

**DETEKSI HALAMAN WEBSITE PHISHING MENGGUNAKAN ALGORITMA MACHINE LEARNING GRADIENT BOOSTING CLASSIFIE**

**DETEKSI HALAMAN WEBSITE PHISHING MENGGUNAKAN ALGORITMA MACHINE LEARNING GRADIENT BOOSTING CLASSIFIER**

**Penulis:**

**Roni Andarsyah, S.T., M.Kom.**

**Muhammad Rizal Supriadi**

**CANTUMKAN**

**Logo Penerbit Disini**

**DETEKSI HALAMAN WEBSITE PHISHING MENGGUNAKAN ALGORITMA MACHINE LEARNING GRADIENT BOOSTING CLASSIFIER**

©TULISKAN NAMA PENERBIT DISINI

Penulis:

Roni Andarsyah, S.T., M.Kom.

Muhammad Rizal Supriadi

Editor:

Roni Andarsyah, S.T., M.Kom.

Muhammad Rizal Supriadi

Cetakan Pertama: Isi dengan Bulan saat upload buku

Cover: Muhammad Rizal Supriadi

Tata Letak: Muhammad Rizal Supriadi

Hak Cipta 2023, pada Penulis. Diterbitkan pertama kali oleh:

**ISI NAMA PENERBIT**

ISI ALAMAT PENERBIT

Website: [WEBSITE](http://www.rcipress.rcipublisher.org/) PENERBIT

E-mail: [EMAIL](mailto:rumahcemerlangindonesia@gmail.com) PENERBIT

Copyright © 2023 by NAMA PENERBIT

All Right Reserved

- Cet. I –: NAMA PENERBIT, TAHUN TERBIT

Dimensi : 14,8 x 21 cm

ISBN: KOSONGKAN DULU

Hak cipta dilindungi undang-undang

Dilarang memperbanyak buku ini dalam bentuk dan dengan cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang

**Hak Cipta Pasal 72**

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah).

# KATA PENGANTAR

Selamat datang di buku "Deteksi Halaman Website Phishing Menggunakan Algoritma Pengklasifikasi Machine Learning Gradient Boosting Classifier". Buku ini ditujukan bagi pemula yang ingin mempelajari cara membangun suatu sistem deteksi website phishing menggunakan teknik machine learning.

Buku ini akan memandu Anda melalui proses membangun sistem deteksi website phishing menggunakan algoritma pengklasifikasi machine learning Gradient Boosting Classifier. Anda akan belajar bagaimana menggunakan python untuk membuat model machine learning dan bagaimana menggunakan framework flask untuk membuat tampilan website yang interaktif dan user friendly. Setelah menyelesaikan tutorial ini, Anda akan memiliki keterampilan yang diperlukan untuk membangun sistem deteksi website phishing sendiri yang dapat membantu melindungi pengguna internet dari ancaman phising.

Saya berharap buku ini dapat menjadi sumber informasi yang bermanfaat bagi Anda dan semoga Anda dapat menyelesaikan tutorial ini dengan sukses. Selamat belajar!

Bandung, 18 Oktober 2021

Penulis

Muhammad Rizal Supriadi

**| i**

# DAFTAR ISI

BUATLAH DAFTAR ISI DISINI

**ii |**

# DAFTAR GAMBAR

BUATLAH DAFTAR ISI DISINI

**2 |**

**BAB 1**

**Pendahuluan**

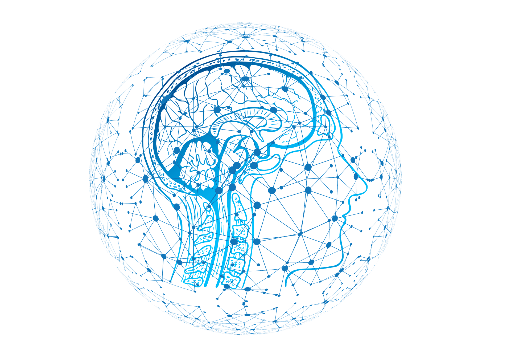
Dalam buku panduan/ tutorial ini, penulis akan memandu Anda melalui proses membangun suatu sistem Deteksi *Website Phising* yang menggunakan teknik *machine learning* dan *model Gradient Boosting Classifier*. Sistem ini akan dibuat menggunakan bahasa pemrograman *python*, yang akan digunakan untuk membuat model machine learning dan juga untuk membuat tampilan website menggunakan framework flask. Tujuan dari sistem ini adalah untuk membantu mengidentifikasi *website phising* yang mungkin merugikan pengguna internet dengan menipu mereka untuk memberikan informasi pribadi atau mengeluarkan dana dengan cara yang tidak sah.

Phishing merupakan tindakan untuk mendapatkan informasi penting seseorang berupa username, password dan informasi sensitif lainnya dengan memberikan informasi palsu situs web yang mirip dengan aslinya. Pengelabuan (pemancing informasi penting) adalah sala satu bentuk tindak pidana yang bermaksud untuk mendapatkan rasahasia informasi dari seseorang, seperti nama pengguna, kata sandi dan kartu kredit, dengan menyamar sebagai orang lain atau bisnis terperaya di elektronik resmi komunikasi, seperti surat elektronik dan instant messages.[1]

Dengan mengikuti tutorial ini, Anda akan belajar bagaimana menggunakan python untuk membuat model *machine learning*, serta bagaimana menggunakan *framework flask* untuk membuat tampilan website yang interaktif dan *user friendly*. Setelah menyelesaikan tutorial ini, Anda akan memiliki keterampilan yang diperlukan untuk membangun sistem Deteksi Website Phising sendiri yang dapat membantu melindungi pengguna internet dari ancaman *phising*.

Sebelum kita memulai pembangunan sistem deteksi website phishing, ada beberapa hal yang perlu dipertimbangkan. Pertama, pastikan bahwa Anda memiliki pengetahuan dasar tentang *machine learning* dan pengklasifikasi algoritma, karena kita akan menggunakan teknik ini untuk membangun sistem deteksi *website phishing*. Kedua, siapkan *tools* yang diperlukan untuk mendukung pembangunan sistem ini, seperti python untuk membuat model *machine learning* dan *framework flask* untuk membuat tampilan *website*. Setelah Anda mempersiapkan hal-hal tersebut, kita dapat mulai membangun sistem deteksi *website phishing* sesuai dengan *tutorial* yang akan kita bahas di buku ini.

* 1. **Pengenalan Machine Learning**



*Gambar 1.1 Machine Learning*

Machine Learning (ML) bukanlah istilah baru dalam bidang ilmu komputer. Namun itu adalah istilah yang sangat berkembang dan berkembang. Machine Learning merupakan mata pelajaran yang mempelajari cara penggunaan mesin untuk meniru kemampuan dan perilaku manusia. Hal ini dianggap sebagai teknologi yang digunakan untuk menjawab pertanyaan dan memperoleh makna dari data. Machine Learning menyatukan statistik dan ilmu komputer untuk mengaktifkan komputer atau mesin untuk melakukan tugas yang diberikan. Tidak seperti manusia, ia memiliki kemampuan untuk belajar dan beradaptasi lebih cepat [2].

Machine Learning adalah bagian dari Kecerdasan Buatan di mana teknik, model, dan Algoritma digunakan untuk belajar dari data dan informasi untuk memperoleh pengetahuan, pola, atau mengirimkan aplikasi AI. Machine Learning terdiri dari kumpulan algoritma yang digunakan sebagai metode untuk melatih kumpulan data historis dan membuat prediksi masa depan [3].

