Metode Penelitian

Fransisco Ready Permana

1910511045 Informatika B

DOSEN : Jayanta, S.Kom., M.Si

**LAPORAN TUGAS INDIVIDU METODE PENELITIAN**

Tema : Machine Learning Supervised Learning

Judul : Penerapan Algoritma *Extreme Gradient Boosting* Untuk

Mengklasifikasikan Cuaca Provinsi DKI Jakarta Berdasarkan Data Cuaca Provinsi DKI Jakarta Tahun 2018

Permasalahan : Cuaca Provinsi DKI Jakarta Tahun 2018 yang mengalami perubahan

secara mendadak

Tujuan Penelitian : Melakukan Pengelompokan cuaca serta menghitung nilai akurasi

klasifikasi algoritma *Extreme Gradient Boosting* terhadap data cuaca Provinsi DKI Jakarta tahun 2018

Tehnik Sampling : Menggunakan tehnik Sampling Bertahap yang menggabungkan tehnik

Sampling Acak Stratifikasi dengan tehnik sampling acak sederhana dimana tehnik sampling yg digunakan akan mengambil sampel secara acak namun dikelompokkan ke dalam klasifikasi yang berbeda antara Hujan, Berawan dan Cerah secara seimbang

Pengambilan Data : Teknik pengambilan data secara kuantitatif dengan melakukan studi

dokumen yaitu mengambil dataset statistik cuaca Provinsi DKI Jakarta yang diambil dari website open data Indonesia

Metode Penelitian :

* Mengambil data cuaca Provinsi DKI Jakarta tahun 2018 pada website Satu Data Indonesia dari bulan januari hingga desember lalu menggabungkannya menjadi 1 file
* Melakukan tahap pre-processing data atau analisis data
* Melihat persebaran data dengan melihat visualisasi data
* Melakukan tehnik sampling dengan membagi data menjadi data latih sebesar 80% dan data uji sebesar 20% dari jumlah data dengan pembagian secara stratifikasi
* Melakukan modeling data menggunakan Algoritma *Extreme Gradient Boosting*
* Melakukan uji Tunning terhadap parameter yang digunakan pada algoritma *Extreme Gradient Boosting* untuk mencari parameter terbaik yang akan digunakan
* Melakukan evaluasi terhadap model yang telah dibuat
* Melihat hasil akurasi yang didapatkan berdasarkan hasil evaluasi

Dataset : Mengambil data pada open dataset indonesia

|  |
| --- |
| <https://katalog.satudata.go.id/dataset/data-prakiraan-cuaca-wilayah-provinsi-dki-jakarta-tahun-2018> |

Contoh Bentuk Data :

|  |  |
| --- | --- |
| Variabel Dataset | Pengertian |
| tanggal | Tanggal pengambilan data pada Provinsi DKI Jakarta tahun 2018 |
| waktu | Waktu saat pengambilan data apakah data diambil pada pagi, siang, sore, malam dan lain sebagainya. |
| cuaca | Cuaca hasil pengamatan pada Provinsi DKI Jakarta |
| kelembaban\_persen | Tingkat kelembaban wilayah tersebut ketika pengamatan data cuaca dilakukan |
| suhu\_derajat\_celcius | Suhu wilayah tersebut ketika pengamatan data cuaca dilakukan |

****

**UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA**

**PENERAPAN ALGORITMA *EXTREME GRADIENT BOOSTING* UNTUK MENGKLASIFIKASIKAN CUACA PROVINSI DKI JAKARTA BERDASARKAN** **DATA CUACA PROVINSI DKI JAKARTA TAHUN 2018**

**PROPOSAL PENELITIAN**

**MATA KULIAH METODE PENELITIAN**

**Fransisco Ready Permana**

**1910511045**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAKARTA**

**2022**

**BAB I**

**PENDAHULUAN**

1. **Latar Belakang**

Cuaca adalah keadaan udara yang pada satu wilayah tertentu dengan jangka waktu terbatas. Cuaca juga bisa diartikan sebagai kondisi temperatur, cahaya matahari kelembapan, kecepatan angin, dan sebagainya dalam waktu yg singkat. Keadaan cuaca juga berbeda-beda setiap waktu dan tempat. Penyebutan cuaca biasanya diikuti oleh waktu tertentu dan tempat tertentu contohnya cuaca pada pagi hari di Jakarta, siang hari di Surabaya, sore hari di Malang dan lain sebagainya. keadaan cuaca juga bisa berbeda-beda pada setiap tempat dan jamnya (NUR ROCHMAN DARMAWAN, 2019). Namun, Seiring meningkatnya suhu global, maraknya pembalakan hutan, meningkatnya pembakaran karbon dan lain sebagainya, cuaca di seluruh dunia dan di kota-kota besar seperti di Jakarta berubah dengan cepat dan terjadi secara terus menerus. Prakiraan yang tepat sangat penting untuk kehidupan di Ibu Kota karena kegiatan bisnis, usaha kecil menengah sampai pekerjaan sehari-hari sangat bergantung pada prakiraan cuaca.

Banyak pendekatan untuk melakukan prakiraan cuaca Salah satunya adalah dengan pendekatan *machine learning*. *Machine Learning* adalah bagian dari kecerdasan buatan yang bertujuan untuk mengoptimalkan kinerja sistem dengan mempelajari data sampel yangdibangun di atas ilmu statistik, matematika, dan penambangan data sehingga mesin dapat belajar menganalisis data tanpa harus diprogram secara khusus. *Machine Learning* dapat digunakan untuk mendapatkan wawasan dari berbagai jenis data dari data berlabel hingga tidak berlabel, data terstruktur hingga data tidak terstruktur, data regresi maupun klasifikasi dan lain sebagainya dapat ditangani menggunakan *machine learning*

Untuk melakukan prediksi cuaca dengan pendekatan *machine learning,* kita perlu untuk mengklasifikasikan data berdasarkan labelnya masing-masing. Kemudian melatih data menggunakan berbagai algoritma yang ada seperti Support Vector Machine, Fuzzy Logic, Genetic Algorithms, dan Artificial Neural Network, Random Forest, *Extreme Gradient Boosting* dan lain sebagainya. Dari beberapa contoh algoritma yang telah disebutkan, algoritma *Extreme Gradient Boosting* adalah metode yang paling baru dan paling andal untuk menangani berbagai bentuk data dan dapat memberikan hasil yang bagus (Shafila, 2020).

Algoritma *Extreme Gradient Boosting* (XGB) merupakan salah satu metode untuk membuat mesin dapat digunakan untuk mengklasifikasikan data dan andal untuk menangani dalam jumlah yang besar (Mursianto, 2021a). XGB memanfaatkan konsep *ensemble learning* dari pohon keputusan (*Decision* *Tree*) yang dibangun dengan menerapkan metode boosting. Metode boosting sendiri merupakan sebuah metode yang digunakan untuk meningkatkan performa dari suatu algoritma *weak learner.* Berdasarkan hal yang telah disebutkan sebelumnya, pemilihan algoritma *Extreme Gradient Boosting* untuk mengklasifikasikan cuaca adalah pilihan yang baik.

Untuk mendapatkan performa yang terbaik dari algoritma *Extreme Gradient Boosting*, perlu dilakukannya optimasi atau tuning hyper parameter. Tujuannya adalah untuk bisa menyesuaikan algoritma tersebut dengan data yang dimiliki sehingga, model yang dibuat memiliki performa terbaik dan dapat mengklasifikasikan cuaca dengan baik.

