BAB II
TINJAUAN PUSTAKA

2.1 Teori Prediksi Cuaca

Cuaca adalah keseluruhan rangkaian peristiwa di atmosfer bumi yang memainkan peran sentral dalam kehidupan sehari-hari manusia di seluruh dunia. Hal ini merujuk pada kondisi atmosfer di permukaan bumi yang dipengaruhi oleh faktor-faktor seperti tekanan udara dan suhu baik di langit maupun di permukaan bumi. Cuaca terbentuk melalui interaksi yang kompleks dari elemen-elemen seperti suhu udara, tingkat kelembapan, kecepatan angin, dan curah hujan . Saat cuaca bersifat cerah, langit tampak terang dengan sinar matahari yang menerangi tanpa memberikan sensasi panas yang berlebihan. Pada saat cuaca cerah, langit terlihat sebagian tertutup oleh awan atau bahkan separuh langit yang jelas, dan biasanya hujan tidak terjadi. Di sisi lain, cuaca berawan menandakan adanya banyak awan di langit yang bergerak karena dorongan angin. [13]. Cuaca memiliki kemampuan untuk mengalami perubahan dalam waktu yang singkat, bahkan dalam hitungan jam, dan ini tercermin dalam variasi antara siang dan malam. Perubahan cuaca disebabkan oleh fluktuasi suhu dan tingkat kelembaban di berbagai lokasi. Oleh karena itu, keberadaan prediksi cuaca yang komprehensif dan akurat menjadi sangat penting, karena dapat secara signifikan meningkatkan efisiensi dalam berbagai bidang kegiatan manusia. [14].

` Pentingnya prediksi cuaca menuntut adanya upaya terus-menerus dalam memahami dinamika atmosfer dan mengembangkan metode yang lebih efektif. Hal ini dilakukan agar kita dapat lebih siap menghadapi perubahan cuaca yang mungkin berdampak pada kegiatan sehari-hari, keselamatan, dan kesejahteraan masyarakat [15].Teori prediksi cuaca menggunakan *Machine Learning* adalah pendekatan yang menggabungkan konsep dan teknik dari ilmu *Machine Learning* dengan data cuaca historis untuk memprediksi kondisi cuaca di masa depan. Dalam konteks ini, Penggunaan *Machine Learning* dalam prediksi cuaca dapat meningkatkan akurasi dan ketepatan prediksi cuaca, terutama dalam jangka waktu jangka pendek. Namun, karena cuaca adalah fenomena alam yang kompleks dan

dipengaruhi oleh banyak variabel, prediksi cuaca yang sangat akurat dan andal dalam jangka waktu jangka panjang tetap menjadi tantangan. Oleh karena itu, penggunaan *Machine Learning* dalam prediksi cuaca harus tetap dikombinasikan dengan metode prediksi tradisional dan dukungan dari para ahli cuaca [16].

2.2 *Machine Learning* (ML)

Machine Learning adalah metode yang digunakan untuk membuat program yang bisa belajar dari data. Berbeda dengan program komputer biasa yang statis, program machine learning adalah program yang dirancang untuk mampu belajar sendiri. Cara belajar program machine learning mengikuti cara belajar manusia, yakni belajar dari contoh-contoh. *Machine learning* akan mempelajari pola dari contoh-contoh yang dianalisa, untuk menentukan jawaban dari pertanyaan-pertanyaan berikutnya. *Machine learning* adalah seperti membuat program yang bias menebak kotak hitam yang memiliki rumus fungsi yang belum diketahui. Kotak hitam itu diberikan sebuah input dan akan menghasilkan sebuah output tertentu, dari data-data input dan output yang diperoleh, maka program akan menebak rumus fungsi yang paling mendekati keakuratan [17].