Berikut ini adalah beberapa contoh penerapan machine learning:

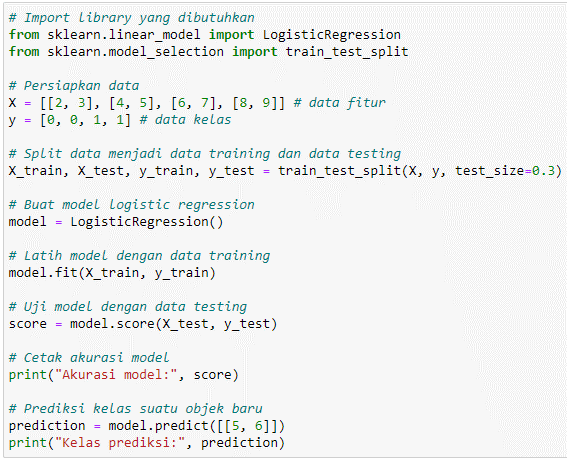
1. Sistem rekomendasi: Sistem rekomendasi menggunakan machine learning untuk mengelompokkan pengguna berdasarkan preferensi dan kebiasaan mereka, dan kemudian menyarankan produk atau layanan yang mungkin mereka sukai.
2. Analisis sentimen: Machine learning dapat digunakan untuk menganalisis sentimen terhadap suatu produk atau layanan dengan mempelajari ulasan atau komentar yang diberikan oleh pengguna.
3. Pendeteksi spam: Machine learning dapat digunakan untuk mengidentifikasi dan memblokir email spam dengan mempelajari pola-pola yang biasanya terdapat pada email spam.
4. Pengenalan wajah: Machine learning dapat digunakan untuk mengenali wajah orang dalam foto atau video dengan mempelajari wajah-wajah yang telah dikenali terlebih dahulu.
5. Penerjemahan otomatis: Machine learning dapat digunakan untuk menerjemahkan teks dari satu bahasa ke bahasa lain dengan mempelajari terjemahan yang telah dibuat oleh manusia.
   1. **Supervised Learning**

Metode supervised learning didasarkan pada kumpulan sampel data yang memiliki label. Kumpulan sampel digunakan untuk meringkas karakteristik distribusi ukuran perilaku dalam setiap jenis aplikasi sehingga membentuk model perilaku dari data [4]. Supervised learning dikelompokkan lebih lanjut dalam masalah klasifikasi dan regresi. Masalah klasifikasi adalah ketika variabel output berbentuk kategori, seperti merah atau biru atau penyakit dan tidak ada penyakit. Sedangkan masalah regresi adalah ketika variabel output adalah nilai riil, seperti dollar atau berat.

* + 1. **Logistic Regression**

Regresi logistik adalah model prediktif yang digunakan untuk mengevaluasi hubungan antara variabel dependen (target) yang merupakan data kategorikal dengan skala nominal atau ordinal dan variabel independen (prediktor) yang merupakan data kategorikal dengan skala interval atau rasio. Algoritma ini juga dapat digunakan untuk pemodelan deret waktu untuk menemukan hubungan antar variabel yang terlibat. Regresi logistik adalah algoritma yang digunakan untuk memprediksi probabilitas variabel dependen kategori. Dalam regresi logistik, variabel dependen ditampilkan sebagai variabel biner yang bernilai 1 (ya) atau 0 (tidak). Model regresi logistik memprediksi sebagai fungsi X. Asumsi yang digunakan dalam regresi Logistik adalah sebagai berikut: regresi logistik biner membutuhkan variabel dependen biner, untuk regresi biner, tingkat faktor 1 dari variabel dependen harus mewakili hasil yang diinginkan, variabel independen harus independen satu sama lain. Dalam hal ini, model harus memiliki sedikit atau tidak ada multikolinearitas dan berhubungan secara linear dengan peluang log [5].

Berikut ini adalah contoh script sederhana untuk melakukan logistic regression menggunakan Python:



*Gambar 1.2.1 Contoh Script Logistic Regression*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

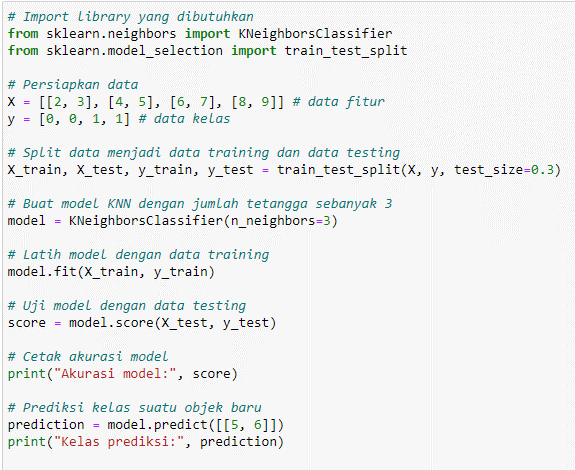
1. *Import library* yang dibutuhkan, yaitu ***LogisticRegression*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model *logistic regression*.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model *logistic regression* yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **K-Nearest Neighbors**

Metode KNN merupakan salah satu metode klasifikasi yang mudah. Metode ini bekerja dengan mencari k pola (diantara semua pola latih disemua kelas) yang terdekat dengan pola masukan kemudian menentukan kelas keputusan berdasarkan jumlah pola terbanyak [6]. Proses pelatihan KNN menghasilkan k yang memberikan akurasi tertinggi dalam menggeneralisasi data yang akan dating.

Berikut ini adalah contoh script sederhana untuk melakukan K-Nearest Neighbors (KNN) menggunakan Python:



*Gambar 1.2.2 Contoh Script KNN*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

1. *Import library* yang dibutuhkan, yaitu ***KNeighborsClassifier*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model KNN.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

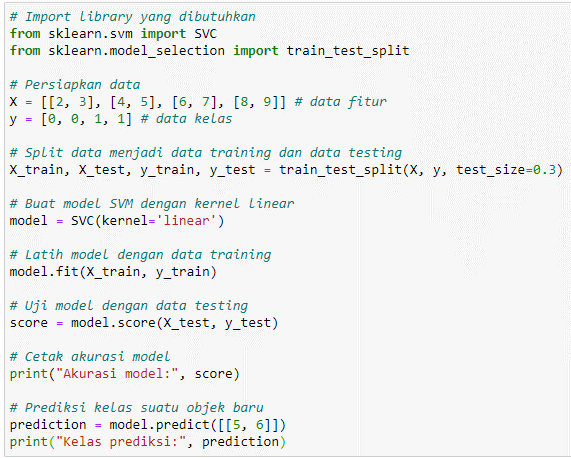
Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model KNN yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **Support Vector Machine**

Mesin vektor pendukung adalah algoritme kuat lainnya di teknologi pembelajaran mesin. Dalam mendukung mesin vector algoritma setiap item data diplot sebagai titik dalam n-dimensi ruang dan mendukung konstruksi algoritma mesin vector garis pemisah untuk klasifikasi dua kelas, pemisahan ini garis dikenal sebagai hyperplane.

Support vector machine mencari titik terdekat yang disebut sebagai dukungan vektor dan setelah menemukan titik terdekat itu menarik a jalur yang menghubungkan mereka. Dukung mesin vektor kalau begitu buatlah garis pemisah yang membagi dua dan tegak lurus jalur penghubung. Untuk mengklasifikasikan data dengan sempurna, margin harus maksimal. Di sini margin adalah jarak antara hyperplane dan support vector. [7].

Berikut ini adalah contoh script sederhana untuk melakukan Support Vector Machine (SVM) menggunakan Python:



*Gambar 1.2.3 Contoh Script SVM*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

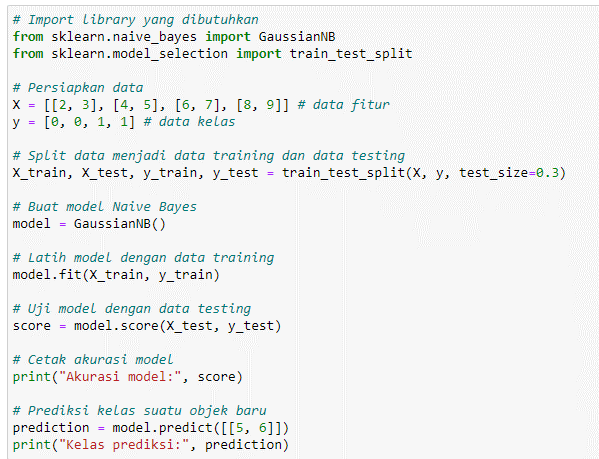
1. *Import library* yang dibutuhkan, yaitu ***SVC*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model SVM.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model SVM yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **Naïve Bayes**

Naive bayes termasuk ke dalam pembelajaran supervised, sehingga pada tahapan pembelajaran dibutuhkan data awal berupa data pelatihan untuk dapat mengambil keputusan. Pada tahapan pengklasifikasian akan dihitung nilai probabilitas dari masing-masing label kelas yang ada terhadap masukan yang diberikan. Label kelas yang memiliki nilai probabilitas paling besar yang akan dijadikan label kelas data masukan tersebut. Naive bayes merupakan perhitungan teorema bayes yang paling sederhana, karena mampu mengurangi kompleksitas komputasi menjadi multiplikasi sederhana dari probabilitas. [8].