1. **Rumusan Masalah**

Berdasarkan uraian dari latar belakang masalah diatas maka diperoleh rumusan masalah sebagai berikut:

1. Seberapa akurat algoritma *Extreme Gradient Boosting* untuk mengklasifikasikan cuaca berdasarkan data cuaca Provinsi DKI Jakarta tahun 2018?
2. Bagaimana nilai dari hyperparameter yang digunakan *extereme gradient boosting* setelah optimasi?
3. **Tujuan Penelitian**

Adapun tujuan penelitian ini berdasarkan rumusan masalah sebagai berikut:

1. Memperoleh nilai akurasi dari penerapan *Extreme Gradient Boosting* dalam mengklasifikasikan cuaca berdasarkan data cuaca Provinsi DKI Jakarta tahun 2018,
2. Mendapatkan nilai dari hyperparameter terbaik yang dihasilkan dari model *Extreme Gradient Boosting* untuk menghasilkan nilai akurasi yang tinggi setelah dilakukannya optimasi.
3. **Manfaat Penelitian**

Adapun manfaat penelitian yang ingin dicapai antara lain :

1. Bagi Peneliti

Mendapatkan wawasan ilmu serta pengalaman dalam melakukan pembuatan model machine learning dengan menggunakan algoritma *Extreme Gradient Boosting.*

1. Bagi Mahasiswa Statistika dan Informatika

Memberikan referensi penelitian *machine learning* berkaitan dengan klasifikasi cuaca dengan dataset cuaca Provinsi DKI Jakarta tahun 2018 menggunakan algoritma *Extreme Gradient Boosting* kepada penelitian selanjutnya.

1. Bagi Penelitian Selanjutnya

Ikut serta mengimpelentasikan algoritma klasifikasi *Extreme Gradient Boosting* serta menjadi pembanding untuk metode klasifikasi lainnya

1. Bagi Masyarakat

Memberikan informasi berkaitan dengan prediksi cuaca sehingga bisa menjadi pertimbangan masyarakat dalam melakukan kegiatan sehari-hari.

1. **Ruang Lingkup**

Untuk menghindari kerancuan dalam proses penelitian, maka diberikan ruang lingkup penelitian sebagai berikut :

1. Data yang digunakan untuk membuat model machine learning adalah data cuaca Provinsi DKI Jakarta tahun 2018 yang diambil dari laman www.katalog.go.id,
2. Menggunakan Bahasa pemrograma pyton untuk memproses dan membangun model *machine learning,*
3. Penelitian ini ditujukan untuk mencari tahu performa terbaik dari algoritma *Extreme Gradient Boosting* terhadap data cuaca Provinsi DKI Jakarta tahun 2018,
4. Penelitian ini menggunakan metode klasifikasi pada supervised learning yang bertujuan untuk mengelompokkan cuaca pada Provinsi DKI Jakarta tahun 2018.
5. **Luaran Yang Diharapkan**

Luaran yang diharapkan dari penelitian ini adalah mendapatkan model dengan nilai akurasi yang tinggi dengan algoritma*Extreme Gradient Boosting* serta dapat menentukan nilai dari hyperparamerter yang sesuai untuk data yang digunakan.

1. **Sistematika Penulisan**

BAB I : PENDAHULUAN

Bab ini mengemukakan latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, ruang lingkup penelitian, luaran yang diharapkan, dan sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Bab ini menjelaskan tentang landasan teori yang mendukung dalam pelaksanaan penelitian serta penelitian terkait.

BAB III : METODOLOGI PENELITIAN

Bab ini menjelaskan tentang alur penelitian, tahapan penelitian, waktu dan tempat penelitian, serta jadwal penelitian untuk dilaksanakan.

DAFTAR PUSTAKA

Daftar pustaka berisikan daftar bacaan yang menjadi sumber referensi yang digunakan di penelitian ini.

LAMPIRAN

Lampiran memuat informasi tambahan yang digunakan di penelitian ini.

**BAB II**

**TINJAUAN PUSTAKA**

1. **Cuaca**

Cuaca adalah keadaan dinamika udara di atmosfer pada waktu dan tempat tertentu. Cuaca umumnya dapat diungkapkan atau dinyatakan dengan kondisi hujan, suhu udara, jumlah tutupan awan, penguapan, kelembaban, dan kecepatan angin di suatu tempat dari hari ke hari. Kurun waktu yang sering digunakan dalam analisa cuaca adalah satu hari sampai satu minggu (Aldrian, Karmini and Budiman, 2011).

Secara umum, cuaca terdiri dari tujuh unsur sebagai berikut:

* Suhu udara, biasanya dinyatakan dalam satuan °C untuk mengetahui derajat panas atau dingin udara.
* Tekanan udara atau berat massa udara per satuan luas. Biasanya dinyatakan dalam satuan mb (milibar).
* Kelembaban udara atau kandungan uap air di udara yang dinyatakan dalam %.
* Penguapan atau laju uap air yang terbentuk akibat proses pergantian fase cair menjadi uap. Biasanya dinyatakan dalam satuan mm (milimeter).
* Awan atau kumpulan massa tampak yang terdiri dari udara kering dengan butir air atau kristal es pada ketinggian tertentu di atas muka Bumi. Biasanya dinyatakan dalam satuan octaf.
* Hujan atau laju air yang jatuh dari awan ke muka Bumi yang dinyatakan dalam satuan mm.
* Angin atau udara yang bergerak. Biasanya dinyatakan dalam satuan knot untuk kecepatan. Sedangkan untuk arah digunakan mata angin (utara, selatan, barat, timur, tenggara, barat laut, dan barat timur) (Aldrian, Karmini and Budiman, 2011).

1. **Machine Learning**

*Machine learning* merupakan teknik melakukan inferensi terhadap data dengan membuat model (matematis) yang merefleksikan pola-pola data. *Machine learning* merupakan ilmu pengetahuan berbentuk matematis karena pada dasarnya diturunkan dari machine learning, statistika, dan komputasi. *Machine learning* menjadi semakin populer akibat meningkatnya kemampuan komputasi dengan mesin dan komputer. *Machine learning* menjadi teknik yang digunakan untuk melakukan inferensi pada data yang relatif besar karena konstruksi model inferensi dapat dilakukan secara otomatis. *Machine learning* merupakan sebuah alat dimana penggunaannya tergantung pada domain permasalahan sehingga diperlukannya pengetahuan teknik-teknik machine learning yang sesuai pada domain yang diteliti (Putra, 2020a).

1. **Supervised Learning**

Secara harfiah *Supervised learning* adalah pembelajaran terarah/terawasi. *Supervised learning* merupakan salah satu pendekatan dalam machine learning dimana machine learning dilatih untuk mengenali pola yang diberikan agar dapat menemukan jawaban yang sesuai dengan hasil pola tersebut. *Supervised learning* digambarkan seperti siswa yang belajar di sekolah, di mana guru memberikan contoh angka kepada siswanya, kemudian guru memberikan cara membaca yang benar untuk masing-masing angka. Contoh angka melambangkan *input*, kemudian cara membaca melambangkan *desired output*. Pasangan input-desired output ini disebut sebagai instance. Metode pembelajaran ini disebut *supervised* karena ada yang memberikan contoh jawaban (desired output) (Putra, 2020b).

Tujuan *supervised learning* adalah untuk melakukan klasifikasi. Terdapat 3 macam klasifikasi yaitu *binary classification, multi-class classification,* dan *multi-label classification*. *Binary classification* jika hanya ada dua kategori yang ingin diklasifikasikan. *Multi-class classification* jika terdapat lebih dari dua kategori yang ingin diklasifikasikan. *Multi-label classification* jika ingin mengklasifikasikan suatu sampel ke dalam suatu himpunan kelas (Putra, 2020b).

