia. Metode ini bekerja dengan mengurangi jumlah fitur secara bertahap dari model, dimulai dari keseluruhan fitur hingga hanya tersisa subset fitur yang paling penting untuk meningkatkan performa model.

Berikut adalah langkah-langkah umum yang dilakukan dalam Recursive Feature Elimination:

- i. **Inisialisasi Model:** Langkah pertama adalah inisialisasi model machine learning yang akan digunakan untuk evaluasi. Model ini dapat berupa model regresi logistik, model Support Vector Machine (SVM), atau model lain yang sesuai dengan masalah yang sedang dihadapi.
- ii. **Evaluasi Fitur:** Setelah model telah diinisialisasi, model tersebut akan dilatih dengan menggunakan semua fitur yang tersedia. Kemudian, setiap fitur akan dievaluasi berdasarkan kepentingannya dalam model. Salah satu

metode evaluasi yang umum digunakan adalah koefisien dari fitur tersebut dalam model (misalnya, koefisien regresi logistik).

- iii. Penghapusan Fitur: Fitur dengan pengaruh paling rendah terhadap model akan dihapus dari dataset.
- iv. Perulangan: Langkah kedua dan ketiga diulang kembali pada dataset yang telah dimodifikasi (tanpa fitur yang telah dihapus) hingga jumlah fitur yang diinginkan telah tercapai atau hingga kriteria berhenti yang ditetapkan telah terpenuhi. Kriteria berhenti ini bisa berupa jumlah fitur yang tersisa atau perubahan yang signifikan dalam performa model.
- v. Seleksi Fitur Terbaik: Setelah sejumlah iterasi, subset fitur yang tersisa akan dipilih sebagai fitur terbaik yang akan digunakan untuk membangun model akhir.

Rumus dalam RFE tidak ada yang spesifik karena ini lebih ke arah algoritma iteratif. Namun, konsep utama dalam RFE adalah untuk menghilangkan satu fitur pada setiap iterasi dan memilih subset fitur yang memberikan performa model terbaik. Pada setiap iterasi, fitur dengan koefisien terendah atau yang memberikan kontribusi paling sedikit terhadap model dihapus, dan iterasi dilanjutkan hingga hanya tinggal subset fitur yang dianggap penting untuk meningkatkan performa model.

c) Select K best

SelectKBest adalah metode seleksi fitur yang digunakan untuk memilih k fitur terbaik dari dataset berdasarkan skor statistik tertentu. Tujuan utama dari

SelectKBest adalah untuk memperoleh subset fitur yang paling informatif atau penting dalam meningkatkan performa model prediktif. Metode ini berguna terutama ketika dimensi fitur dari dataset sangat besar dan ingin menyeleksi fitur-fitur yang paling relevan untuk model.

Berikut adalah langkah-langkah umum yang dilakukan dalam SelectKBest:

- i. Penghitungan Skor Statistik: Pada langkah pertama, skor statistik dihitung untuk setiap fitur dalam dataset. Skor statistik ini menunjukkan seberapa baik setiap fitur berhubungan dengan variabel target atau label dalam dataset.
- ii. Rangkaian Fitur: Setelah skor statistik dihitung, fitur-fitur tersebut diberi peringkat berdasarkan skor tersebut. Fitur dengan skor statistik tertinggi dianggap sebagai fitur terbaik atau paling informatif.
- iii. Seleksi K Fitur Terbaik: Langkah terakhir adalah memilih k fitur terbaik dari dataset berdasarkan peringkat yang telah dihasilkan. K adalah jumlah fitur yang ingin dipertahankan dalam subset fitur akhir.

Rumus yang umum digunakan dalam SelectKBest bergantung pada jenis skor statistik yang dipilih. Dua skor statistik yang umum digunakan adalah skor chi-square dan skor F-score.