Berikut ini adalah contoh script sederhana untuk melakukan Naive Bayes menggunakan Python:



*Gambar 1.2.4 Contoh Script Naïve Bayes*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

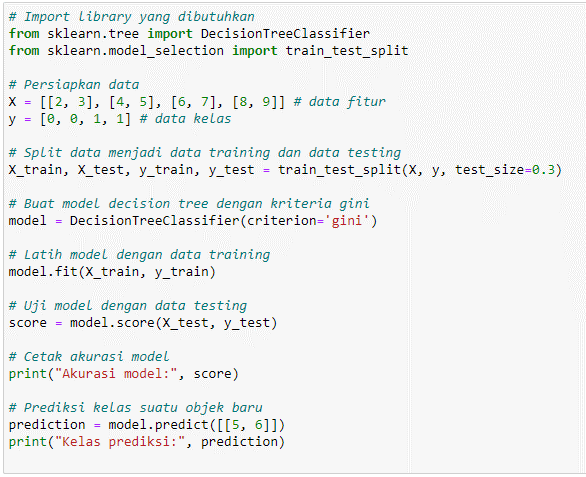
1. *Import library* yang dibutuhkan, yaitu ***GaussianNB*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model Naïve Bayes.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model Naïve Bayes yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **Decision Tree**

Algotirma decision tree merupakan algoritma yang umum digunakan untuk pengambilan keputusan. Decision tree akan mencari solusi permasalahan dengan menjadikan kriteria sebagai node yang saling berhubungan membentuk seperti struktur pohon [9]. Setiap pohon memiliki cabang, cabang mewakili suatu atribut yang harus dipenuhi untuk menuju cabang selanjutnya hingga berakhir di daun (tidak ada cabang lagi). Konsep data dalam decision tree adalah data dinyatakan dalam bentuk tabel yang terdiri dari atribut dan record.Atribut digunakan sebagai parameter yang dibuat sebagai kriteria dalam pembuatan pohon.

Berikut ini adalah contoh script sederhana untuk melakukan decision tree menggunakan Python:



*Gambar 1.2.5 Contoh Script Decision Tree*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

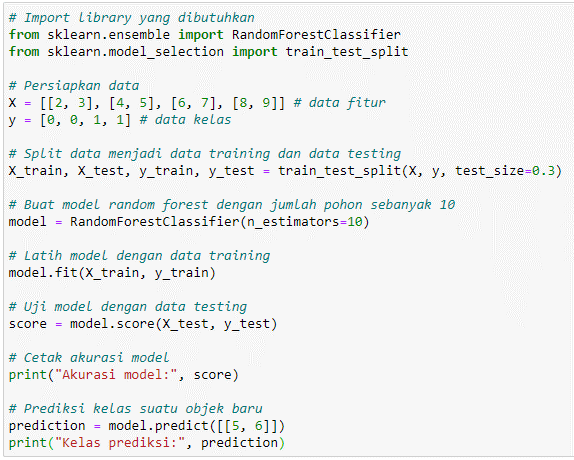
1. *Import library* yang dibutuhkan, yaitu ***DecissionTreeClassifier*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model *Decision Tree*.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model *Decision Tree* yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **Random Forest**

Metode random forest adalah pengembangan dari metode CART, yaitu dengan menerapkan metode bootstrap aggregating (bagging) dan random feature selection [10]. Dalam random forest, banyak pohon ditumbuhkan sehingga terbentuk hutan (forest), kemudian analisis dilakukan pada kumpulan pohon tersebut.

Berikut ini adalah contoh script sederhana untuk melakukan random forest menggunakan Python:



*Gambar 1.2.6 Contoh Script Random Forest*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

1. *Import library* yang dibutuhkan, yaitu ***RandomForestClassifier*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model *Random Forest*.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

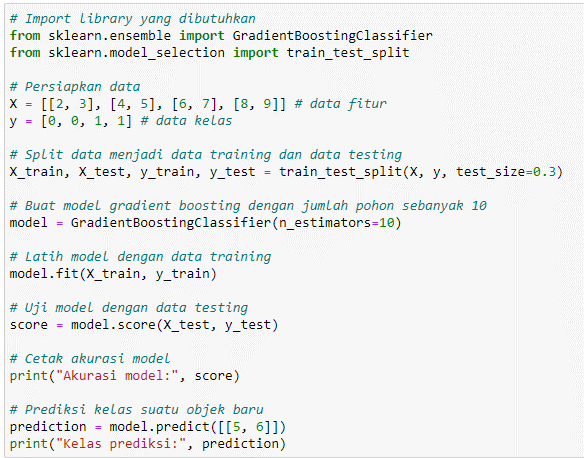
Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model *Random Forest* yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **Gradient Boosting**

Tujuan utama dari Boosting adalah menggabungkan semua train yang lemah bersama-sama untuk membentuk model yang kuat.

* Gradient boosting adalah suatu teknik yang sangat kuat untuk mengembangkan model prediktif. Ini berlaku untuk beberapa fungsi risiko dan mengoptimalkan akurasi prediksi model. Ini juga menyelesaikan masalah multikolinearitas di mana korelasi antar variabel prediktor tinggi.
* Gradient Boosting adalah algoritma pembelajaran mesin ansambel dan biasanya digunakan untuk menyelesaikan klasifikasi dan regresi [11].

Berikut ini adalah contoh script sederhana untuk melakukan gradient boosting menggunakan Python:



*Gambar 1.2.7 Contoh Script Gradient Boosting*

Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

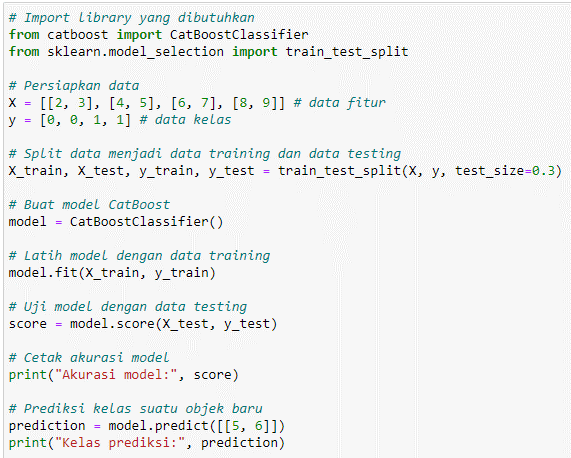
1. *Import library* yang dibutuhkan, yaitu ***GradientBoostingClassifier*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model *Gradient Boosting*.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model *Gradient Boosting* yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* + 1. **Catboost**

CatBoost menggunakan statistik target terurut dan peningkatan terurut yang menjadikannya bagus untuk nilai kategorikal data heterogen dan memiliki kinerja yang kuat relatif terhadap pohon keputusan penguat gradien lainnya implementasi. Sebuah studi terbaru menemukan salah satu isu penting dari penelitian interdisipliner mereka adalah sensitivitasnya ke hyper-parameter dan pentingnya penyetelan hyper-parameter [12]. Peneliti dapat mengatur pengaturan untuk jumlah maksimum iterasi yang digunakan CatBoost, kedalaman maksimum Pohon Keputusan konstituen, dan jumlah maksimum kombinasi fitur kategorikal untuk meningkatkan performa model. Nilai-nilai itu peneliti menggunakan hyper-parameter ini dapat menjelaskan perbedaan dalam kinerja CatBoost.

Berikut ini adalah contoh script sederhana untuk melakukan CatBoost menggunakan Python:



*Gambar 1.2.8 Contoh Script Catboost*

*­­* Penjelasan *script* diatas:

*Script* tersebut terdiri dari beberapa langkah:

1. *Import library* yang dibutuhkan, yaitu ***CatBoostClassifier*** dan ***train\_test\_split*** dari *scikit-learn.*
2. Persiapkan data fitur dan kelas yang akan digunakan untuk melatih model.
3. *Split* data menjadi data training dan data *testing* dengan ukuran data *testing* sebesar 30%.
4. Buat model *Cat Boost*.
5. Latih model dengan data *training*.
6. Uji model dengan data *testing* dan dapatkan skor akurasi.
7. Cetak akurasi model.
8. Prediksi kelas suatu objek baru.
9. Cetak kelas prediksi.

Setelah melakukan semua langkah tersebut, Anda akan mendapatkan model *Cat Boost* yang telah dilatih dan diuji serta dapat digunakan untuk melakukan prediksi kelas suatu objek baru.