Machine learning adalah pembelajaran mesin yang sangat membantu dalam menyelesaikan masalah, membuat mudah dalam mengerjakan sesuatu. Pengembangan teknologi dan dlam melakukan prediksi, *machine learning* membuat mudah dalam mengerjakan sesuatu, contohnya seorang peneliti cuaca bisa membuat suatu kemungkinan cuaca yang akan terjadi kedepannya dalam waktu cepat tanpa memakan waktu yang lama. *Machine learning* (ML) adalah bidang studi yang fokus kepada desain dan analisis algoritma sehingga memungkinkan komputer untuk dapat belajar. *Machine Learning* (ML) berisi sebuah algoritma yang bersifat umum dimana algoritma tersebut dapat menghasilkan sesuatu yang menarik atau bermanfaat dari sejumlah data tanpa harus menulis kode yang spesifik[18].

Pada intinya, algoritma yang generik tersebut ketika diberikan sejumlah data yang dapat membangun sebuah aliran aturan atau model atau interferensi dari data tersebut. *Machine Learning* (ML) digunakan untuk mengajari mesin cara

menangani data dengan lebih efisien. Tujuan pembelajaran mesin adalah untuk belajar dari data. Banyak penelitian telah dilakukan tentang bagaimana membuat mesin belajar sendiri tanpa diprogram secara eksplisit [19]. *Machine learning* juga dapat diartikan sebuah komputer yang memiliki kemampuan belajar tanpa diprogram secara spesifik. Program tersebut memanfaatkan data untuk membangun model dan mengambil keputusan berdasarkan model yang telah dibangun. *Machine learning* adalah satu program komputer yang dikatakan telah melakukan pembelajaran dari pengamalan E (*Experience*) terhadap tugas T (*Task*) dan mengukur peningkatan kinerja P (*Perfomance Measure*), jika kinerja Tugas T diukur oleh kinerja P, maka meningkatkan pengalaman E. Dari definisi ini dapat dikatakan sebuah aplikasi *machine learning* memiliki 3 komponen yaitu *Task* T, *Performance Measure* P, dan *Experience* E. Oleh karena itu, untuk membangun sebuah aplikasi *Machine Learning* (ML) maka komponen T,P dan E harus dapat diidentifikasi [20]. Prinsip cara kerja *machine learning* masih sama, meliputi pengumpulan data, eksplorasi data, pemilihan model atau teknik, memberikan pelatihan terhadap model yang dipilih dan mengevaluasi hasil dari machine learning. *Machine Learning* mempunya dua tipe yaitu [21] :

1. *Supervised Learning*

   Supervised learning adalah salah satu jenis pembelajaran mesin di mana model diberikan dataset dengan label yang sudah ditentukan untuk melatih model membuat prediksi terhadap data baru. Tujuannya adalah membuat model yang dapat membuat prediksi yang akurat pada data baru. Contohnya adalah *regresi linear*, pembelajaran dengan klasifikasi, dan pembelajaran dengan *clustering*.

2. *Unsupervised Learning*

   Unsupervised learning adalah jenis pembelajaran mesin di mana model diberikan *dataset* tanpa label dan tugas model adalah menemukan struktur atau pola dalam data. Tujuannya adalah membuat model yang dapat mengekstrak informasi yang berguna dari data tanpa bantuan label. Contohnya adalah klustering, reduksi dimensi, dan deteksi anomali.

2.3 Klasifikasi

Klasifikasi merupakan suatu proses pembelajaran yang bertujuan untuk mengelompokkan setiap set atribut ke label kelas yang sudah ada [22]. Proses klasifikasi ini mencari pola-pola atau fungsi-fungsi yang dapat menggambarkan dan memisahkan data dari kelas yang berbeda, sehingga dapat digunakan untuk memprediksi kelas data yang belum diketahui [23]. Untuk melakukan klasifikasi dan prediksi data, digunakan teknik *data mining*. Proses klasifikasi terdiri dari dua langkah. Pertama, tahap pembelajaran (*training*) di mana algoritma klasifikasi dianalisis berdasarkan data training untuk menghasilkan aturan klasifikasi. Kedua, tahap klasifikasi data uji digunakan untuk mengestimasi akurasi dari aturan klasifikasi yang telah dibuat [24].