1. **Klasifikasi**

Dalam *machine learning*, klasifikasi (*classification*) yaitu mencari kelas yang berkorespondensi terhadap suatu *input*. Contoh dari klasifikasi adalah mengkategorikan gambar sayuran seperti seledri, cabai, dsb (Putra, 2020c).

Jika klasifikasi hanya terdiri dari dua kelas, klasifikasi datra ini disebut *binary classification*. Apabila kelas klasifikasi lebih dari dua kelas (*mutually exclusive*), disebut *multi-class classfication*. Apabila kelas-kelas tersebut tidak bersifalt *mutually exclusive*, maka disebut *multi-label classification*. Selanjutnya harap bedakan antara *multi-label classification* dan *multi-level/hierarchical classification*. Pada *multi-level/hierarchical classification*, pertama dilakukan klasifikasi untuk suatu kelas generik, lalu dilanjutkan mengklasifikasikan data ke kelas yang lebih spesifik.

*Multi-class classification* juga disebut *hard classification*, artinya setelah data diklasifikasikan ke kelas tertentu, maka tidak mungkin datra berada di kelas lainnya. *Multi-label classification* bewrsifat lebih *soft*, karena dapat mengklasifikasikan ke beberapa kelas, misal data X masuk ke kategori kelas A, B, dan C sekaligus (dengan nilai probabilitas masing-masing) (Putra, 2020c).

1. **Ensemble Learning**

*Ensemble learning* merupakan salah satu algoritma dari *supervised learning*, karena dapat dilakukan dilatih dan kemudian dilakukan untuk membuat prediksi. Oleh karena itu, *ensemble learning* terlatih mewakili hipotesis tunggal. Hipotesis ini tidak selalu terkandung dalam ruang hipotesis model dari mana ia dibangun. Dengan demikian *ensemble learning* dapat terbukti memiliki lebih banyak fleksibilitas dalam fungsi yang dapat direpresentasikan. Fleksibilitas ini memungkinkan *ensemble learning* untuk melakukan ***overfitting***. Secara empiris, *ensemble learning* cenderung menghasilkan hasil yang lebih baik ketika ada yang signifikan keragaman di antara model (Ying, 2014).

1. **Boosting**

Boosting merupakan salah satu metode ensemble yang digunakan untuk mengimprovisasi performa suatu algoritma learning dengan mengkombinasikan kumpulan classifier lemah guna membentuk suatu classifier akhir yang kuat. Boosting mengasumsikan ketersediaan dari suatu algoritma learning yang lemah (suatu algoritma learning yang menghasilkan model classifier lemah atau model yang kurang akurat dalam memprediksi suatu training). Ide utama didalam proses boosting yaitu memilih sekumpulan data training (sampel training) dengan beberapa cara untuk kemudian dipelajari oleh suatu base learner, dimana base learner tersebut dipaksa menarik sesuatu yang baru tentang sampel tersebut setiap kali base learner itu dipanggil. Proses ini dapat dicapai dengan memilih sampel training yang diharapkan dapat membuat performa dari base classifier menjadi sangat buruk bahkan lebih buruk dari pada performa base classifier secara reguler. Jika hal ini dapat dicapai, kemudian pada iterasi selanjutnya diharapkan base learner dapat menghasilkan suatu base classifier baru yang secara signifikan berbeda dari pendahulunya. (Schapire dan Freund, 2012).

Algoritma boosting adalah algoritma iteratif yang memberikan bobot yang berbeda pada distribusi training data pada setiap iterasi. Setiap iterasi boosting menambahkan bobot pada contoh-contoh kesalahan klasifikasi dan menurunkan bobot pada contoh klasifikasi yang benar, sehingga secara efektif dapat merubah distribusi pada data training (S. Kotsiantis, Kanellopoulos, & Pintelas, 2006). Metode Boosting (AdaBoost) yang diusulkan (S. B. Kotsiantis & Pintelas, 2009) dengan selective costing ensemble dapat menjadi solusi yang lebih efektif untuk masalah class imbalance dan memungkinkan untuk meningkatkan identifikasi dari class minoritas yang sulit serta menjaga kemampuan klasifikasi dari class mayoritas.Pendekatan untuk mengatasi masalah tersebut dapat dilakukan dengan beberapa metode yaitu over-sampling, under-sampling, dan cost-sensitive. Adapun skema teknik boosting dapat dilihat pada dibawah ini



Gambar 2.1 Skema Teknik Boosting

Pada prinsipnya Boosting membentuk satu classifier yang kuat dengan mengkombinasikan sekumpulan classifier. Boosting mempertahankan sekumpulan bobot pengamatan pada saat training pengamatan dan secara adaptif menyesuaikan (updating) bobot-bobot ini pada akhir tiap iterasi boosting. Bobot-bobot dari pengamatan yang salah terklasifikasikan pada saat training akan dinaikkan sementara bobot-bobot pengamatan yang diklasifikasikan dengan benar akan diturunkan nilainya (Li dkk, 2008). Dengan kata lain, Boosting memaksa suatu classifier untuk memberi perhatian yang lebih pada pengamatan yang salah diklasifikasikan (Garcia dan Lozano, 2007). Boosting dilakukan secara sekuensial dengan membangun beberapa DT dengan pembelajar lemah secara gradient dengan memanfaatkan data residu untuk menumbuhkan DT berikutnya

1. ***Extreme Gradient Boosting***

Extreme Gradient Amplification adalah proyek open source yang diusulkan oleh Chen dan Guestrin pada tahun 2016 yang menerapkan algoritma *gradient tree boosting* untuk mengatasi masalah pembelajaran mesin. XGBoost dikembangkan dari Classification and Regression Trees (CART) atau dengan nama lain Decision Trees. Metode ini merupakan gabungan dari dua jenis pohon, yaitu pohon klasifikasi dan pohon regresi. Jika variabel terikat bertipe kategorikal, CART akan menghasilkan pohon klasifikasi, sedangkan jika variabel terikat bertipe kontinu atau numerik, CART akan menghasilkan pohon regresi (Shafila, 2020). XGBoost juga menerapkan metode boost dengan beberapa set pohon keputusan yang pembuatan pohon berikutnya akan bergantung pada *resedual* pohon sebelumnya. Pohon pertama di XGB akan memiliki klasifikasi yang lemah karena itu merupakan hasil dari inisialisasi probabilitas yang ditentukan oleh peneliti dan kemudian akan dilakukan pembaruan bobot pada setiap pohon yang dibuat untuk menghasilkan kumpulan pohon klasifikasi yang kuat (Rombe, 2021).

