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 seperti cuaca pada pagi hari, cuaca pada siang hari, cuaca pada sore hari, ataupun cuaca pada malam hari. Keadaan cuaca juga bisa berbeda-beda pada setiap tempat dan jamnya [1]. Namun, Seiring meningkatnya suhu global, maraknya pembalakan hutan, pembakaran karbon dan lain sebagainya, cuaca di seluruh dunia dan juga di Jakarta berubah dengan cepat dan terus menerus. Prakiraan yang tepat sangat penting untuk kehidupan di Ibu Kota. Dari kegiatan bisnis, usaha kecil menengah sampai pekerjaan sehari-hari dan lain sebagainya 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 gunakan untuk mendapatkan wawasan dari berbagai jenis data dari data berlabel hingga tidak berlabel, data terstruktur hingga data tidak terstruktur, data regresi 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. Setelah itu kita dapat 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 [2].

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 [3]. XGB dibangun dengan 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.

1. **Rumusan Masalah**

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

1. Bagaimana cara mendapatkan parameter terbaik yang dihasilkan pada model Extreme Gradient Boosting?
2. Seberapa akurat algoritma *Extreme Gradient Boosting* untuk mengklasifikasikan cuaca data cuaca Provinsi DKI Jakarta tahun 2018?
3. **Tujuan Penelitian**

Adapun tujuan penelitian ini berdasarkan rumusan masalah sebagai berikut:

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

Adapun manfaat penelitian yang ingin dicapai antara lain :

1. Dengan didapatkannya hasil dari data training, maka dapat diketahui seberapa baik hasil pembelajaran yang diberikan dari data, sehingga dapat digunakan untuk menguji pada data testing untuk melihat evaluasi.
2. Dengan didapatkannya hasil dari data testing yang baik, maka dapat digunakan untuk melakukan prediksi kedepannya.
3. Penelitian ini dapat memberikan informasi mengenai algoritma untuk menentukan model terbaik yang akan digunakan pada dataset cuaca Provinsi DKI Jakarta tahun 2018.
4. **Ruang Lingkup**

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

1. Penelitian ini ditujukan untuk mencari algoritma yang lebih baik antara algoritma Extreme Gradient Boosting.
2. Penelitian ini menggunakan metode klasifikasi pada supervised learning yang bertujuan untuk mengelompokkan cuaca pada Provinsi DKI Jakarta tahun 2018.
3. **Luaran Yang Diharapkan**

Luaran yang diharapkan dari penelitian ini adalah mendapatkan nilai akurasi yang tinggi dari model Random Forest dan Extreme Gradient Boosting serta dapat menentukan model yang terbaik antara kedua algoritma tersebut.

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. **Extreme Gradient Boosting (XGBoost)**

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 [2]. 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 [4].

* 1. **Fungsi Objektif (Objective Function)**

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 ) |

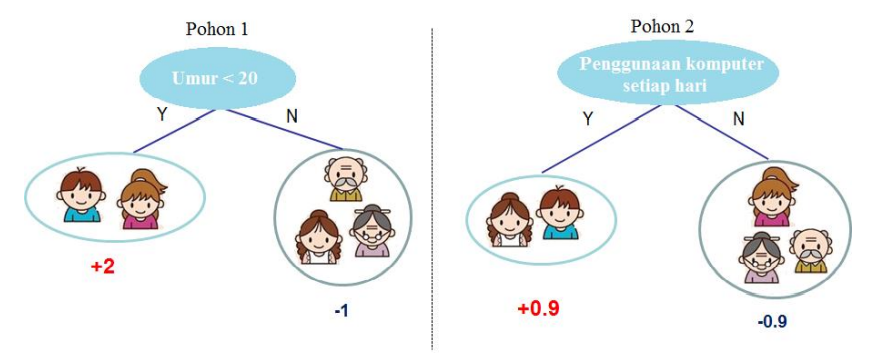
Dimana merupakan fungsi *training loss*, adalah fungsi *regularization term*. Fungsi training adalah fungsi yang berperan untuk mengukur performa model berdasarkan data latih. Sedangkan fungsi *regularization term* adalah fungsi yang berperan untuk mengatur kompleksitas dari model sehingga bisa mengurangi 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 ) |

* 1. **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 1

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 ) |

* 1. **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.