Klasifikasi adalah salah satu peran penting dalam data mining. Klasifikasi termasuk dalam kategori *supervised learning* karena melibatkan proses pembelajaran dengan menggunakan data yang telah ada sebelumnya. Dalam proses ini, algoritma digunakan untuk mengidentifikasi pola dari data yang kemudian dapat diterapkan pada data baru yang belum diketahui kelasnya. Teknik klasifikasi banyak digunakan dalam berbagai bidang, seperti bidang medis, pendidikan, teknik bangunan, jaringan komputer, dan lainnya. Dalam melakukan klasifikasi, data yang telah ada sebelumnya diolah untuk menghasilkan aturan atau pengetahuan baru. Masalah klasifikasi pada dasarnya melibatkan langkah-langkah berikut [25] :

1. Masalah klasifikasi dimulai dengan menggunakan data *training* yang sudah tersedia.
2. Data *training* diproses dengan menggunakan algoritma klasifikasi.
3. Masalah klasifikasi berakhir dengan menghasilkan pengetahuan yang direpresentasikan dalam bentuk diagram, aturan, atau bentuk pengetahuan lainnya.

2.4 *Dataset*

Dataset adalah kumpulan data yang diambil dari berbagai sumber yang mewakili informasi dalam bentuk tabel dan hubungannya, dengan struktur yang

serupa dengan data yang ada dalam *database. Dataset* biasanya digunakan dalam berbagai konteks, termasuk dalam analisis data, penelitian ilmiah, pengembangan model prediksi, dan pembelajaran mesin. Penggunaan *dataset* memungkinkan analisis dan penarikan kesimpulan yang lebih baik dari data yang ada, serta membantu dalam mengembangkan model dan algoritma yang dapat memahami atau memprediksi pola dalam data tersebut.

Dataset yang baik harus memiliki struktur yang jelas, informasi yang akurat, serta representasi yang memadai dari entitas atau fenomena yang ingin dipelajari atau diprediksi. Terkadang, *dataset* dapat bersifat kecil dan dibuat untuk tujuan tertentu, atau bisa juga sangat besar, seperti dataset dalam *big data*, yang berisi jumlah data yang sangat besar dan kompleks [26].

2.5 *Python*

Python adalah bahasa pemrograman interpretatif yang dianggap mudah dipelajari serta berfokus pada keterbacaan kode, dengan kata lain, *Python* diklaim sebagai bahasa pemprograman yang memiliki kode-kode pemprograman yang sangat jelas, lengkap, dan mudah untuk dipahami. *Python* dianggap memiliki kehebatan untuk menangani pembuatan aplikasi- aplikasi kekinian yang mengadung kata kunci *big data, data mining, deep learning*, *data science,* hingga *machine learning*, dengan kata lain, *Python* adalah bahasa pemrograman simpel untuk pembuatan aplikasi berbasis kecerdasan buatan (*artificial intelligence*) [27].

Python secara umum berbentuk pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. *Python* dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi. Pada prinsipnya, *Python* dapat diperoleh dan digunakan secara bebas oleh siapa pun, bahkan bagi para *developer* yang menggunakan bahasa pemrograman ini untuk kepentingan komersial. Namun demikian, penggunaan *packages* atau module-module dari pihak ketiga, mungkin saja membutuhkan lisensi yang berbeda, misalnya lisensi berbayar [28].

Gambar 2.1 Bahasa Pemrograman *Python* [28]

2.5.1 Fitur Bahasa Pemrograman *Python*

Sisi utama yang membedakan *Python* dengan bahasa lain adalah dalam hal aturan penulisan kode program, bagi para programmer yang tidak terbiasa menggunakan *Python* akan dibingungkan dengan aturan indentasi, tipe data, *tuple*, dan *dictionary*. *Python* memiliki kelebihan tersendiri dibandingkan dengan bahasa lain terutama dalam hal penanganan modul, ini yang membuat beberapa programmer menyukai *Python*. Selain itu *Python* merupakan salah satu produk yang *open source*, gratis, dan *multiplatform*. Beberapa fitur dan kelebihan yang dimiliki *Python* adalah [29] :