**2.7.1 Fungsi Objektif**

Fungsi objektif adalah metrik untuk menentukan seberapa baik model berdasarkann data latih. Fungsi *training loss* dan dan fungsi *regularization term* adalah dua aspek dari fungsi objektif. Persamaan fungsi tujuan adalah sebagai berikut:

|  |  |
| --- | --- |
|  | ( 1 ) |

|  |  |
| --- | --- |
|  | ( 2 ) |

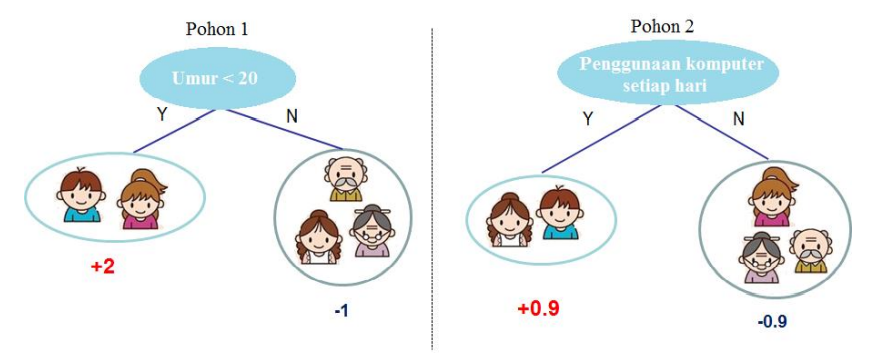
Dimana merupakan fungsi *training loss*, adalah fungsi *regularization term*. Fungsi training adalah fungsi yang berperan untuk mengukur performa model berdasarkan data latih. Sedangkan fungsi *regularization term* adalah fungsi yang berperan untuk mengatur kompleksitas dari model sehingga bisa mengurangi model *overfitting.* Secara umum fungsi *training loss* dapat dituliskan sebagai berikut:

Disini adalah nilai aktual yang dianggap benar, adalah nilai hasil perdiksi dari model dan n adalah jumlah iterasi yang ditentukan oleh pengguna. Formula umum yang sering digunakan untuk pengukuran *training loss* adalah *cross entropy loss.* Persamaan *cross entropy loss* adalah sebagai berikut

|  |  |
| --- | --- |
|  | ( 3 ) |

**2.7.2 Decision Tree Ensemble**

XGBoost adalah sebuah algoritma *ensemble learning* yang memanfaatkan konsep pohon keputusan. Pohon keputusan sendiri adalah metode pembelajaran yang melibatkan grafik seperti pohon untuk memodelkan data yang bersifat kontinu atau kategorik. XGBoost menggabungkan beberapa hasil dari pohon keputusan sehingga mendapatkan hasil yang terbaik.

Berikut adalah ilustrasi bagaimana pohon yang digunakan oleh XGBoost melakukan klasifikasi apakah seseorang suka memainkan permainan pada komputer.

Gambar 2.3 Klasifikasi Anggota Keluarga

Sumber: Chen T, Guestrin C (2016)

Gambar diatas mengklasifikasikan anggota keluarga ke dalam daun yang berbeda, dan memberikan skor pada daun yang sesuai. Skor dari masing-masing pohon akan disimpulkan untuk mendapatkan skor akhir. Setiap pohon yang dibangun akan saling melengkapi dan pembuatan pohon untuk iterasi selanjutnya akan memanfaatkan residual dari pohon sebelumnya. Hasilnya akan dievaluasi menggunakan fungsi aditif sebagai berikut:

|  |  |
| --- | --- |
|  | ( 4 ) |

Yang mana melambangkan label data training, t adalah jumlah tree, adalah himpunan dari semua CART yang memungkinkan. Dikarenakan XGBoost memanfaatkan sekumpulan pohon, maka untuk mengukur kompleksitas dari pohon fungsi *regularization term* akan menghitung jumlah node, *depth*, dan skor *leaf* pada setiap pohonnya sehinga kompleksitas model secara keseluruhan dapan diukur sehingga persamaan untuk fungsi objektif akan menjadi seperti ini:

|  |  |
| --- | --- |
|  | ( 5 ) |

Menurut Chen dan Guestrin, untuk mempelajari semua pohon sekaligus adalah hal yang sangat sulit. Sehingga mereka merenapkan strategi aditif yaitu memperbaiki apa yang telah dipelajari dengan menambah satu pohon baru sekaligus. Oleh karena itu, persamaan fungsi objektif 5 diubah sedemikian rupa sehingga menciptakan satu persamaan baru sebagai berikut

|  |  |
| --- | --- |
|  | ( 6 ) |

nilai 𝑔𝑖 dan ℎ𝑖 didefinisikan dengan persamaan 7 dan persamaan 8:

|  |  |
| --- | --- |
|  | ( 7 ) |
|  | ( 8 ) |

Dengan persamaan 6 fungsi objektif, dapat kita lihat bahwa nilai dari fungsi objektif sangat bergantung pada dan serta kita dapat mengetahui bagaimana cara *Extreme Gradient Boosting* dalam mengoptimalkan fungsi *training loss*.

Seperti yang telah disampaikan sebelumnya bahwa fungsi objektif terdiri dari dua komponen yaitu fungsi *training loss* dan fungsi *regularization term.* Sejauh ini kita sudah membahas tentang fungsi *traning loss* serta berbagai macam transformasinya. Sekarang kita akan membahas tentang tentang fungsi *regularization term.* Fungsi *regularization term* secara umum dapat dirumuskan seperti persamaan berikut:

|  |  |
| --- | --- |
|  | ( 9 ) |

Dimana 𝑤 adalah vector skor dari leaf, q adalah fungsi yang menempatkan setiap input data ke leaf yang sesuai, dan 𝑡 adalah jumlah leaf. Sama seperti fungsi *training loss,* pada XGBoost fungsi *regularization term* juga mengalami tranformasi sedemikian rupa sehingga persamaan fungsi *regularization term* pada XGBoost adalah sebagai berkut:

|  |  |
| --- | --- |
|  | ( 10 ) |

Sehingga secara lengkap fungsi objektif pada XGBoost akan terlihat seperti persamaan berikut:

|  |  |
| --- | --- |
|  | ( 11 ) |

**2.7.3 Feature Importance**

Feature importance atau tingkat kepentingan fitur adalah sebuah teknik untuk menghitung skor dari semua fitur yang diberikan untuk sebuah model. Skor tersebut merepresentasikan tingkat kepentingan dari setiap fitur. Salah satu keuntungan menggunakan XGBoost adalah setelah *boosted tree* dibangun, skor untuk setiap atribut mudah ditentukan. Hal ini karena skor yang diberikan dapat menunjukan seberapa besarnya sebuah fitur dalam pembuatan *boosted tree.* Untuk mendapatkan *feature importance* dari setiap fitur pada XGBoost, kita dapat menggunakan fungsi yang sudah disediakan oleh XGBoost yaitu xgb.importance.

Salah satu parameter dari fungsi xgb.importance adalah *importance type. Importance type* adalah parameter yang menentukan bagaimana cara untuk menghitung skor pada *feature importance*. Salah satu metode dalam menghitung skor *feature importance* adalah dengan menggunakan *gain*. *Gain* adalah rata-rata *gain of splits* dari fitur yang digunakan didalam pohon. Gain dapat didefinisikan sebagai persamaan berikut:

|  |  |
| --- | --- |
|  | ( 12 ) |

1. **Preprocessing**

Preprocessing adalah tahapan pengolahan data menjadi format tertentu dengan tujuan agar data dapat terstandarisasi dan memiliki struktur yang konsisten. Preprocessing adalah tahapan pengolahan data menjadi format tertentu dengan tujuan agar data dapat terstandarisasi dan memiliki struktur yang konsisten. Teknik ini perlu diterapkan karena pada dasarnya data yang kita miliki tidak terstruktur, banyak duplikasi, dan inkonsisten(Ruslam, 2021).

Diagram

Description automatically generated

Gambar 2.4 Proses Preprocessing

Dalam data science proses preprocessing meliputi beberapa tahapan. Gambar menunjukkan tahapan-tahapan dalam preprocessing.