* 1. **Python**

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile.Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televise kesukaan Guido Monty Python’s Flying Circus. Van Rossum mengembangkan Python sebagai hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak sintuitif dan memiliki pustaka yang luas [13].

**BAB 2**

**Pengenalan Tools**

Setelah Anda memahami materi pengenalan dasar tentang *machine learning* dan pengklasifikasi algoritma yang akan kita gunakan dalam membangun sistem deteksi *website phishing*, selanjutnya Anda perlu mengetahui *tools* apa saja yang akan digunakan dalam proses pembangunan sistem ini. Pastikan bahwa Anda sudah memasang *tools* tersebut di komputer Anda sebelum memulai pembangunan sistem deteksi *website phishing* sesuai dengan tutorial yang akan kita bahas di buku ini.

**2.1 Framework Flask**

Flask adalah sebuah web framework yang ditulis dengan bahasa Python dan tergolong sebagai jenis microframework . Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah web yang terstruktur dan dapat mengatur behaviour suatu web dengan lebih mudah. Flask termasuk pada jenis microframework karena tidak memerlukan suatu alat ataupun pustaka tertentu dalam penggunaannya. Sebagian besar fungsi dan komponen umum seperti validasi form, database, dan sebagainya tidak terpasang secara default di Flask [14]. Hal ini dikarenakan fungsi dan komponen-komponen tersebut sudah disediakan oleh pihak ketiga dan Flask dapat menggunakan ekstensi yang membuat fitur dan komponenkomponen tersebut seakan diimplementasikan oleh Flask sendiri.

**2.2 Jupyter Notebook**

Jupyter Notebook (file yang berekstensi ipynb) adalah dokumen yang dihasilkan oleh Jupyter Notebook App yang berisikan kode komputer dan rich text element seperti paragraf, persamaan matematik, gambar dan tautan (links). [15]

Jupyter Notebook tersedia dalam berbagai bahasa pemrograman, termasuk python, R, Julia, dan banyak lagi. Anda dapat menggunakan Jupyter Notebook untuk menulis dan menjalankan kode python secara interaktif, serta memvisualisasikan data dan hasilnya dengan mudah.

Untuk menggunakan Jupyter Notebook, Anda perlu memasang aplikasinya terlebih dahulu. Anda dapat mengikuti instruksi pemasangan di halaman resmi Jupyter Notebook di **https://jupyter.org/install**. Setelah terpasang, Anda dapat membuka Jupyter Notebook dari command prompt atau terminal dengan mengetik perintah "jupyter notebook" dan mengikuti instruksi selanjutnya untuk membuat dan menjalankan notebook.

**2.3 Visual Studio Code**

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows. Teks editor ini secara langsung mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace Visual Studio Code (seperti C++, C#, Python, Go, Java, dst).

Banyak sekali fitur-fitur yang disediakan oleh Visual Studio Code, diantaranya Intellisense, Git Integration, Debugging, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi Visual Studio Code. Pembaruan versi Visual Studio Code ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan VS Code dengan teks editor-teks editor yang lain. [16]

**2.4 Pickel**

Modul pickle mengimplementasikan protokol biner untuk serialisasi dan de-serialisasi struktur objek Python. "Pickling" adalah proses di mana hierarki objek Python diubah menjadi aliran byte, dan "unpickling" adalah operasi kebalikannya, di mana aliran byte (dari file biner atau objek mirip byte) diubah kembali menjadi hierarki objek. Pengawetan (dan penghilangan pengawetan) secara alternatif dikenal sebagai "serialisasi", "penyusunan," 1 atau "perataan"; namun, untuk menghindari kebingungan, istilah yang digunakan di sini adalah “mengawetkan” dan “mengurai”. [17]

**2.5 Web Browser**

Web browser disebut juga sebagai perambah, adalah perangkat lunak yang berfungsi menampilkan dan melakukan interaksi dengan dokumen-dokumen yang disediakan oleh server web. Browser pada umumnya juga mendukung berbagai jenis URL dan protokol, misalnya ftp: untuk file transfer protocol (FTP), rtsp: untuk real-time streaming protocol (RTSP), and https: untuk versi http yang terenkripsi (SSL). File format sebuah halaman web biasanya hyper-text markup language (HTML) dan diidentifikasikan dalam protokol HTTP menggunakan header MIME, format lainnya antara lain XML dan XHTML. Sebagian besar browser mendukung bermacam format tambahan pada HTML seperti format. gambar JPEG, PNG and GIF image formats, dan dapat dikembangkan dukungannya misal terhadap SVG dengan menambahkan/ menggunakan plugin. Ada beberapa web browser yang populer diantaranya Internet Explorer, Crome, Opera dan Mozilla. [18]

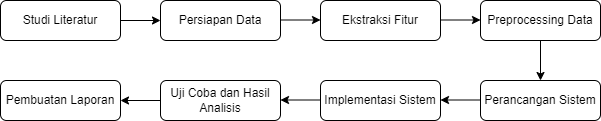
**BAB 3**

**Metodologi Penelitian**

**3.1 Diagram Alur Metodologi Penelitian**

Sistem yang diusulkan pada penelitian ini terdiri dari dua sistem utama, yaitu Sistem Evaluasi dan Implementasi Sistem, dimana implementasi sistem mengandung proses penting, yaitu fitur ektraksi, sementara pada Sistem Evaluasi akan ada menjadi proses pengoptimalan parameter dari algoritma klasifiasi yang akan digunakan. Algoritma yang akan digunakan dalam penelitian ini adalah Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting dan Catboost sebagai perbandingan kinerja sistem.

Tahapan yang akan dilakukan dalam penelitian ini dapat dilihat pada gambar berikut:



*Gambar 3.1 Diagram Alur Metodologi Penelitian*

Berdasarkan diagram pada diatas, secara umum penelitian dapat digambarkan sebagai berikut:

1. Studi Literatur merupakan tahapan awal dari penelitian ini. Tahapan ini dilakukan untuk mengumpulkan penelitian yang berkaitan dengan metode yang digunakan kepada ektraksi fitur dan klasifikasi.
2. Persiapan data merupakan langkah yang dilakukan untuk mendapatkan dataset yang akan digunakan dalam mengklasifikasikan website. Dataset yang digunakan bersumber dari Kaggle.
3. Ektraksi fitur dilakukan untuk mengekstraksi fiur terdapat di situs web berdasarkan dataset yang diperoleh dari Kaggle. Hasil ektraksi fitur ini kemudian akan digunakan untuk mendeteksi situs web phising.
4. Preprocessing Data dilakukan dalam bentuk menganalisis data yang akan digunakan untuk memilih data berdasarkan hasil ekstraksi ciri yang telah dilakukan sebelumnya.
5. Perancangan sistem pada penelitian ini dilakukan untuk menentukan algoritma yang akan dinguakan dalam klasifikasi situs web phising. Algoritma-algoritma yang digunakan dalam mengklasifikasi adalah Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Naïve Bayes, Decision Tree, Random Forest, Gradient Boosting dan Catboost sebagi perbandingan kinerja sistem. Pada tahapan ini dibuat flowchat yang berkaitan dengan alur kerja sistem.
6. Implementasi sistem dilakukan sesuai dengan flowchart yang telah dibuat ditahap sebelumnya. Pada penelitian ini, sistem dibuat dengan menggunakan bahasa pemrograman python dan framework flask.
7. Pengujian sistem dilakukan untuk mengetahui keakuratan sistem yang dibuat, sistem akan diuji dengan beberapa percobaan dan bentuk URL yang berbeda.
8. Tahapan akhir dalam penelitian ini adalah menulis laporan penelitian dalam bentuk laporan.

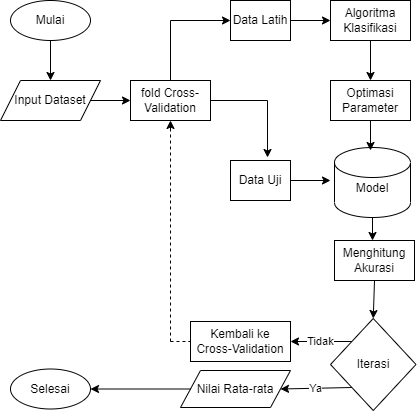
**3.2 Tahapan – tahapan Diagram Alur Metofologi Penelitian**

Pada tahapan penelitian ini dilakukan untuk mengektrak fitur yang terdapat pada situs web dengan memasukkan URL ke dalam sistem dankemudian sistem akan mengkase URL dan kemudian istem akan mengakses URL dan melakukan fitur ekstraksi di situs website. Karena untuk mengakses situs web yang akan dilakukan deteksi harus terhubung kedalam internet. Sistem ini secara otomatis menjadi sistem online.