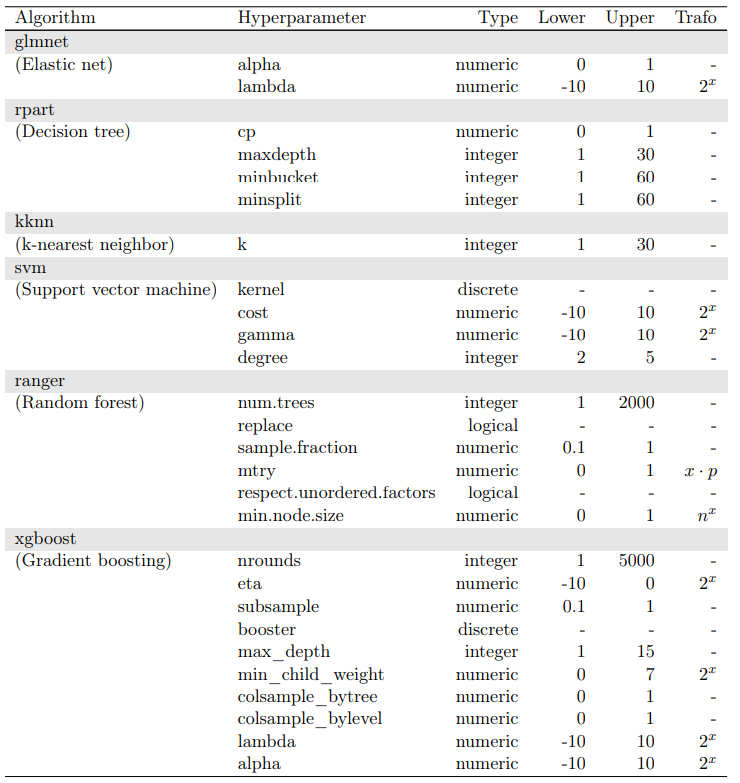
Salah satu parameter dari fungsi xgb.importance adalah *importance type. Importance type* adalah parameter yang menentukan bagaimana cara untuk menghitung skor pada *feature importance*. Salah satu metode dalam menghitung skor *feature importance* adalah dengan menggunakan *gain*. *Gain* adalah rata-rata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

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

1. ***Tunning Hyperparameter***

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 [5].

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, kami menggunakan sampel 60 hyper-parameter dalam penelitian ini. Mirip dengan pencarian grid, kami menggunakan pendekatan validasi silang 10 kali lipat [6]

Berikut adalah daftar hyperparameter dari beberapa algoritma yang bisa kita atur:

Gambar 2

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.

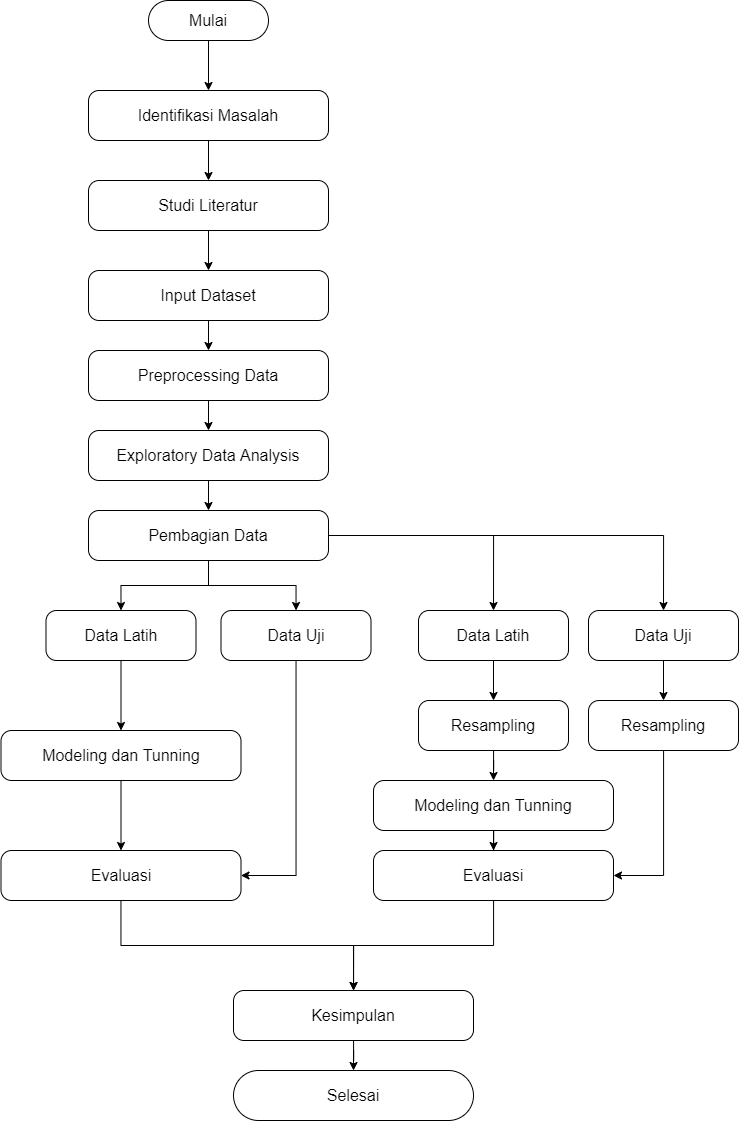
|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Nilai sebenarnya | |
| True | False |
| Nilai Prediksi | True | TP (True Positive) | FP (False Positive) |
| False | FN (False Negative) | TN (True Negative) |