1. Memiliki koleksi kepustakaan yang banyak. Artinya, telah tersedia modul-modul siap pakai untuk berbagai keperluan, seperti pembuatan game hingga *artificial intelligence* (misal: *Tensor Flow*).
2. Memiliki struktur bahasa yang jelas, sederhana, dan mudah dipelajari.
3. Berorientasi prosedural dan objek sekaligus (*multi-paradigma*)
4. Memiliki sistem pengelolaan memori otomatis *(garbage collection)* seperti halnya *Java*.
5. Bersifat modular sehingga mudah dikembangkan dengan menciptakan modul-modul baru, baik dibangun dengan bahasa *Python* maupun *C/C++* .

2.5.2 *Anaconda*

Anaconda adalah distribusi *Python* dan *R* yang dikhususkan untuk *Data Science* dan *Machine Learning*. *Anaconda* menyediakan lingkungan virtual yang membuat instalasi, pengaturan, dan manajemen paket *Python* dan *R* menjadi mudah

dan efisien. *Anaconda* juga menyediakan alat-alat untuk membuat, mengelola, dan berbagi lingkungan virtual yang terisolasi, memastikan bahwa setiap proyek memiliki versi paket yang tepat dan bahwa proyek-proyek tersebut saling independen.*Anaconda* menyediakan paket-paket populer seperti *NumPy, pandas, Matplotlib*, dan banyak paket lainnya yang dibutuhkan dalam bidang *Data Science* dan *machine learning*. *Anaconda* juga menyediakan akses ke berbagai distribusi dan pustaka *machine learning* seperti *scikit-learn, TensorFlow, PyTorch*, dll. *Anaconda* membuat proses pembelajaran dan implementasi teknologi *Data Science* dan *Machine Learning* menjadi lebih mudah dan efisien [30].

Gambar 2.2 Logo *Software* Anaconda [31]

Anaconda merupakan sebuah distribusi bahasa *pemrograman Python* dari *Continuum Analytics* yang berisi paket *Python* ditambah beberapa paket tambahan untuk keperluan komputasi ilmiah *(scientific computing*) seperti *data science, machine learning, data processing* skala-luas, analisis prediksi, dan lain sebagainya dalam satu distribusi *platform* yang *user friendly*. *Anaconda* diciptakan agar mempermudah pengguna me*management* paket *python*, dengan menggunakan *Anaconda*, maka versi dari paket yang ada, di manajemen oleh *package management system anaconda* [32].

2.5.3 *Jetbrains Pycharm*

PyCharm merupakan *text editor* atau *Integrated Development Environment* (IDE). *PyCharm* Edu merupakan *text editor* dengan tampilan *user interface* yang mudah dipahami sehingga mudah digunakan dalam tujuan pembelajaran. File *Python* menggunakan format .py [33].

IDE yang cukup populer dikalangan *developer Python* adalah *PyCharm*. *PyCharm* sendiri memiliki dua versi yaitu *Professional Edition* dan *Community Edition*. *PyCharm Professional Edition* merupakan versi berbayar dari *PyCharm* dan *Community Edition* merupakan versi gratis yang tersedia bagi komunitas *python* dengan lisensi *Apache 2*. [34]

Gambar 2.3 Logo *Pycharm*[34]

2.5.4 *Google Collab*

Google Colab adalah sebuah IDE untuk *pemrograman Python* dimana pemrosesan akan dilakukan oleh *server Google* yang memiliki perangkat keras dengan performa yang tinggi [35]. Dari sisi perangkat lunak, *Google Colab* telah menyediakan hampir sebagian besar pustaka (*library*) yang dibutuhkan. Beberapa pustaka yang tersedia dalam *Google Collab* adalah *Keras,TensorFlow, NumPy, Pandas*, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat *Matplotlib*. Bahkan seluruh versi disediakan, misalnya *TensorFlow* versi 1.x maupun 2.x tersedia, begitu juga versi *Python* yang mulai dari versi 2.x hingga 3.x. Dari sisi perangkat keras [36].