Sistem yang dibuat terdiri dari dua bagian yaitu sistem pelaksanaan dan sistem evaluasi. Implementasi sistem dapat digunakan ketika pengguna penginputkan URL kedalam form. Sedangkan sistem evaluasi adalah sistem dibuat untuk menganalisis kinerja dari sistem implementasi.

**3.2.1 Sistem Evaluasi**

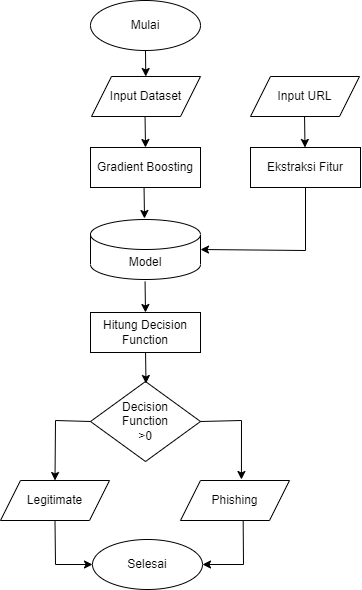
Sistem evaluasi merupakan sistem yang dirancang sebagai untuk evaluasi sistem produksi. Evaluasi terhadap sistem ini dilakukan dengan cara mengevaluasi dataset yang digunakan dalam implementasi sistem. Flowchart sistem evalusi dapat dilihat pada gambar berikut:



*Gambar 3.2.1 Flowchart Sistem Evaluasi*

**3.2.2 Sistem Implementasi**

Sistem implementasi merupakan sistem yang dirancang untuk digunakan dan diimplementasikan dalam melakukan deteksi website phising. Hal ini yang membedakan sistem implementasi dengan sistem evaluasi adalah sistem ini dirancang untuk digunakan oleh pengguna dengan input berupa URL. Selain ini perbedaannya terdapat pada proses ekstraksi yang sangat penting dalam menentukan kinerja sistem. Untuk lebih jelasnya dapat dilihat pada flowchart sistem berikut:

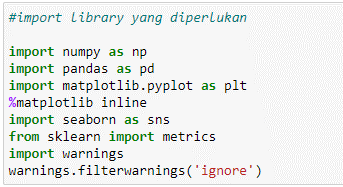


*Gambar 3.2.2 Flowchart Sistem Impelemtasi*

**BAB 4**

**Pembuatan Model Machine Learning**

**4.1** **Import Library Tahap Awal**



*Gambar 4.1 Import Library Tahap Awal*

Pada script di atas, pertama-tama terdapat beberapa library yang akan diimpor ke dalam program. Library yang diimpor adalah:

1. NumPy: Library ini menyediakan fungsi-fungsi matematika yang berguna untuk melakukan operasi pada array.
2. pandas: Library ini memungkinkan kita untuk membaca dan mengelola data serta menyediakan berbagai fungsi untuk mengeksplorasi data.
3. Matplotlib: Library ini memungkinkan kita untuk membuat berbagai jenis plot, seperti plot garis, plot bar, dan lain-lain.
4. seaborn: Library ini merupakan library yang berbasis pada Matplotlib dan menyediakan tipe plot yang lebih bervariasi serta lebih mudah digunakan.
5. sklearn: Library ini merupakan library yang menyediakan berbagai algoritma machine learning yang berguna untuk melakukan pembelajaran mesin.

Pada akhir script, terdapat perintah "warnings.filterwarnings('ignore')" yang berfungsi untuk menonaktifkan warning yang mungkin muncul saat menjalankan program.

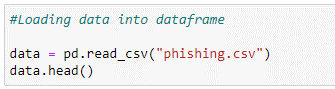
**4.2 Memuat Dataset**

Sumber Dataset https://www.kaggle.com/eswarchandt

/phishing-website-detector.

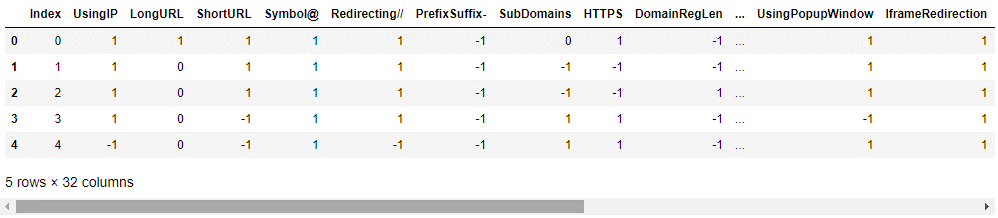
Kumpulan URL situs web untuk 11000+ situs web. Setiap sampel memiliki 30 parameter situs web dan label kelas yang mengidentifikasinya sebagai situs web phishing atau bukan (1 atau -1).

Gambaran umum dari dataset ini adalah, memiliki 11054 sampel dengan 32 fitur. Unduh dataset dari tautan yang disediakan.



*Gambar 4.2.1 Memuat Dataset*

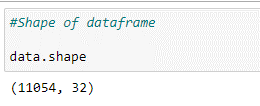
Hasil:



*Gambar 4.2.2 Hasil Memuat Dataset*

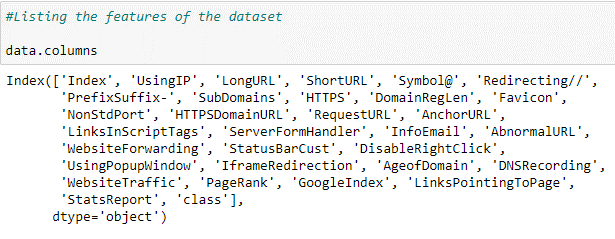
**4.3 Familiar dengan data & EDA**

EDA merupakan singkatan dari *Exploratory Data Analysis* (Analisis Data Eksploratif). EDA merupakan suatu proses yang dilakukan untuk mengeksplorasi data dengan tujuan untuk menemukan pola-pola yang terdapat dalam data tersebut.



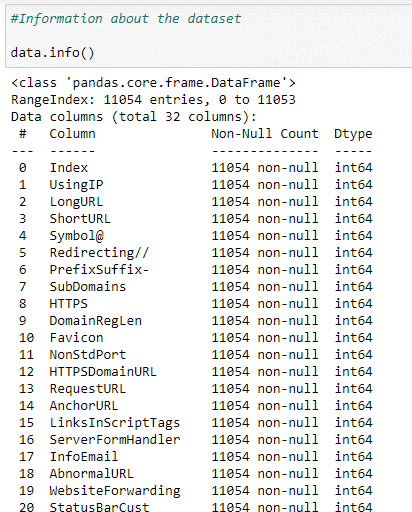
*Gambar 4.3.1 Data Shape*

Fungsi ***shape*** adalah suatu atribut yang dimiliki oleh object ***DataFrame*** yang terdapat dalam library pandas. Atribut ini mengembalikan tuple yang menunjukkan dimensi dari data, yaitu jumlah baris dan jumlah kolom.



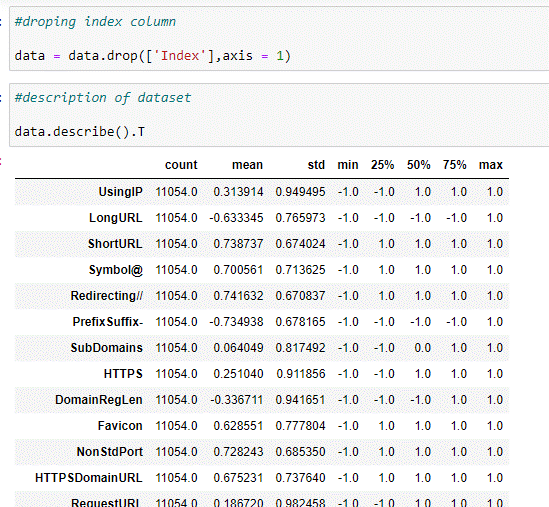
*Gambar 4.3.2 Data Columns*

Fungsi ***columns*** adalah suatu atribut yang dimiliki oleh ***object DataFrame*** yang terdapat dalam library pandas. Atribut ini mengembalikan nama-nama kolom dari data yang tersimpan dalam ***object DataFrame*** tersebut.