1. Skor Chi-square (untuk fitur kategorikal): Skor chi-square mengukur seberapa erat hubungan antara setiap fitur kategorikal dengan variabel target. Rumusnya adalah sebagai berikut:

$$\chi^2(X, y) = \sum \frac{(O-E)^2}{E}$$

(8)

di mana:

- a) O adalah frekuensi observasi aktual dari kombinasi nilai fitur dan label,
 - b) E adalah frekuensi yang diharapkan dari kombinasi nilai fitur dan label jika tidak ada hubungan antara fitur dan label.
2. Skor F-score (untuk fitur numerik): Skor F-score mengukur seberapa baik variabel independen (fitur) membedakan antara kelompok-kelompok pada variabel dependen (label). Rumusnya adalah sebagai berikut:

$$F\text{-score} = \frac{\text{Between-group Variability}}{\text{Within-group Variability}} \quad (9)$$

di mana:

- Variabilitas antara kelompok dihitung sebagai jumlah kuadrat perbedaan antara rata-rata kelompok.
- Variabilitas dalam kelompok dihitung sebagai jumlah kuadrat perbedaan antara nilai individu dan rata-rata kelompok.

Dengan menggunakan salah satu dari skor statistik tersebut, SelectKBest akan memilih k fitur terbaik yang memiliki skor statistik tertinggi untuk dimasukkan ke dalam model. Ini membantu meningkatkan efisiensi model dengan menggunakan hanya fitur-fitur yang paling informatif.

c) *Mutual Information*

Mutual Information adalah metode yang digunakan untuk mengukur seberapa banyak informasi yang dibagikan oleh dua variabel acak. Dalam konteks seleksi fitur, Mutual Information digunakan untuk menentukan seberapa informatif atau pentingnya suatu fitur terhadap variabel target. Fitur-fitur yang memiliki mutual information yang tinggi dengan variabel target akan dipilih untuk dimasukkan ke dalam model, karena fitur-fitur tersebut menyediakan informasi yang lebih banyak tentang variabel target.

Berikut adalah penjelasan lebih rinci beserta rumus Mutual Information:

Konsep Dasar: Mutual Information (MI) mengukur seberapa banyak informasi yang terkandung dalam satu variabel tentang variabel lain. Jika dua variabel saling tergantung, Mutual Information antara keduanya akan tinggi, menunjukkan bahwa informasi dari satu variabel dapat digunakan untuk memprediksi nilai dari variabel lainnya.

Perhitungan Mutual Information: Untuk dua variabel diskrit X dan Y , Mutual Information dihitung sebagai berikut:

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (10)$$

di mana:

- $I(X; Y)$ adalah Mutual Information antara variabel X dan Y ,
- $p(x, y)$ adalah probabilitas bersama dari variabel X dan Y ,
- $p(x)$ dan $p(y)$ adalah probabilitas masing-masing variabel.

Rumus ini mengukur perbedaan antara distribusi bersama variabel X dan Y dengan hasil perkalian dari distribusi masing-masing variabel. Semakin besar nilai $I(X;Y)$, semakin banyak informasi yang saling terkandung di antara kedua variabel.

Seleksi Fitur dengan Mutual Information: Dalam konteks seleksi fitur, Mutual Information digunakan untuk menentukan seberapa informatifnya suatu fitur terhadap variabel target. Fitur-fitur yang memiliki nilai Mutual Information yang tinggi dengan variabel target akan dipilih untuk dimasukkan ke dalam model, karena fitur-fitur tersebut memberikan informasi yang lebih relevan atau berguna dalam memprediksi variabel target.

Dengan menggunakan Mutual Information dalam seleksi fitur, kita dapat memilih fitur-fitur yang paling informatif untuk digunakan dalam pembangunan model. Ini membantu memperbaiki kualitas model dengan menggunakan hanya fitur-fitur yang paling relevan dengan variabel target.

3.4.1.6 Balancing Data

Pada penelitian ini dilakukan balancing data agar distribusi kelas dalam dataset menjadi seimbang. Ketidakseimbangan kelas data dapat menyebabkan bias pada model yang dilatih, di mana kelas mayoritas mendominasi dan kelas minoritas sulit diprediksi dengan akurat.

Untuk mengatasi masalah ini, Teknik yang digunakan pada penelitian ini adalah SMOTE. SMOTE dapat menghasilkan sampel sintetis yang memperkaya variasi dalam kelas minoritas, mengurangi risiko overfitting pada data tersebut.

Secara matematis, rumus untuk menghitung titik sintetis baru x_{new} adalah sebagai berikut:

$$x_{new} = x + (nn - x) \times \lambda \quad (11)$$

di mana:

- x adalah contoh yang dipilih dari kelas minoritas.
- nn adalah tetangga yang dipilih dari contoh x .
- λ adalah angka acak antara 0 dan 1.
- x_{new} adalah titik sintetis baru yang dihasilkan.