GoogleColab menyediakan layanan berupa media penyimpan yang terintegrasi dengan *Google Drive*, prosesor yang berupa CPU, GPU, dan TPU, serta RAM, dengan jaminan kemampuan servernya yang stabil hampir

keseluruhan pemrosesan tidak menemukan kendala dengan *Google Colab* selama koneksi jaringan internet lancar [37].

Gambar 2.4 Logo *Google Collab*[38]

2.6 *Pandas*

Pandas adalah sebuah pustaka (*library*) *open-source* yang digunakan dalam bahasa pemrograman *Python* untuk analisis data dan manipulasi data terstruktur. Pustaka ini menyediakan struktur data dan fungsi-fungsi yang memungkinkan pengguna untuk dengan mudah melakukan berbagai operasi pada data, seperti membersihkan, transformasi, penyajian, dan analisis.

Pandas menyediakan struktur data yang digunakan dalam bahasa pemrograman *python*, memiliki pustaka *open source* berlisensi BSD dan analisis data berkinerja tinggi. Tahun 2008 Wes McKinney menginisiasi *Pandas* saat berada di lokasi AQR *Management Capital*. *Pandas* sering digunakan untuk membaca data atau memanggil data dari berbagai macam jenis data khususnya tekstual. Untuk memanggil fungsi *pandas* menggunakan *script* "*Import pandas as pd"* [39]

2.7 Sklearn

Sklearn atau *Scikit-learn* adalah *library python* yang berfokus kepada *machine learning.* Pada library ini disediakan banyak fungsi yang membantu dalam pembuatan model atau melakukan pemprosesan data, seperti regresi, klasifikasi, *clustering*, dan lainnya [40] .Pada riset ini fungsi yang digunakan pada

library ini adalah *sklearn.naive_bayes,,sklearn.model_selection, sklearn.metrics ,sklearn.model_selection, sklearn.preprocessing, sklearn.feature_selection.*

2.8 Seleksi Fitur

Seleksi Fitur adalah proses untuk mengurangi jumlah fitur yang digunakan pada saat membentuk model. Pengurangan fitur dilakukan untuk mengurangi waktu komputasi dan juga pada beberapa kasus meningkatkan akurasi [41].

Seleksi fitur secara statistik dilakukan dengan menghitung dan melakukan evaluasi antara tiap fitur dengan label. Apabila fitur tersebut memiliki nilai yang tinggi terhadap labelnya maka fitur tersebut dapat dikatakan berhubungan dengan label tersebut, sehingga terpilih. Cara ini dapat dilakukan dengan cepat dan efektif, tetapi metode yang digunakan bergantung kepada fitur dan labelnya.

Secara umum ada 2 teknik seleksi fitur, yaitu *suprvised* dan *unsupervised. Supervised* adalah teknik seleksi fitur dimana label digunakan dalam penentuan apakah fitur tersebut digunakan atau tidak. Pada teknik ini fitur diseleksi berdasarkan pengaruhnya terhadap label. *Unsupervised* adalah teknik dimana label tidak berpengaruh dengan seleksi fitur, melainkan antara satu fitur dan fitur lain. Teknik ini digunakan untuk melakukan seleksi terhadap fitur yang redundan, dengan menghitung korelasinya [42].

2.8.1 Seleksi Fitur ANOVA-*f Test*

ANOVA atau *Analysis of Variance* adalah metode statisik untuk mengambil suatu kesimpulan berdasarkan data atau kelompok statustuk inferentif. Pada seleksi fitur, ANOVA dapat mengevaluasi dan melakukan scoring fitur terhadap keterkaitannya dengan label yang ada. Setiap fitur diberikan skor dan diranking. Perhitungan yang didapat menggunakan ANOVA disebut dengan *f-ratio*. Semakin tinggi *f-ratio* tersebut maka semakin terpisah label yang ada [43].