*Gambar 4.3.3 Data Info*

Fungsi ***info*** adalah suatu method yang dimiliki oleh ***object DataFrame*** yang terdapat dalam *library* pandas. Method ini digunakan untuk menampilkan informasi mengenai tipe data, jumlah baris, dan nama-nama kolom yang terdapat dalam data.

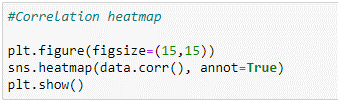


*Gambar 4.3.4 Data Info*

Fungsi ***describe*** adalah suatu method yang dimiliki oleh object ***DataFrame*** yang terdapat dalam library pandas. Method ini digunakan untuk menampilkan statistik deskriptif dari data, seperti mean, std, min, max, dan quartile.

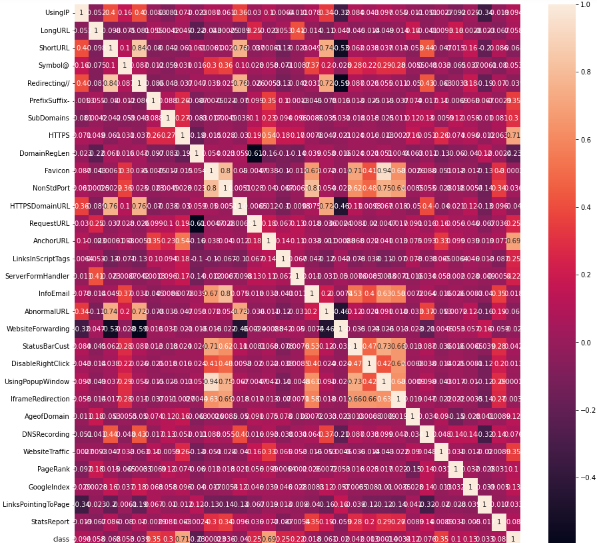
Method **T** merupakan method yang dimiliki oleh object DataFrame yang digunakan untuk men-transpose tabel, yaitu menukar posisi baris dan kolom.

**4.4 Memvisualisasikan Data**

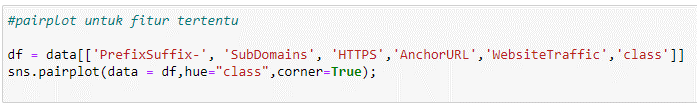


*Gambar 4.4.1 Correlation heatmap*

Pada script di atas, pertama-tama dilakukan inisialisasi figure dengan ukuran 15x15 inch menggunakan perintah **plt.figure(figsize=(15,15))**. Kemudian, dilakukan plot heatmap dengan menggunakan perintah **sns.heatmap().** Argumen yang diberikan kepada perintah ini adalah matriks korelasi dari data **(data.corr())** serta argumen **annot=True** yang digunakan untuk menampilkan nilai korelasi pada setiap sel heatmap. Terakhir, plot ditampilkan dengan perintah **plt.show().**

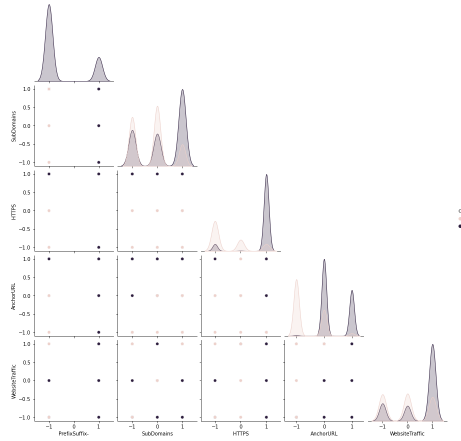


*Gambar 4.4.2 Hasil Correlation heatmap*

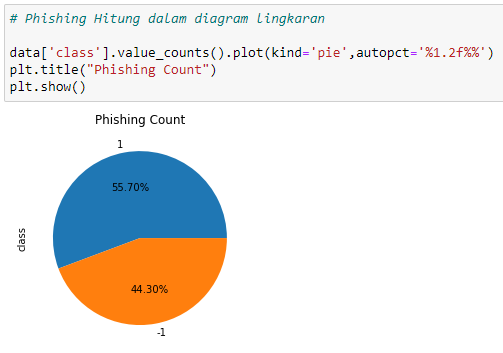


*Gambar 4.4.3 Pairplot*

Pada script di atas, pertama-tama dilakukan pemilihan fitur yang akan dianalisis dengan menggunakan notasi *indexing* pada *object* ***DataFrame***. Kemudian, dilakukan plot pairplot dengan menggunakan perintah ***sns.pairplot().*** Argumen yang diberikan kepada perintah ini adalah data yang akan dianalisis ***(data = df)*,** kolom yang akan digunakan sebagai hue ***(hue="class")*,** dan argumen ***corner=True*** yang digunakan untuk menampilkan plot diagonal yang menunjukkan distribusi data pada setiap fitur.



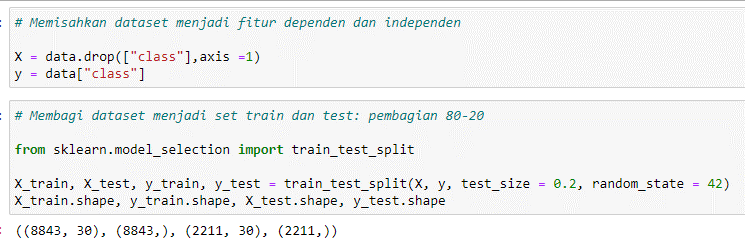
*Gambar 4.4.4 Hasil Pairplot*



*Gambar 4.4.4 Diagram Lingkaran*

Pada script di atas, pertama-tama dilakukan penghitungan jumlah kelas yang terdapat dalam data dengan menggunakan method ***value\_counts().*** Kemudian, dilakukan plot diagram lingkaran dengan menggunakan perintah ***plot(kind='pie')***. Argumen ***autopct='%1.2f%%'*** digunakan untuk menampilkan nilai persentase dari setiap kelas pada diagram lingkaran. Terakhir, judul plot ditambahkan dengan perintah ***plt.title()*** dan plot ditampilkan dengan perintah ***plt.show()***.

**4.5 Mimisahkan Dataset**



*Gambar 4.4.5 Memisahkan Dataset*

Pada script di atas, pertama-tama dilakukan *import module* ***train\_test\_split*** dari *library* ***sklearn. model\_selection*.** Kemudian, *module* tersebut digunakan untuk membagi dataset menjadi *set train* dan *set test* dengan perintah ***train\_test\_split(X, y, test\_size = 0.2, random\_state = 42)***. Argumen X dan y merupakan fitur dan label data, **test\_size** merupakan proporsi data yang akan digunakan sebagai set test (dalam hal ini 0.2 atau 20%), dan ***random\_state*** merupakan ***seed*** yang digunakan untuk memastikan bahwa *set train* dan *set test* yang dihasilkan tidak berubah-ubah setiap kali *script* dijalankan.

Setelah dataset terbagi, maka akan dihasilkan 4 *object* yaitu ***X\_train*** dan ***y\_train*** yang merupakan *set train*, serta ***X\_test*** dan ***y\_test*** yang merupakan *set test*. Kemudian, *object* tersebut ditampilkan dengan perintah ***X\_train.shape***, ***y\_train.shape***, ***X\_test.shape***, dan ***y\_test.shape*** untuk menampilkan dimensi dari setiap *object*.

Pembagian dataset menjadi *set train* dan *set test* merupakan langkah yang penting dalam proses pembuatan model *machine learning*. *Set train* digunakan untuk melatih model sedangkan *set test* digunakan untuk mengevaluasi model yang telah dilatih. Dengan membagi dataset menjadi *set train* dan *set test*, kita dapat memastikan bahwa model yang dihasilkan tidak hanya dapat menangani data yang telah diketahui tapi juga dapat menangani data baru yang belum pernah dilihat sebelumnya.

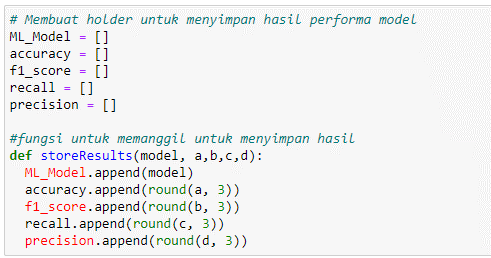
**4.6 Membangun dan Melatih Model**

Supervised adalah salah satu jenis pembelajaran mesin yang paling umum digunakan dan berhasil. Pembelajaran yang supervised digunakan kapan pun ingin memprediksi hasil/label tertentu dari sekumpulan fitur tertentu, dan memiliki contoh pasangan fitur-label.Membangun model pembelajaran mesin dari pasangan fitur-label ini, yang terdiri dari set train. Tujuannya adalah membuat prediksi akurat untuk data baru.