Dengan melakukan balancing data, penelitian ini diharapkan dapat menghasilkan model yang lebih baik dalam memprediksi kelas minoritas, mengurangi bias, dan meningkatkan keseimbangan performa model di semua kelas.

3.4.2 Pembelajaran: Melatih dan Memilih Model Prediksi

Ada dua langkah utama dalam *supervised learning*. Pertama, model prediksi dilatih dengan menggunakan data pelatihan yang sudah diberi label. Tujuannya adalah untuk menghasilkan model klasifikasi yang dapat memprediksi dengan tepat apakah seseorang miskin atau tidak miskin.

Pada langkah kedua, model yang sudah dilatih diuji menggunakan data uji. Model tersebut akan mencoba memprediksi variabel target dengan benar, yaitu menentukan apakah seseorang miskin atau tidak. Kinerja model dinilai dengan menggunakan berbagai metrik kinerja, dan model terbaik dipilih berdasarkan hasil perbandingan.

Secara keseluruhan, pelatihan model dalam *supervised learning* melibatkan langkah-langkah ini untuk menghasilkan model yang akurat dalam memprediksi status kemiskinan seseorang.

3.4.2.1 Model Selection

Karena tujuan penelitian ini adalah untuk memprediksi variabel target "miskin" atau "tidak miskin", maka masalah yang dihadapi adalah klasifikasi biner. Setelah data dimasukkan ke dalam Python, algoritma di dalamnya menganalisis dan belajar dari data pelatihan untuk menemukan parameter prediksi terbaik. Parameter ini digunakan untuk membuat model klasifikasi. Model klasifikasi tersebut kemudian diuji menggunakan dataset pengujian untuk mengevaluasi kinerjanya dalam memprediksi dengan tepat apakah seseorang "miskin" atau "tidak miskin".

Dalam penelitian ini, dua model klasifikasi yang dianggap paling sesuai berdasarkan tinjauan literatur di Bab 2, yang membahas metode pembelajaran mesin yang digunakan dalam klasifikasi kemiskinan menggunakan data survei, dipilih. Model-model tersebut adalah: i) C4.5 dan; ii) K-Neares Neighbour. Kedua model ini digunakan untuk melatih, menguji, mengevaluasi, dan memprediksi label dalam masalah klasifikasi biner ini.

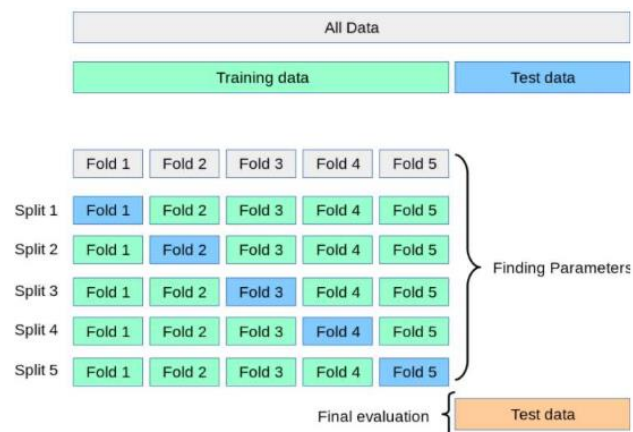
3.4.2.2 Cross Validation

Pelatihan dan pengujian saja tidak cukup karena hal tersebut dapat menyebabkan model terlalu sesuai dengan data pelatihan dan berdampak negatif pada kinerjanya saat digunakan pada data baru. Untuk mengatasi masalah ini, diperlukan subset

terpisah dari dataset yang digunakan untuk memvalidasi model. Oleh karena itu, dataset dibagi menjadi tiga bagian: pelatihan, pengujian, dan validasi.

Namun, dengan membagi data menjadi tiga bagian, jumlah sampel yang tersedia untuk melatih algoritme menjadi lebih sedikit, yang dapat mempengaruhi kinerja model. Validasi silang memberikan solusi untuk mengatasi masalah ini. Dalam validasi silang, dataset pelatihan dibagi menjadi k subset yang lebih kecil atau fold. Model kemudian dilatih pada $k-1$ fold dan diuji pada fold yang tersisa. Proses ini diulang sebanyak k kali, dan kinerja model yang dilaporkan adalah rata-rata dari hasil pengujian di setiap iterasi. Dalam hal ini, proses ini disebut k -fold cross-validation.