1. Data cleaning merupakan tahapan pembersihan dengan mengidentifikasi data yang rusak atau tidak sesuai dengan data yang berada dari sumber tabel. Pada tahapan ini fokus dari pengelolaan data adalah mengidentifikasi data yang tidak konsisten, data yang rusak, dan tidak akurat lalu melakukan modifkasi pada data tersebut atau menghapusnya.
2. Data Integration adalah tahapan penggabungan data dari berbagai sumber menjadi satu kesatuan. Dalam prosesnya, penggabungan ini akan memunculkan error karena data yang tersedia pasti memiliki struktur yang berbeda-beda. Dengan demikian, ketelitian dalam mencari kesamaan dari setiap data yang tersedia lalu menggabungkannya merupakan kemampuan yang sangat diperlukan dalam proses ini.
3. Data Transformation merupakan proses pengubahan data dari format yang sulit dipahami menjadi format yang mudah untuk dimengerti. Pada tahap ini dilakukan normalisasi, agregasi, dan generalisasi terhadap data yang dimiliki. Normalisasi meliputi pengolaan data yang redundan sehingga dapat memperkecil ukuran dari data yang dimiliki. Agregasi berfungsi untuk menggabungkan data berdasarkan variabel tertentu sehingga penyerapan informasi dapat dilakukan dengan lebih mudah. Generalisasi adalah proses untuk melakukan rangkuman terhadap data yang dimiliki untuk mengevaluasi kualitas dari data yang diolah.
4. Data Reduction merupakan tahapan pengurangan dimensi yang bertujuan untuk mempermudah komputasi dari pengelolaan data sehingga proses pengelolaan data yang besar menjadi lebih cepat.
5. Data Discretization adalah tahapan untuk mengubah data yang bersifat kontinu menjadi data diskret.

Salah satu tools untuk melakukan preprocessing pada python adalah Pandas. Pandas merupakan library python yang memiliki kemampuan pengolahan data dalam bentuk tabel (bariskolom) dan kalkulasi statistik. Pandas dibuat untuk memenuhi kebutuhan akademis dan industry data science dalam preprocessing data menggunakan python. Sebagai tools preprocessing, Pandas memiliki berbagai kemampuan pengolahan data, seperti membentuk struktur dari sebuah data menjadi bentuk baris dan kolom. Salah satu data yang paling sering diolah adalah data dari API yang berbentuk JSON (Ruslam, 2021).

1. **Splitting Data**

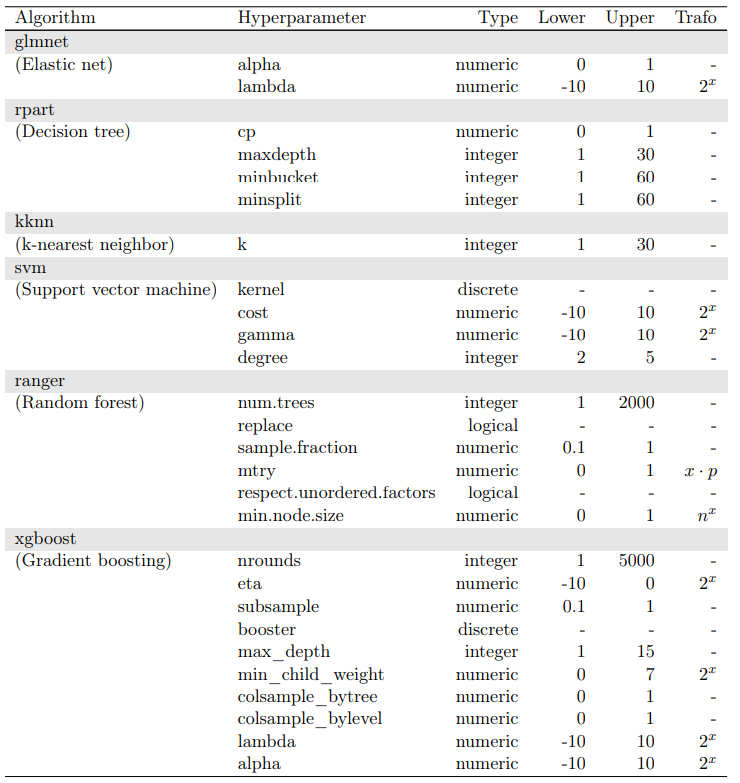
Dalam Machine Learning, salah satu persyaratan utama adalah membangun model komputasi yang menggeneralisasi dengan baik ke pengetahuan yang diekstraksi. Ketika pelatihan misalnya jaringan saraf tiruan, generalisasi yang buruk sering bermanifestasi sebagai overtraining. Cara umum untuk menghindari overtraining adalah dengan menjaga validasi silang. Namun, masalah mendasar dengan pendekatan ini adalah pemisahan data yang tepat. Dalam sebagian besar aplikasi, sampling acak sederhana digunakan. Namun, ada beberapa metode pengambilan sampel statistik lanjutan yang berlaku untuk berbagai jenis kumpulan data, yang dapat dipisahkan menggunakan teknik pengambilan sampel statistik klasik. Metode pengambilan sampel ini dapat dibagi ke dalam kategori kompleks berikut sesuai dengan prinsip, tujuan, algoritma, dan daya komputasinya:

* Simple random sampling (SRS),
* Trial-and-error methods,
* Systematic sampling,
* Convenience sampling,
* CADEX, DUPLEX,
* Stratified sampling (Reitermanova, 2010).

1. **Turning Hyperparameter Random Search**

Algoritma *machine learning* seperti *neural network, gradient boosting, random forest, Extreme Gradient Boosting* dan lain sebagainya dibangun untuk membantu manusia dalam menyelesaikan masalah berkaitan dengan data. Algoritma-algoritma tersebut memiliki berbagai parameter yang dapat membantu kita untuk dapat memaksimalkan algoritma tersebut terhadap data yang kita miliki. Kita dapat menentukan nilai dari parameter-parameter tersebut dengan nilai bawaan yang sudah disiapkan ataupun menentukan nilai tersebut sesuai dengan keinginan kita (Probst and Bischl, 2019).

Ada berbagai cara yang dapat dilakukan untuk mengatur nilai parameter terbaik untuk model salah satunya adalah dengan menggunakan *random search. Random search* adalah penyetelan hyper-parameter dengan pendekatan dengan menggunakan konsep dari ruang sampel atau ruang pencarian. Salah satu keuntungan menggunakan *random search* adalah proses komputasi yang terbilang cukup cepat dan memberikan nilai yang setara dibandingkan dengan metode yang lain. Secara teoritis, random search memanfaatkan probabitas asumsi bahwa setidaknya 5% dari semua titik di ruang hyperparameter adalah solusi optimal (atau dekat), dengan mengambil sampel 60 titik, ada peluang 95% setidaknya satu dari mereka akan berada di atas -melakukan hyper-parameter. Jadi, peneliti menggunakan sampel 60 hyper-parameter dalam penelitian ini. Mirip dengan pencarian grid, peneliti menggunakan pendekatan validasi silang 10 kali lipat (Villalobos-Arias *et al.*, 2020)

Berikut adalah daftar hyperparameter dari beberapa algoritma yang bisa dilakukan *tuning*:

Gambar 2.5 Macam-macam Algoritma Hyperparameter

Sumber: Probst P, Bischl B (2019)

1. **Confusion Matrix**

*Confusion matrix* adalah sebuah metode yang dapat kita gunakan untuk mengevaluasi model yang kita buat. *Confusion matrix* berisi tentang perbandingan hasil klasifikasi dari model yang kita buat dengan klasifikasi yang sebenarnya. Empat komponen penting dalam *confusion matrix* yaitu *true positive* (TP) atau data positif yang terdeteksi benar, *true Negative* (TN) atau data negative yang terdeteksi benar, *false positive* (FP) atau data positif yang terdeteksi salah, dan *false negative* (FN) atau data negatif yang terdeteksi salah.