Ada dua jenis utama masalah supervised machine learning, yang disebut klasifikasi dan regresi. Kumpulan data berada di bawah masalah regresi. Model pembelajaran mesin supervised (regresi) yang dianggap melatih kumpulan data di notebook ini adalah:

1. Logistic Regression
2. k-Nearest Neighbors
3. Support Vector Clasifier
4. Naive Bayes
5. Decision Tree
6. Random Forest
7. Gradient Boosting
8. Catboost
9. Multilayer Perceptrons

Metrik yang dipertimbangkan untuk mengevaluasi performa model adalah Accuracy & F1 score.

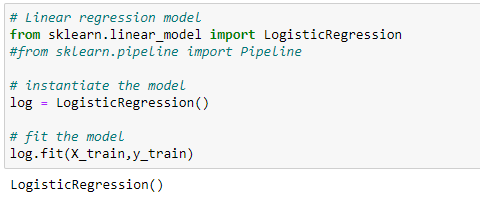


*Gambar 4.6 Membuat Holder Model*

Pada script di atas, pertama-tama dibuat empat list yaitu ***ML\_Model, accuracy, f1\_score, recall***, dan ***precision*** yang akan digunakan untuk menyimpan hasil performa model machine learning. Kemudian, dibuat sebuah fungsi ***storeResults()*** yang memiliki empat argumen yaitu ***model***, a, b, c, dan d. Fungsi ini akan menyimpan argumen argumen tersebut ke dalam masing-masing list yang telah dibuat sebelumnya.

Fungsi ***storeResults()*** ini akan berguna untuk menyimpan hasil performa model machine learning yang diuji. Dengan menyimpan hasil performa tersebut, kita dapat membandingkan performa model yang berbeda serta menentukan model terbaik yang akan digunakan pada data.

**4.6.1 Model Logistic Regression**



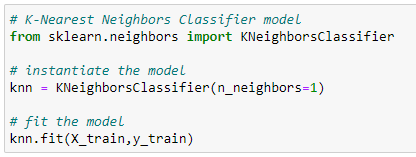
*Gambar 4.6.1 Model Logistic Regression*

Script di atas memuat sebuah model regresi logistik yang akan dilatih menggunakan data latih. Terdapat beberapa bagian yang terdapat pada script di atas, yaitu:

1. Import kelas ***LogisticRegression*** dari modul ***sklearn.linear\_model***. Kelas LogisticRegression merupakan kelas yang digunakan untuk membuat model regresi logistik.
2. Instansiasi objek model dengan menggunakan kelas ***LogisticRegression***. Objek model ini akan dibuat dengan menggunakan default parameter yang telah ditentukan oleh kelas ***LogisticRegression***.
3. Melatih model dengan menggunakan data latih dengan memanggil fungsi ***fit*** pada objek model yang telah dibuat. Fungsi ***fit*** akan melatih model dengan menggunakan data latih yang diberikan.

Setelah model dilatih, kita dapat menggunakannya untuk memprediksi target pada data uji dengan memanggil fungsi ***predict*** pada objek model yang telah dilatih. Selanjutnya, kita dapat mengevaluasi performa model dengan menghitung beberapa metrik seperti akurasi, f1-score, recall, precision dan mulai membuat scriptnya.

**4.6.2 Model k-Nearest Neighbors**



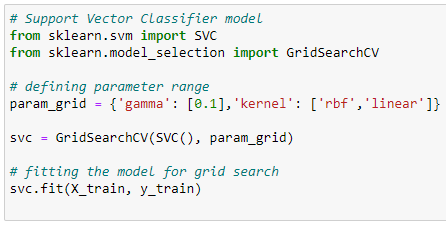
*Gambar 4.6.2 Model KNN*

Script di atas memuat sebuah model K-Nearest Neighbors (KNN) yang akan dilatih menggunakan data latih. Terdapat beberapa bagian yang terdapat pada script di atas, yaitu:

1. Import kelas ***KNeighborsClassifier*** dari modul ***sklearn.neighbors***. Kelas ***KNeighborsClassifier*** merupakan kelas yang digunakan untuk membuat model KNN.
2. Instansiasi objek model dengan menggunakan kelas ***KNeighborsClassifier***. Objek model ini akan dibuat dengan menggunakan parameter ***n\_neighbors=1***, yang menandakan bahwa model KNN yang akan dibuat akan menggunakan 1 tetangga terdekat dari setiap titik data.
3. Melatih model dengan menggunakan data latih dengan memanggil fungsi ***fit*** pada objek model yang telah dibuat. Fungsi ***fit*** akan melatih model dengan menggunakan data latih yang diberikan.

Setelah model dilatih, kita dapat menggunakannya untuk memprediksi target pada data uji dengan memanggil fungsi predict pada objek model yang telah dilatih. Selanjutnya, kita dapat mengevaluasi performa model dengan menghitung beberapa metrik seperti akurasi*, f1-score, recall,* precision dan mulai membuat scriptnya.

**4.6.3 Model SVM**



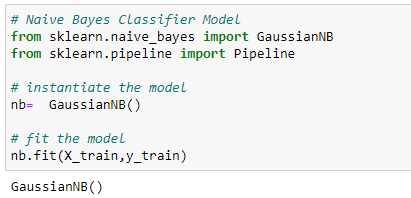
*Gambar 4.6.3 Model SVM*

Script di atas memuat sebuah model SVM yang akan dilatih menggunakan data latih. Terdapat beberapa bagian yang terdapat pada script di atas, yaitu:

1. Import kelas ***SVC*** dari modul ***sklearn.svm***. Kelas ***SVC*** merupakan kelas yang digunakan untuk membuat model SVM.
2. Instansiasi objek model dengan menggunakan kelas ***SVC*** Objek model ini akan dibuat dengan menggunakan parameter ***param grid***.
3. Melatih model dengan menggunakan data latih dengan memanggil fungsi ***fit*** pada objek model yang telah dibuat. Fungsi ***fit*** akan melatih model dengan menggunakan data latih yang diberikan.

Setelah model dilatih, kita dapat menggunakannya untuk memprediksi target pada data uji dengan memanggil fungsi predict pada objek model yang telah dilatih. Selanjutnya, kita dapat mengevaluasi performa model dengan menghitung beberapa metrik seperti akurasi*, f1-score, recall,* precision dan mulai membuat scriptnya.

**4.6.4 Model Naïve Bayes**



*Gambar 4.6.4 Model Naïve Bayes*

**4.6.5 Model Decision Tree**

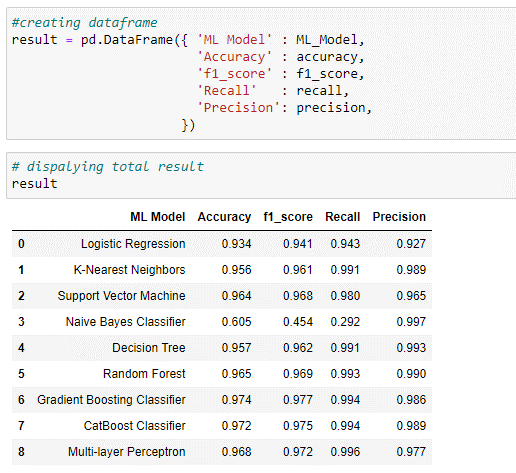
**4.6.6 Model Random Forest**

**4.6.7 Model Gradient Boosting**

**4.6.8 Model Catboost**

**4.6.9 Model Multilayer Perceptrons**

**4.7 Pendandingan Model**

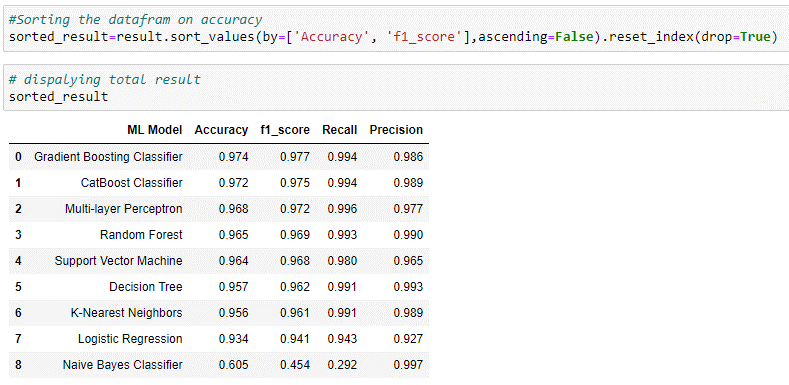


*Gambar 4.7.1 Membuat Dataframe Perbandingan Model*

Pada *script* di atas, dibuat sebuah *object result* yang merupakan sebuah ***DataFrame*** dengan menggunakan *library pandas*. ***DataFrame*** dibuat dengan menggunakan argumen yang berisi *dictionary* dengan *key* sebagai nama kolom dan *value* sebagai isi dari kolom tersebut.