Dengan menggunakan validasi silang, model dapat dilatih dan dievaluasi dengan menggunakan lebih banyak data, sementara tetap mempertahankan subset yang cukup untuk validasi. Proses ini membantu menghindari overfitting dan memberikan estimasi yang lebih akurat tentang kinerja model pada data yang belum pernah dilihat sebelumnya. Gambar 3.2 menggambarkan ilustrasi dari proses k -fold cross-validation.



Source: SciKit Learn (2021, p. 33)

Gambar 3. 2 Skema 5-fold Cross Validation

Dalam penelitian ini menggunakan 10-fold cross validation karena data yang tergolong relatif sedikit yaitu 600.

3.4.2.3 Pengujian Performa

Dalam klasifikasi, ada beberapa metrik yang dapat digunakan untuk mengevaluasi performa sebuah model.

Ini termasuk (i) akurasi; (ii) *Area Under Curve* (AUC); (iii) Recall; (iv) Presisi; (v) F1 measure; dan (vi) Kappa. Pada penelitian ini kinerja model yang dilatih menggunakan semua metrik ini. Metrik-metrik tersebut dijelaskan di bawah ini.

a) *Confusion Matrix*

Confusion Matrix adalah alat yang merangkum kinerja model klasifikasi dalam memprediksi dengan benar kelas variabel target, dalam hal ini, kelas positif "miskin" dan kelas negatif "tidak miskin". Untuk setiap model klasifikasi, tempat sampah dalam matriks menangkap prediksi yang benar (positif dan negatif) dan prediksi yang salah atau salah klasifikasi (positif palsu dan negatif palsu). Semua

metrik kinerja yang digunakan untuk mengevaluasi kinerja dalam penelitian ini yang dijelaskan di bawah ini dapat disimpulkan dari confusion matrix [31].

b) *Accuracy*

Akurasi adalah metrik yang paling sering digunakan dalam klasifikasi [29]. Secara sederhana, akurasi melaporkan porsi atau persentase prediksi yang benar, yang secara matematis dinyatakan sebagai berikut.

$$\text{Akurasi} = \frac{TP + TN}{TP + FP + TN + FN} \quad (12)$$

Di mana TP = true positive (yaitu rumah tangga miskin); TN = true negative (yaitu rumah tangga tidak miskin); FP = positif False dan FN = negatif False. Akurasi dapat diandalkan sebagai metrik kinerja tunggal ketika dataset memiliki keseimbangan yang baik.

Namun, metrik ini sangat sensitif terhadap dataset yang tidak seimbang, di mana satu kelas jauh lebih lebih dominan daripada yang lain. Dalam kasus seperti itu, mengandalkan akurasi saja sebagai ukuran kinerja model dapat menyesatkan [13]. Oleh karena itu, metrik ini sering digunakan dalam kombinasi dengan metrik kinerja lainnya dan kinerja model dinilai berdasarkan hasil keseluruhan. Meskipun demikian, untuk tujuan ketahanan, kinerja model kinerja model dalam hal metrik lain juga dipertimbangkan.

c) *Recall*

Recall berfokus pada bagian dataset yang positif. Oleh karena itu, ini adalah tingkat di mana pengklasifikasi dapat memprediksi dengan benar positif yang sebenarnya.

Ini adalah rasio dari true positive (TP) terhadap semua positif, yang merupakan TP dan FN. Representasi persamaan disediakan di bawah ini.

$$\text{TPR} = \frac{TP}{TP+FN} \quad (13)$$

Pengklasifikasi yang baik adalah pengklasifikasi yang memiliki sedikit FN, dan oleh karena itu, memiliki tingkat positif yang tinggi (TPR). Recall sangat membantu untuk mengukur kinerja model ketika dataset tidak seimbang [13]. Dalam studi ini, recall yang rendah berarti terlalu banyak rumah tangga miskin yang tidak menerima bantuan. bantuan.