Skor dari tiap fitur dapat dihitung menggunakan rumus berikut :

$$\sigma_{cl}^2 = \frac{\sum(\bar{x}_i - \bar{x})^2 n_i}{(k-l)} \tag{2.1}$$

Rumus pertama adalah menghitung jarak antar kelas. Jarak antar kelas dapat dihitung menggunakan rumus (1), dimana n_i adalah jumlah kelas *i* muncul dalam set, $\bar{x}_i$ adalah rata – rata dari kelas *i*, dan $\bar{x}$ adalah rata – rata dari fiturnya.

$$\sigma_{err}^2 = \frac{\left(\sum\sum\left(x_{ij} - \bar{x}\right)^2\right) - \left(\sum(\bar{x}_i - \bar{x})^2 n_i\right)}{(k - l)} \tag{2.2}$$

Selanjutnya dihitung jarak didalam kelas. Rumus ini mirip dengan rumus analisis varian. Hasil penjumlahan kuadrat nilai perkelas pada fitur tersebut dikurangi rata – rata fitur tersebut, dikurangi dengan penjumlahan hasil kuadrat rata – rata kelas tersebut dikurangi dengan rata – rata fitur.

$$f\ ratio = \frac{\sigma_{cl}^2}{\sigma_{err}^2} \tag{2.3}$$

Pada akhirnya kita mendapatkan skor untuk fitur tersebut dengan membagi jarak antara kelas dengan jarak didalam kelas, semakin tinggi skor tersebut maka semakin relevan fitur tersebut terhadap labelnya.

2.9 *Hyperparameter Tuning GridSearchCV*

Tujuan *hyperparameter tuning* adalah untuk mengetahui parameter-parameter yang dimasukkan ke dalam model yang dapat menghasilkan performa yang paling optimal. Sementara, *cross-validation grid serach* (*GridSearchCV*) digunakan untuk mengoptimalkan nilai akurasi. *GridSearchCV* merupakan bagian dari modul *scikit-learn* yang melakukan validasi untuk lebih dari satu model serta menyediakan *hyperparameter* masing-masing secara otomatis dan sistematis [44].

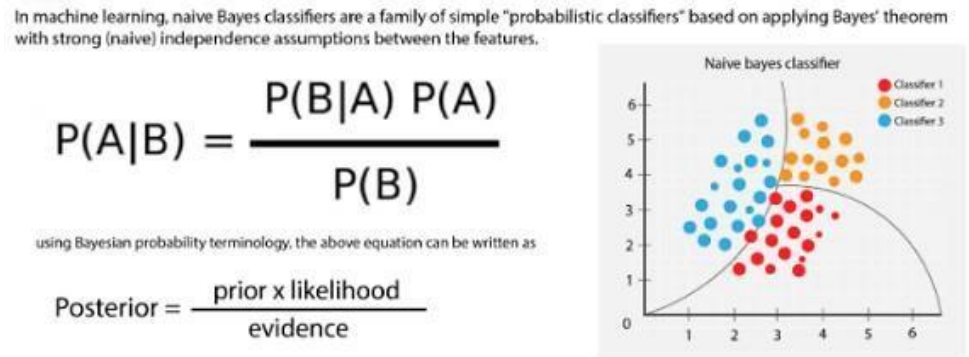
Setelah dilakukan *hyperparameter tuning*, maka akan diketahui parameter-parameter yang diprediksi, sehingga dapat memberikan performa yang optimal pada model yang dibangun. Oleh karena itu, perlu disimpan supaya mempermudah proses evaluasi model. Pada tahap ini dilakukan pembangunan model menggunakan parameter-parameter yang telah disimpan sebelumnya, lalu model tersebut akan menyesuaikan antara data yang menjadi *features* dan *class-label*, biasanya ini diketikkan pada baris kode sebagai *model.fit()*. Lalu dilakukan evaluasi terhadap model yang telah dibangun dengan cara menjalankan skenario

pengujian yang telah dirancang, serta menghitung dan visualisasi *performance metrics* untuk sistem klasifikasi [45].