***DataFrame*** merupakan tipe data yang terdiri dari baris dan kolom yang mirip dengan tabel pada *database*. ***DataFrame*** dapat dibuat dengan menggunakan data yang berasal dari file atau dengan memasukkan data secara manual seperti yang telah dilakukan pada ***script*** di atas.

Dalam kasus ini, ***DataFrame*** yang dihasilkan akan berisi informasi mengenai nama model *machine learning*, akurasi, *f1-score, recall*, dan *precision*. Informasi tersebut akan disimpan dalam masing-masing kolom yang telah didefinisikan sebelumnya. ***DataFrame*** ini akan berguna untuk menyimpan dan menampilkan hasil performa model *machine learning* dengan lebih mudah.

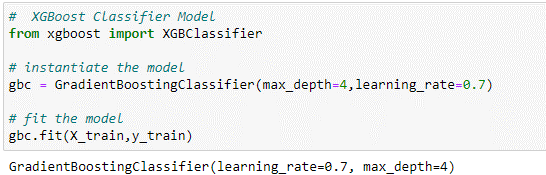


*Gambar 4.7.2 Sorting Dataframe Accuracy*

Dalam kasus ini, ***DataFrame*** *result* akan disortir berdasarkan kolom *Accuracy* dan ***f1\_score*** dengan urutan *descending* (nilai yang lebih tinggi lebih dulu). Kemudian, ***method reset\_index()*** dengan argumen ***drop=True*** digunakan untuk menghilangkan index yang lama dan menggantinya dengan index yang baru yang diurutkan sesuai dengan urutan data.

Hasil dari *sorting* tersebut disimpan ke dalam *object* baru yaitu ***sorted\_result***. *Sorting* pada ***DataFrame*** dapat berguna untuk menyusun data sesuai dengan prioritas yang diinginkan sehingga lebih mudah untuk dianalisis.

**4.8 Menyimpan Model Terbaik**

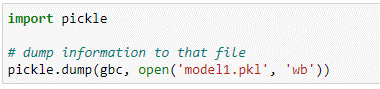


*Gambar 4.8 Menyimpan Model Terbaik*

Pada script di atas, pertama-tama dilakukan *import library* ***XGBClassifier*** dari *library* ***xgboost***. Kemudian, dilakukan instansiasi object gbc dengan menggunakan *class* ***XGBClassifier***. Argumen yang diberikan kepada class tersebut adalah *max\_depth* yang merupakan jumlah maksimum dari tingkat dalam pohon model, dan *learning\_rate* yang merupakan tingkat pembelajaran dari model.

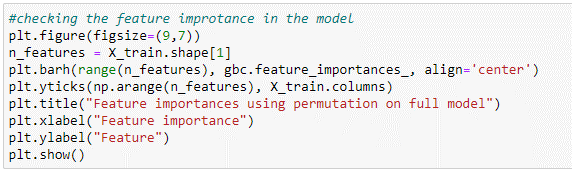
Setelah *object gbc* terbuat, dilakukan *training* model dengan menggunakan *method f****it()***. *Method* ini membutuhkan dua argumen yaitu *X\_train* dan *y\_train* yang merupakan *set train* yang telah dibuat sebelumnya. Setelah *training* selesai, maka model yang telah dilatih akan tersimpan dalam *object gbc*.

**4.9 Ubah Menjadi Format Pickle**



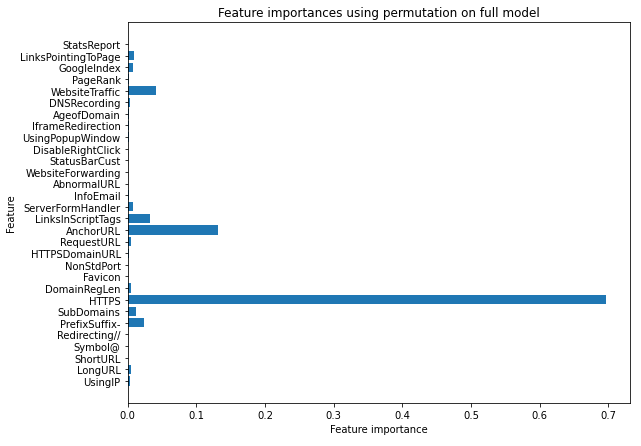
*Gambar 4.9.1 Import Library pickle*

Pada script di atas, dilakukan ***import library pickle***. Kemudian, dilakukan proses serialisasi dengan menggunakan *method* ***dump()*** dari ***library pickle***. *Method* ini membutuhkan dua argumen yaitu object yang akan diserialisasi (dalam hal ini *object gbc* yang merupakan model yang telah dilatih) dan file yang akan digunakan untuk menyimpan hasil serialisasi (dalam hal ini file **model1.pkl**).



*Gambar 4.9.2 Cek Fitur pada Model*

Pada script di atas, dilakukan *plot bar* dengan menggunakan *library matplotlib*. *Figure* yang dibuat akan memiliki ukuran 9x7. Kemudian, dihitung jumlah fitur yang terdapat dalam *set train* dengan ***X\_train.shape[1]*** dan dibuat *bar* dengan *method barh()* yang memiliki argumen ***range(n\_features)*** sebagai posisi sumbu x, *gbc.feature\_importances\_* sebagai lebar bar, dan *align='center'* sebagai posisi sumbu y. Kemudian, ditambahkan label pada sumbu y dengan menggunakan method *yticks()* dan menampilkan nama-nama fitur yang terdapat dalam set train dengan ***X\_train.columns***. Terakhir, ditambahkan judul plot dengan method title(), label sumbu x dengan method xlabel(), dan label sumbu y dengan *method* ylabel(). Plot tersebut ditampilkan dengan method ***show()***.



*Gambar 4.9.3 Hasil Cek Fitur pada Model*

| **61**

**CONTOH DAFTAR PUSTAKA**

Blogs.reed.edu. 2017. Text analysis using Voyant Tools. [online] Available at: <https://blogs.reed.edu/ed- tech/2017/03/text-analysis-using-voyant-tools/> [Accessed 11 September 2022].

Calado, F., 2018. Using Voyant-Tools to Formulate Research Questions for Textual Data - GC Digital Fellows. [online] GC Digital Fellows. Available at:

<https://digitalfellows.commons.gc.cuny.edu/2018/11/01

/using-voyant-tools-to-formulate-research-questions-for- textual- data/#:~:text=One%20of%20the%20benefits%20of,text

%20is%20displayed%20for%20reading.> [Accessed 11

August 2022].

En.wikipedia.org. 2022. Voyant Tools - Wikipedia. [online] Available at:

<https://en.wikipedia.org/wiki/Voyant\_Tools#:~:text=Ran ge%20of%20Uses-

,History,Lisa%20Goddard%2C%20and%20Mark%20Turc ato.> [Accessed 10 August 2022].

GitHub. 2016. *GitHub - voyanttools/Voyant*. [online] Available at:

<https://github.com/voyanttools/Voyant> [Accessed 10

August 2022].

-oo00oo-

DIHALAMAN INI, SILAHKAN ANDA ISI DENGAN PROFIL PENULIS, BAIK ITU PEMBIMBING INTERNSHP 1 DAN MAHASISWA

BUAT COVER BELAKANG BUKU

YANG BERISI :

SINOPSIS DARI BUKU YANG ANDA TULIS, SINOPSIS DAPAT DIAMBIL DARI ABSTRAK PADA JURNAL YANG ANDA BUAT

CATATAN :

DESAIN HARUS ORIGINAL