d) *Presisi*

Presisi didefinisikan sebagai nilai prediksi positif (PPV) [13]; [32]. Hal ini melihat totalitas dari apa yang diidentifikasi oleh model sebagai positif, dan berapa banyak dari mereka yang benar-benar positif. yang benar-benar positif. Oleh karena itu, persamaan untuk presisi adalah sebagai berikut, di mana FP = False positif.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (14)$$

Semakin tinggi jumlah positif palsu, semakin rendah presisi model. Oleh karena itu, ketepatan adalah penyeimbang dari recall, dan akan sangat membantu untuk melihat kedua metrik tersebut secara bersamaan. Dalam penelitian ini, presisi yang rendah berarti bantuan ditargetkan pada terlalu banyak rumah tangga yang tidak membutuhkannya.

e) Ukuran F1

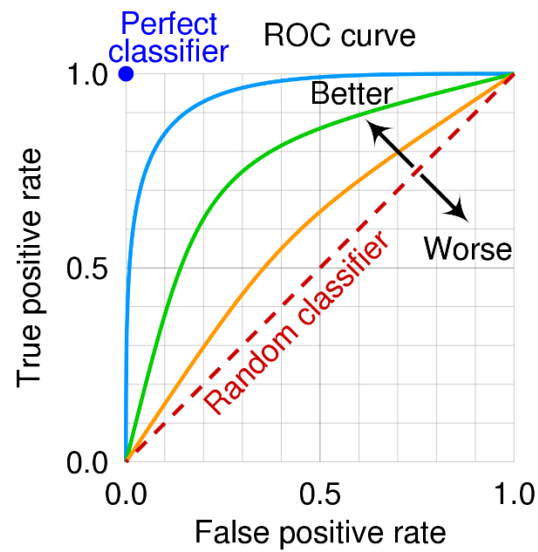
Nilai F1 menggabungkan presisi dan recall yang memberikan "rata-rata harmonis". Hal ini memberikan cara yang mudah untuk melihat ketepatan dan recall secara bersamaan [13] merepresentasikan F1 sebagai berikut.

$$F1 = \frac{2TP}{2TP + FN + FP} \quad (15)$$

Sebuah pengklasifikasi mendapatkan nilai F1 yang tinggi jika recall dan presisi tinggi.

f) Area di Bawah Kurva

Kurva Karakteristik Operasi Penerima (ROC) adalah metrik kinerja yang dapat direpresentasikan dalam bentuk grafik, yang menampilkan hubungan antara kelas True Positive Rate (TPR) dan False Kelas prediksi False Positive Rate (FPR). Ini adalah salah satu alat yang paling sering digunakan untuk mengukur kinerja pengklasifikasi ML, terutama secara visual [33] dan [13]. ROC dibangun dengan memplot tingkat positif yang benar (atau Recall) terhadap tingkat positif palsu. Satu titik pada kurva ROC, TPR dan FPR terkait, adalah diperoleh dari *confussion matrix* pengklasifikasi. Nilai AUC yang tinggi mewakili TPR yang tinggi dan FPR yang rendah. Gambar 3.2 memberikan panduan untuk menginterpretasikan kurva ROC dan nilai AUC.



Gambar 3. 3 Grafik AUC (Area Under the Curve)[34]

Pengklasifikasi yang sempurna akan sejajar di sudut kiri atas, menandakan Tingkat Positif Palsu (FPR) = 0 dan Tingkat Positif Benar (TPR) = 1. Pengklasifikasi kasus terburuk akan mendapat nilai di sudut kanan bawah di mana FPR = 1, TPR = 0. Pengklasifikasi acak diharapkan akan mendapat nilai di suatu tempat di sepanjang diagonal positif (garis putus-putus, di mana TPR = FPR) yang menandakan bahwa model akan menghasilkan prediksi positif palsu dan negatif palsu pada tingkat yang sama.

f) Kappa

Cohen's Kappa κ adalah metrik yang digunakan untuk mengukur pengaruh kebetulan pada akurasi model. Hal ini dilakukan dengan membandingkan akurasi yang sebenarnya dengan akurasi yang diharapkan. Menurut Fitzpatrick et al. [13], akurasi yang diharapkan adalah tingkat akurasi yang dapat dicapai oleh pengklasifikasi acak berdasarkan matriks kebingungan prediksi. Akurasi (A)