2.10 *Naive Bayes*

Naive Bayes merupakan algoritma klasifikasi yang berlandaskan pada probabilitas, digunakan untuk memproyeksikan kategori suatu entitas berdasarkan fitur-fitur yang dimilikinya. Algoritma ini mengambil dasar pada asumsi independensi fitur, yakni anggapan bahwa setiap fitur berdiri sendiri secara independen dari yang lain. Meskipun asumsi ini tidak selalu sepenuhnya relevan dalam konteks dunia nyata, namun hal ini menjadikan perhitungan probabilitas lebih sederhana dan efisien. Algoritma *Naive Bayes* umumnya dimanfaatkan dalam berbagai aplikasi, seperti penapisan spam, klasifikasi dokumen, serta analisis sentimen. [46].

Naïve Bayes adalah pengklasifikasi linier yang dikenal sederhana namun sangat efisien. Model probabilistik pengklasifikasi *Naive Bayes* didasarkan pada teorema *Bayes*, dan kata sifat *Naive* berasal dari asumsi bahwa fitur dalam kumpulan data saling independen. Dalam praktiknya, asumsi independensi sering dilanggar, tetapi pengklasifikasi *Naive Bayes* masih cenderung berkinerja sangat baik di bawah asumsi yang tidak realistis ini Namun, pelanggaran kuat terhadap asumsi independensi dan masalah klasifikasi *non-linier* dapat menyebabkan kinerja pengklasifikasi *Naive Bayes* yang sangat buruk [47].



Gambar 2.5 *Naïve Bayes* [48]

2.10.1 *Gaussian Naive Bayes*

Gaussian Naive Bayes adalah algoritma klasifikasi probabilitas berdasarkan teorema *Bayes* dengan asumsi fitur-fitur memiliki distribusi normal (*Gaussian*). Ini merupakan variasi dari algoritma *Naive Bayes* dan umumnya digunakan untuk tugas klasifikasi ketika berhadapan dengan data numerik kontinu. Berikut ini adalah penjabaran komponen utamanya [49]:

1. Distribusi *Gaussian*, juga dikenal sebagai distribusi normal, adalah distribusi probabilitas kontinu yang ditandai oleh *mean* (μ) dan variansi (σ^2). Dalam konteks ini, asumsinya adalah fitur-fitur numerik dari data mengikuti distribusi *Gaussian*.
2. Bagian *"Naïve"* dari nama algoritma merujuk pada asumsi independensi fitur. Asumsinya adalah keberadaan atau nilai suatu fitur bersifat independen dari keberadaan atau nilai fitur lainnya, dengan mempertimbangkan label kelas. Ini merupakan penyederhanaan yang membantu membuat algoritma efisien secara komputasi.

   Berikut ini adalah bentuk pendistribusian dari *Gaussian Naive Bayes*:

   a. Tentukan data latih

   b. Menghitung jumlah dan probabilitas, jika terdapat data numerik, maka:

   c. Tentukan nilai mean dan standar deviasi dari masing-masing parameter yang termasuk data numerik.

   d Tentukan nilai probabilistik dengan cara menghitung jumlah data yang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.

   e. Menemukan nilai probabilitas

2.11 Model Evaluasi

2.11.1 *Confusion Matrix*

Confusion matrix adalah alat evaluasi yang digunakan dalam analisis klasifikasi untuk mengukur kinerja model dengan membandingkan hasil prediksi model dengan nilai sebenarnya dari data uji. *Confusion matrix* dapat membantu

memahami sejauh mana model klasifikasi mampu memprediksi dengan benar dan di mana model tersebut membuat kesalahan [50].

Confusion matrix berbentuk tabel dengan empat sel utama [51]:

- True Positive (TP): Jumlah data yang benar-benar positif dan juga diprediksi sebagai positif oleh model.
- True Negative (TN): Jumlah data yang benar-benar negatif dan juga diprediksi sebagai negatif oleh model.
- False Positive (FP): Jumlah data yang sebenarnya negatif tetapi diprediksi sebagai positif oleh model (error Type I).
- False Negative (FN): Jumlah data yang sebenarnya positif tetapi diprediksi sebagai negatif oleh model (error Type II).