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Nilai sebenarnya | |
| True | False |
| Nilai Prediksi | True | TP (True Positive) | FP (False Positive) |
| False | FN (False Negative) | TN (True Negative) |

Nilai-nilai dalam confusion matrix juga berkaitan dengan tingkat *accuracy, precision, recall / sensitivity dan spesicifity*. Berikut adalah persamaan untuk menghitung keempat nilai tersebut:

|  |  |
| --- | --- |
|  | ( 13 ) |
|  | ( 14 ) |
|  | ( 15 ) |
|  | ( 16 ) |

1. ***Review* Penelitian Terdahulu**

Hasil penelitian terdahulu yang dicantumkan berikut sebagai referensi terhadap penelitian yang penulis akan dilakukan, diantaranya:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Jurnal | Penulis | Metode yang digunakan | Luaran |
| 1 | Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan | Ghaitsa Amany Mursianto, Isma’il Muhammad Falih, Muhammad Irfan ,Tiara Sakinah | *Extreme Gradient Boosting* dan Random Forest pada dataset Weather AUS yang berasal dari situs Kaggle.com | Sebuah model machine learning dengan tingkat akurasi sebesar 89.54% untuk RF tanpa smote, 88,96% untuk XGBoost, 95,59% untuk SMOTE RF, dan 94,34% untuk SMOTE XGBoost |
| 2 | Klasifikasi Cuaca Provinsi Dki Jakarta Menggunakan  Algoritma Random Forest Dengan Teknik Oversampling | Faqih Hamami , Iqbal Ahmad Dahlan2 | Random Forest pada dataset cuaca provinsi DKI Jakarta Tahun 2018 dari situs katalog.data.go.id | Sebuah model machine learning random forest dengan tingkat akurasi sebesar 70% |
| 3 | PENERAPAN METODE K-NEAREST NEIGHBOR(KNN) UNTUK KLASIFIKASI CUACA DI WILAYAH TANJUNGPINANG | Syarifaturrahmah, Alena Uperiati, Eka Suswaini | K-NEAREST NEIGHBOR(KNN) pada dataset Stasiun Meteorologi Raja Haji Fisabilillah Tanjungpinang mulai dari Januari tahun 2016-Desember tahun 2019 | Sebuah model machine learning KNN dengan tingkat akurasi sebesar 70% |

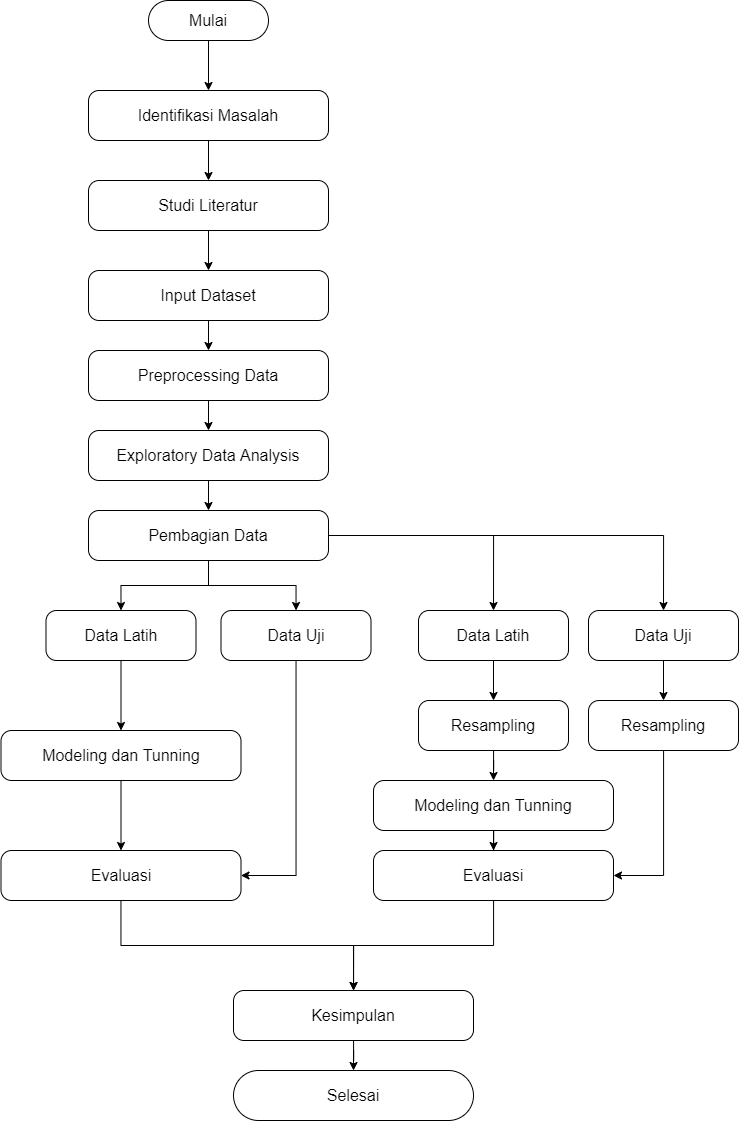
Tabel diatas merupakan daftar menelitian terdahulu yang dapat dijadikan sebagai referensi atau literatur untuk penelitian ini. Ketiga penelitian diatas memiliki persamaan dan perbedaan masing-masing. Persamaannya adalah ketiga penelitian tersebut membuat sebuah model *machine learning* untuk memprediksi cuaca. Perbedaanya ada pada data dan algoritma yang digunakan. Penelitian pertama menggunakan data weatherAUS yang diperoleh dari situs Kaggle dengan algoritma *Extreme Gradient Boosting*, random forest , dan metode resampling dengan SMOTE, penelitian kedua mnggunkan data cuaca Pronvinsi DKI Jakarta tahun 2018 dengan algoritma *random forest* dan resampling. Sementara itu, penelitian ketiga sedikit berbeda dengan dua peneleitian sebelumnya. Penelitian ketiga menggunakan algoritma KNN pada data Stasiun Meteorologi Raja Haji Fisabilillah Tanjungpinang mulai dari Januari tahun 2016-Desember tahun 2019 tanpa adanya resampling.

Dari ketiga penelitian tersebut, belum ada yang membuat sebuah model *machine learning* dengan menggunakan algoritma *Extreme Gradient Boosting* dengan data cuaca Provinsi DKI Jakarta tahun 2018. Untuk itu, peneliti tertarik untuk membuat sebuah percobaan untuk menemukan performa algoritma *Extreme Gradient Boosting* yang sedang ramai dibicarakan di komunitas karena performanya yang cukup baik untuk menangani data dengan persebaran data yang acak seperti data cuaca Provinsi DKI Jakarta tahun 2018.

Setelah membaca ketiga penelitian tersebut, peneliti mendapatkan gambaran mengenai tahapan-tahapan yang akan dilakukan untuk membuat pemodelan machine learning dengan data cuaca Provinsi DKI Jakarta tahun 2018, yaitu melakukan input dataset, melakukan praproses terhadap data, menganalisis dan mengeksplorasi data, pembagian data, pelatihan data, dan evaluasi performa model. Sebagai pembeda dari ketiga penelitian tersebut, peneliti ingin membuat columtransformer, pipeline dan melakukan tuning hyperparameter dari algoritma yang dipakai. Selain itu peneliti juga ingin memperkecil jumlah dari kelas label sehingga dapat memberikan hasil yang lebih umum.