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 ) |

**BAB III**

**METODOLOGI PENELITIAN**

1. **Alur Penelitian**

Gambar 3 alur penelitian

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

Pada tahapan identifikasi masalah, kami membandingkan hasil akurasi algoritma random forest dan extreme gradient boosting berdasarkan cuaca pada Provinsi DKI Jakarta tahun 2018. Kami menggunakan random forest dan extreme gradient boosting karena algoritma tersebut sesuai digunakan pada data yang besar dan juga memiliki persebaran data yang acak.Kami menemukan bahwa algoritma random forest dan extreme gradient boosting sedang ramai dibicarakan pada komunitas karena performanya yang baik serta menghasilkan akurasi yang tinggi di sebagian besar dataset. Oleh karena itu, kami ingin membuktikan kebenaranya dengan menggunakan sebuah data aktual yaitu data cuaca provinsi DKI Jakarta pada tahun 2018.

1. **Studi Literatur**

Pada tahapan ini kami 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 tersebut diambil pada setiap kota administrasi yang ada di DKI Jakarta setiap 4 kali dalam sehari selama satu tahun berdasarkan kelembaban udara serta suhu kota tersebut pada waktu pengambilan data. Data ini memiliki jumlah 6 kolom dan 8535 baris.

1. **Preprocessing Data**

Tahapan selanjutnya, kami akan melakukan pra-proses pada data yang telah kami peroleh. Tahapan pra-proses yang kami lakukan disini bertujuan untuk membersihkan data dari derau (noise), melakukan polynomial feature pada data, serta melakukan integrasi pada data. Berikut adalah beberapa hal yang kami 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() pada pandas. 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 metode 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 variable 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 data bernilai integer atau angka akan diubah menjadi tipe data integer, untuk memudahkan dalam melakukan modelling. Fitur-fitur yang diubah menjadi integer yaitu, kelembaban\_min, suhu\_min, kelembaban\_max, dan suhu\_max.

1. Menghapus Kolom

Tahapan terakhir dilakukan yaitu memilih fitur yang akan digunakan dengan cara menghapus fitur yang tidak mempengaruhi dalam proses klasifikasi dengan menggunakan fungsi .drop(). Kolom fitur yang dihapus dan tidak digunakan yaitu tanggal, kelembaban\_persen, dan suhu\_derajat\_celcius.

1. **Exploratory Data Analysis**

Setelah dilakukan pra-proses, kemudian dilanjutkan dengan melihat dan menganalisis data yang telah diperoleh dengan tujuan untuk membuat model yang maksimal. Ada beberapa hal yang kami lakukan pada tahapan ini yaitu:

1. Melihat Jumlah Total Baris dan Kolom

Melihat jumlah baris dan kolom dari semua fitur dan label dataset menggunakan fungsi .shape yang menghasilkan jumlah kolom sebanyak 6 dengan 8535 baris dengan tujuan untuk melihat seberapa besar data yang dipunya.Melihat korelasi antar setiap data kolom numerik

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 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

Untuk melihat data agar lebih mudah dipahami dan dianalisis dilakukannya proses visualisasi data. Visualisasi data akan menampilkan grafis dari informasi dan data berdasarkan data yang dimiliki. Proses visualisasi data yang kami lakukan dengan menggunakan package Seaborn lalu memanggil fungsi .pairplot() yang nantinya akan menghasilkan grafis visualisasi dari data cuaca Provinsi DKI Jakarta tahun 2018.