Tabel 2.1. *Confusion matrix*

		Prediksi Label	
		1	**0**
Aktual Label	**1**	**TP**	**FN**
	0	**FP**	**TN**

Pada table 2.1 dengan menggunakan nilai-nilai *confusion matrix* tersebut, dapat menghitung berbagai metrik evaluasi seperti akurasi, presisi, *recall* (sensitivitas), dan F1-*score* dengan rumus umum dibawah ini [52] :

1. **Akurasi (*Accuracy*):**

   Akurasi mengukur sejauh mana model mampu memprediksi secara benar baik kelas positif maupun kelas negatif. Ini adalah perbandingan dari total prediksi benar (TP dan TN) dengan jumlah total data uji.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.4)$$

2. **Presisi (*Precision*):**

   Presisi mengukur sejauh mana prediksi positif model adalah benar. Ini adalah perbandingan dari jumlah prediksi benar positif (TP) dengan total jumlah prediksi positif (TP + FP).

$$Precision = \frac{TP}{TP + FP} \quad (2.5)$$

3. ***Recall* :**

   Recall mengukur sejauh mana model mampu menemukan semua contoh positif yang sebenarnya. Ini adalah perbandingan dari jumlah prediksi benar positif (TP) dengan total jumlah contoh positif sebenarnya (TP + FN).

$$recall = \frac{TP}{TP+FN} \quad (2.6)$$

4. **F1-*Score*:**

   F1-*score* adalah harmonik rata-rata antara presisi dan *recall*. F1-score memberikan gambaran lebih baik tentang kinerja model ketika ada keseimbangan antara presisi dan *recall*.

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.7)$$

Tabel 2.2 Penelitian Terdahulu Tentang Penggunaan Algoritma *Machine Learning* Untuk Prediksi Prediksi Cuaca

Penulis	**Judul**	**Tahun Jurnal**	**Metodologi /Model**	**Hasil**
Alvi Syahrini Utami , Dian Palupi Rini , Endang Lestari	Prediksi Cuaca di Kota Palembang Berbasis *Supervised Learning* Menggunakan Algoritma *K-Nearest Neighbour*	2021	*Supervised Learning* Menggunakan Algoritma *K-Nearest Neighbour*	tingkat keakuratan dari *hybrid* metode ini adalah sekitar 70%
Amril Mutoi Siregar, Tukino, Sutan Faisal, Ahmad Fauzi,dan Ilman Kadori	Klasifikasi untuk Prediksi Cuaca Menggunakan *Esemble Learning*	2020	*Esemble Learning*	Akurasi yang didapatkan 81.21% dan MSE 18.79%.
Oshodi	*Machine Learning-based Algorithms for Weather Forecasting*	2022	*Gaussian Naïve Bayes*	Algoritma *Gaussian Naive Bayes* terbukti menjadi algoritma dengan performa terbaik dengan akurasi prediksi sebesar 84.153 %

Amril Mutoi Siregar , Sutan Faisal, Yana Cahyana, Bayu Priyatna	Perbandingan Algoritme Klasifikasi Untuk Prediksi Cuaca	2020	*Naive Bayes, Decision Tree, Random Forest*	Sistem prediksi yang telah dibuat mendapatkan tingkat akurasi *Naïve Bayes* sebesar 77.22% ,tingkat akurasi *Decision Tree* sebesar 79.46% , tingkat akurasi *Random forest* sebesar 82.38%
Muhammad Yusuf Rizqon, Muhamamd Valensyah Alfansyuri, Wawan Gunawan	Penerapan Algoritma *K-Nearest Neighbor* (KNN) Dalam Memprediksi Dan Menghitung Tingkat Akurasi Data Cuaca Di Indonesia		Algoritma *K-Nearest Neighbor* (KNN)	menghasilkan prediksi dengan tingkat akurasi data sebesar 0,8993 atau sekitar 89%.