**BAB III**

**METODOLOGI PENELITIAN**

1. **Alur Penelitian**

Gambar 3 alur penelitian

1. **Tahapan Penelitian**
2. **Identifikasi Masalah**

Pada tahapan indentifikasi masalah, peneliti menemukan bahwa prakiraan cuaca adalah hal yang cukup penting terlebih lagi untuk kehidupan di ibu kota. Maka dari itu, peneliti ingin membuat sebuah sistem yang dapat membuat prakiraan cuaca sambil menerapkan ilmu yang telah dipelajari sebelumnya yaitu *machine learning.* Peneliti menggunakan algoritma *Extreme Gradient Boosting* karena algoritma tersebut terkenal baik untuk menangani data yang memiliki tipe data yang acak seperti data cuaca Provinsi DKI Jakarta tahun 2018. Oleh karena itu, selain untuk membuat sistem yang dapat membuat prakiraan cuaca peneliti juga ingin mencobakan algoritma *Extreme Gradient Boosting* langsung pada data yang aktual.

1. **Studi Literatur**

Pada tahapan ini peneliti melakukan studi literatur untuk mendapatkan teori-teori yang valid terkait dengan penelitian yang dilakukan sebelumnya. Sumber literatur yang digunakan untuk dijadikan referensi antara lain jurnal, artikel, dan situs internet.

1. **Input Dataset**

Data yang digunakan untuk penelitian ini diambil dari open dataset Indonesia pada website https://katalog.data.go.id/dataset/data-prakiraan-cuaca-wilayah-provinsi-dki-jakarta-tahun-2018. Data ini memiliki jumlah 6 kolom dan 8535 baris yang berisinya catatan kelembaban udara serta suhu dari setiap kota administrasi di DKI Jakarta yang dicatat setiap 4 kali dalam sehari selama satu tahun..

1. ***Preprocessing* Data**

Tahapan selanjutnya, peneliti akan melakukan pra-proses pada data yang telah diperoleh. Tahapan pra-proses yang lakukan disini bertujuan untuk membersihkan data dari derau (*noise)*, memperbaiki tipe data, memperbaiki nilai dari setiap kolom, serta menghapus kolom yang tidak digunakan untuk pemrosesan. Berikut adalah beberapa hal yang peneliti lakukan saat melakukan tahapan *preprocessing*, yaitu:

1. Memproses Missing Value

Salah satu noise yang terdapat pada data ini adalah missing value. Melihat apakah terdapat data yang kosong dan menangani data kosong tersebut dengan cara menghapus data tersebut menggunakan fungsi dropna(). Setelah dilakukan penghapusan data pada missing value, dapat di cek kembali untuk melihat apakah masih ada missing value atau tidak.

1. Memproses Kolom Cuaca dan Waktu

Tahapan ini bertujuan untuk membersihkan dan mengelompokkan data pada kolom cuaca dan waktu agar data lebih mudah untuk diproses pada tahap selanjutnya. Pada kolom cuaca akan mengelompokkan data menjadi 3 yaitu Cerah, Hujan dan Berawan sedangkan pada kolom waktu mengelompokkan data menjadi 4 yaitu Pagi, Siang, Malam dan Dini Hari. Proses ini dilakukan menggunakan fungsi Regular Expression.

1. Memproses Kolom kelembaban\_persen Dan suhu\_derajat\_celcius

Tahapan ini bertujuan untuk membagi data pada kolom kelembaban\_persen dan suhu\_derajat\_celcius menjadi 2 bagian yaitu data minimal dan maksimal. Proses ini menggunakan fungsi split() untuk membagi data lalu disimpan kedalam variabel kelembaban\_min dan suhu\_min untuk data minimal serta kelembaban\_max dan suhu\_max untuk data bernilai maksimal berdasarkan kolom kelembaban\_persen dan suhu\_derajat\_celcius.

1. Memperbaiki Tipe Data

Selanjutnya, kolom dataset yang masih memiliki tipe data objek tetapi memiliki nilai numerik atau angka akan diubah menjadi bertipe data integer, untuk memudahkan dalam melakukan proses modelling. Fitur-fitur yang diubah menjadi integer yaitu, kelembaban\_min, suhu\_min, kelembaban\_max, dan suhu\_max.

1. Menghapus Kolom

Tahapan terakhir pada pra-proses ini adalah memilih fitur yang akan digunakan dengan cara menghapus kolom yang tidak mempengaruhi proses klasifikasi menggunakan fungsi .drop(). Beberapa kolom yang dihapus dan tidak digunakan yaitu tanggal, kelembaban\_persen, dan suhu\_derajat\_celcius.

1. ***Exploratory Data Analysis***

Setelah dilakukan pra-proses akan dilanjutkan dengan melihat serta menganalisis data yang telah diperoleh untuk membuat model yang maksimal. Ada beberapa hal yang dilakukan pada tahapan ini yaitu:

1. Melihat Jumlah Total Baris dan Kolom

Melihat jumlah baris dan kolom dari dataset menggunakan fungsi .shape yang menghasilkan jumlah 6 kolom dan 8535 baris dengan tujuan untuk melihat seberapa besar data yang dipunya.

1. Melihat Tipe Data

Selanjutnya dilakukan kembali pengecekan terhadap tipe data pada setiap kolom yang belum sesuai menggunakan fungsi .dtypes() agar data lebih mudah diproses pada tahap berikutnya.

1. Melihat Statistik Deskriptif

Tahapan berikutnya adalah dengan melihat statistik deskriptif pada setiap kolom untuk melihat ringkasan dari data secara keseluruhan menggunakan fungsi .describe(). Fungsi ini dapat memberikan informasi mengenai nilai rata-rata, standar deviasi dan interquartile pada setiap kolom.

1. Melihat Bentuk Persebaran Data

Tahap selanjutnya adalah melihat grafis visualisasi data yang bertujuan untuk menganalisis data agar lebih mudah dipahami. Proses visualisasi data yang peneliti lakukan menggunakan package Seaborn lalu memanggil fungsi .pairplot() yang nantinya akan menghasilkan grafis visualisasi dari data cuaca Provinsi DKI Jakarta tahun 2018.

1. **Pembagian Data**

Tahapan setelah pra-proses dan exploratory data analysis adalah pembagian data menjadi dua yaitu data latih dan data uji dengan perbandingan 80% data latih dan 20% data uji. Selanjutnya akan dipisahkan antara kolom label dan kolom fitur. Pada kolom label dilakukan pembagian secara stratifikasi atau pembagian data secara seimbang karena label yang digunakan bertipe kategorikal.