1. **Pembagian Data**

Data yang telah dilakukan tahap pra-proses serta exploratory data analysis kemudian data akan dibagi menjadi dua yaitu data latih dan juga data uji dengan perbandingan 80% data latih dan 20% data uji. Kemudian data akan dipisahkan antara kolom label dengan kolom fitur dimana pada kolom label dilakukan pembagian secara stratifikasi atau pembagian data dengan seimbang karena label yang digunakan bertipe kategorikal

1. ***Modeling***

Setelah membuat *ColumnTransformer*, kami akan melatih data dengan algoritma yang ingin kami bandingkan yaitu *Random Forest* dan *Extreme Gradient Boosting.* Kemudian kami membuat *Pipeline* untuk menggabungkan *ColumnTransformer* yang telah dibuat sebelumnya dengan jenis algoritma yang akan kami gunakan untuk proses *tuning* nantinya.

1. ***Random Forest***

*Random forest* merupakan algoritma yang baik digunakan untuk melakukan klasifikasi data dalam jumlah yang besar. *Random forest* akan melakukan suatu penggabungan data dengan *decision tree* serta untuk melakukan proses seleksi sehingga pada saat penentuan klasifikasi pada *decision tree*, algoritma ini akan melakukan pemecahan terhadap jenis fitur yang ada dalam dataset. *Decision tree* akan melakukan prediksi secara acak dengan tujuan untuk menghasilkan akurasi yang maksimal.

1. ***Extreme Gradient Boosting***

Extreme Gradient Boosting adalah metode boosting dengan cara menggabungkan kumpulan pohon keputusan yang akan digunakan untuk pembangunan pohon selanjutnya. Model ini dibangun dengan cara membuat model yang baru berdasarkan model yang lama yang bertujuan untuk menghasilkan kesalahan prediksi (error) yang lebih kecil dari model sebelumnya.

1. ***Tuning***

Tujuan pembuatan *ColumnTransformer* dan *Pipeline* adalah untuk mempermudah dalam memproses serta memprediksi data baru dengan model telah dibuat dan untuk melakukan *tuning hyperparameter* dari setiap algoritma. *Tuning hyperparameter* merupakan proses untuk menemukan nilai hyperparameter yang sesuai dengan data yang dimiliki sehingga dapat menghasilkan model dengan performa yang terbaik.

1. **Evaluasi**

Setelah semua tahapan selesai, selanjutnya akan dilakukan tahapan evaluasi performa dari model Random Forest dan Extreme Gradient Boosting menggunakan data uji. Kami 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. **Kesimpulan**

Pada tahap terakhir ini, peneliti akan mengemukakan kesimpulan yang dapat diambil dari penelitian kami. Serta peneliti memberikan saran berupa rekomendasi untuk penelitian selanjutnya.

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. **Jadwal Penelitian**

**DAFTAR PUSTAKA**

[1] Nur Rochman Darmawan, “Prediksi Kondisi Cuaca Kota Surabaya Menggunakan Metode Artificial Neural Network Prediction Of Surabaya City Weather Conditions Using Artificial Neural Network Method,” 2019.

[2] G. A. Shafila, “Implementasi Metode Extreme Gradient Boosting (Xgboost) Untuk Klasifikasi Pada Data Bioinformatika,” 2020.

[3] G. A. Mursianto, “Perbandingan Metode Klasifikasi Random Forest Dan Xgboost Serta Implementasi Teknik Smote Pada Kasus Prediksi Hujan,” 2021.

[4] Y. Rombe, “Penggunaan Metode Xgboost Untuk Klasifikasi Status Obesitas Di Indonesia,” 2021.

[5] P. Probst And B. Bischl, “Tunability: Importance Of Hyperparameters Of Machine Learning Algorithms,” *Journal Of Machine Learning Research*, Vol. 20, Pp. 1–32, 2019, Accessed: May 24, 2022. [Online]. Available: Http://Jmlr.Org/Papers/V20/18-444.Html.

[6] L. Villalobos-Arias, C. Quesada-López, J. Guevara-Coto, A. Martínez, And M. Jenkins, “Evaluating Hyper-Parameter Tuning Using Random Search In Support Vector Machines For Software Effort Estimation,” 2020, Doi: 10.1145/3416508.3417121.