1. **Resampling Data Menggunakan Teknik Smote**

Setelah tahapan pembagian data, peneliti melakukan pengecekan terhadap jumlah data pada kolom label dan menemukan bahwa label pada dataset yang peneliti gunakan tidak seimbang dengan perbandingan imbalance ratio sebesar 45% sehingga diperlukan proses untuk menyeimbangkan jumlah label terlebih dahulu menggunakan metode resampling dengan teknik oversampling yaitu SMOTE. Resampling adalah teknik untuk memanipulasi kelas distribusi dengan memperbaiki data latih sehingga didapatkan kelas yang seimbang (Sir and Soepranoto, 2022). Oversampling adalah salah satu teknik resampling yang bertujuan untuk memanipulasi data dengan membuat data baru pada data minoritas sehingga jumlahnya akan seimbang dengan jumlah data mayoritas (Mohammed, Rawashdeh and Abdullah, 2020). Metode yang populer diterapkan untuk menangani permasalahan keseimbangan data adalah SMOTE. SMOTE atau Synthetic Minority Over-sampling Technique adalah metode pembangkitan data minoritas sebanyak data mayoritas dengan SMOTE yang bekerja dengan cara mengambil secara acak tetangga terdekat sebanyak k dari setiap instance dalam kelas minoritas kemudian membuat instance baru (sintetis) antara instance tersebut dengan tetangga terdekat k yang dipilih secara acak. Dengan pendekatan SMOTE maka dapat dipastikan tidak terjadi masalah duplikasi data sehingga lebih kebal terhadap masalah overfitting (Sir and Soepranoto, 2022).

1. ***Preprocessing, Modeling,* dan *Tuning***
2. ***Preprocessing***

Sebelum tahapan selanjutnya, data perlu untuk disiapkan terlebih dahulu. peneliti melakukan tahapan pra-proses kembali setelah tahap pembagian data. Hal ini dilakukan untuk menghindari kebocoran data sehingga akan mengurangi tingkat overfitting ataupun underfitting. Pada tahapan pra-proses kali ini, peneliti menggunakan ColumnTransformer dengan membagi kolom menjadi kolom numerik dan kolom kategorik. Untuk kolom numerik dilakukan pra-proses berupa PolynomialFeature untuk sedangkan kolom kategorik dilakukan pra-proses berupa OneHotEncoder dengan tujuan untuk membangun model yang sederhana yang dapat digunakan untuk pemahaman dan melakukan prediksi.

1. ***Modeling***

Setelah membuat ColumnTransformer, kemudian data dilatih dengan algoritma yang ingin diimplementasikan yaitu *Extreme Gradient Boosting*. Peneliti membuat Pipeline untuk menggabungkan *ColumnTransformer* yang telah dibuat, algoritma yang akan peneliti gunakan untuk proses *tuning* nantinya, serta *resampling* SMOTE.

1. ***Tuning***

Pembuatan *Column Transformer* dan Pipeline bertujuan untuk mempermudah proses dan memprediksi data baru dengan model telah dibuat serta memungkinkan kita dalam melakukan *tuning hyperparameter* berdasarkan algoritma yang ada di dalam pipeline. *Tuning hyperparameter* sendiri merupakan proses untuk menemukan nilai *hyperparameter* terbaik yang sesuai dengan data yang dimiliki sehingga dapat menghasilkan model dengan performa yang terbaik (Radhi *et al.*, 2021).

1. **Evaluasi**

Setelah semua tahapan selesai, selanjutnya akan dilakukan tahapan evaluasi performa dari *Extreme Gradient Boosting* menggunakan data uji. Peneliti menggunakan *confusion matrix* untuk mengetahui efisiensi kinerja model agar dapat diketahui seberapa banyak data yang salah diklasiﬁkasi pada data uji dan *classification report* untuk mengukur tingkat akurasi, presisi, dan *recall.*

1. **Waktu dan Tempat Penelitian**

Adapun waktu dan tempat terlaksananya penelitian ini :

|  |  |
| --- | --- |
| Waktu | : Aprill - Juni 2022 |
| Tempat | : Universitas Pembangunan Nasional Veteran Jakarta Jl. RS. Fatmawati Raya, Pd. Labu, Kec. Cilandak, Kota Depok, Jawa Barat 12450 |

1. **Alat Bantu Penelitian**
2. **Hardware**

Pada penelitian ini, peneliti menggunakan komputer desktop dengan spesifikasi sebagai berikut:

1. Processor : AMD Ryzen 5 3600
2. RAM : 16384 MB
3. VGA : AMD Radeon RX 570
4. HDD : 1TB
5. SSD : 128GB
6. **Software**

Penelitian ini menggunakan beberapa software (perangkat lunak) sebagai berikut:

1. Windows 10 Pro
2. Visual Studio Code
3. Microsoft Excel
4. Microsoft Edge
5. **Jadwal Penelitian**

**DAFTAR PUSTAKA**

Aldrian, E., Karmini, M. and Budiman (2011) *Adaptasi dan Mitigasi Perubahan Iklim di Indonesia*. Available at: http://www.bmkg.go.id.

Manorathna, R. (2020) *Polynomial Regression with a Machine Learning Pipeline*. Available at: https://www.researchgate.net/publication/344616388.

Mohammed, R., Rawashdeh, J. and Abdullah, M. (2020) “Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results,” in *2020 11th International Conference on Information and Communication Systems, ICICS 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 243–248. doi:10.1109/ICICS49469.2020.239556.

Mursianto, G.A. (2021a) “Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan.”

Mursianto, G.A. (2021b) “Perbandingan Metode Klasifikasi Random Forest dan XGBoost Serta Implementasi Teknik SMOTE pada Kasus Prediksi Hujan.”

NUR ROCHMAN DARMAWAN (2019) *PREDIKSI KONDISI CUACA KOTA SURABAYA MENGGUNAKAN METODE ARTIFICIAL NEURAL NETWORK PREDICTION OF SURABAYA CITY WEATHER CONDITIONS USING ARTIFICIAL NEURAL NETWORK METHOD*.

Probst, P. and Bischl, B. (2019) “Tunability: Importance of Hyperparameters of Machine Learning Algorithms,” *Journal of Machine Learning Research*, 20, pp. 1–32. Available at: http://jmlr.org/papers/v20/18-444.html. (Accessed: May 24, 2022).

Putra, J.W.G. (2020a) *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning Edisi 1.4 (17 Agustus 2020)*. Tokyo.

Putra, J.W.G. (2020b) *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning Edisi 1.4 (17 Agustus 2020)*. Tokyo.

Putra, J.W.G. (2020c) *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning Edisi 1.4 (17 Agustus 2020)*. Tokyo.

Radhi, M. *et al.* (2021) “PREDIKSI WATER QUALITY INDEX (WQI) MENGGUNAKAN ALGORITMA REGRESSI DENGAN HYPER-PARAMETER TUNING,” *Jurnal Sistem Informasi dan Ilmu Komputer Prima)*, 5(1).

Reitermanova, Z. (2010) “Data Splitting,” *WDS’10 Proceedings of Contributed Papers*, Part I, pp. 31–36.

Rombe, Y. (2021) “PENGGUNAAN METODE XGBOOST UNTUK KLASIFIKASI STATUS OBESITAS DI INDONESIA.”

Ruslam, A.W.A. (2021) *ANALISIS KEBERLANJUTAN PENGGUNA JALA MENGGUNAKAN FACTOR ANALYSIS*. Jakarta.

Shafila, G.A. (2020) *IMPLEMENTASI METODE EXTREME GRADIENT BOOSTING (XGBOOST) UNTUK KLASIFIKASI PADA DATA BIOINFORMATIKA*.

Sir, Y.A. and Soepranoto, A.H.H. (2022) “Pendekatan Resampling Data Untuk Menangani Masalah Ketidakseimbangan Kelas,” *Jurnal Komputer dan Informatika*, 10(1), pp. 31–38. doi:10.35508/jicon.v10i1.6554.

Villalobos-Arias, L. *et al.* (2020) “Evaluating Hyper-parameter Tuning using Random Search in Support Vector Machines for Software Effort Estimation.” doi:10.1145/3416508.3417121.

Ying, X. (2014) *Ensemble Learning*. Georgia. Available at: https://www.researchgate.net/publication/262369664.