LAPORAN PRAKTIKUM PM

PENERAPAN METODE KERNEL PADA DATASET OPP

Dede Masita [1)], Ayu Erlinawati [2)],Sella Dianka Fitri [3)], Rizki Adrian Bennovry [4)] , Saiful Haris Muhammad [5)] ,Muhammad Kaisar Firdaus [6)]

Program Studi Sains Data, Jurusan Sains, Institut Teknologi Sumatera

Email : dede.120450007@student.itera.ac.id [1] ayu.1204500025@student.itera.ac.id [2)]sella.1204500043@student.itera.ac.id [3)],rizki.120450073@student.itera.ac.id [4)] saiful.120450115@student.itera.ac.id[5)]muhammad.120450135@student.itera.ac.id[6]

Abstrak

Studi ini meneliti analisis morfometrik Possum dengan fokus pada metode Perceptron Kernel menggunakan data dari 'opp.csv'. Pendekatan ini memungkinkan klasifikasi biner untuk memisahkan dan mengidentifikasi Possum berdasarkan ciri morfometrik, yang mengungkap variasi populasi mamalia ini. Tujuan penelitian ini adalah memperoleh pemahaman yang lebih mendalam tentang bentuk, karakteristik, dan variasi jenis Possum. Hasil penelitian ini dapat memberikan wawasan mendalam tentang diversitas spesies Possum , memberikan landasan yang kuat untuk pemahaman lebih lanjut terkait ekologi dan evolusi mamalia .

Kata kunci : Possum, akurasi, dan metode Perceptron Kernel.

1. Pendahuluan

1.1 Latar Belakang

Pada era big data dan pengolahan informasi yang semakin berkembang, analisis data menjadi elemen kunci dalam pengambilan keputusan yang informasional. Namun, banyak dataset di kehidupan sehari-hari yang tidak selalu memiliki hubungan antar variabel yang bersifat linier. Sebagai contoh, dalam bidang biologi dan ilmu hayati, pengamatan tentang karakteristik morfometrik dari organisme seringkali melibatkan hubungan kompleks antara berbagai parameter fisik. Dataset OPP yang berisi informasi morfometrik adalah salah satu contoh dataset yang tidak memiliki hubungan antar variabelnya yang bersifat linear. Dalam kasus ini , metode kernel dapat digunakan untuk pembuatan modelnya. Kernel perceptron memungkinkan penentuan keputusan terbaik dengan memanfaatkan teknik pemetaan data ke ruang fitur yang lebih tinggi. Dalam bidang biologi, informasi mengenai karakteristik hewan merupakan hal yang penting karena dapat memberikan pengetahuan dalam spesies serta adaptasi lingkungannya. Karena itu, aplikasi metode kernel pada dataset OPP dapat memberikan kontribusi yang besar dalam pemahaman lebih dalam pada kasus bidang biologi kali ini.

1.2 Tujuan

Tujuan dari aplikasi metode kernel perceptron pada dataset OPP kali ini adalah untuk meningkatkan kemampuan model dalam menangkap pola dan relasi kompleks yang mungkin bersifat non-linear dalam informasi morfometrik. Dengan memanfaatkan kekuatan metode kernel, sehingga dapat ditemukan struktur yang lebih mendalam di antara parameter morfometrik, yang akan sulit diakses oleh model linear konvensional.. Sehingga dapat memberikan wawasan yang lebih baik tentang karakteristik biologis dan hubungan evolusioner dalam dataset OPP, sehingga didapat informasi yang lebih efektif dan akurat di bidang biologi.

2. Metode penelitian

2.1 Dataset

Dataset yang digunakan pada penelitian ini adalah dataset *opp.csv* yang kami ambil dari kaggle. Dataset *opp.csv* terdiri dari 101 entri dan 11 kolom yang menyajikan informasi terkait beberapa individu dalam suatu populasi. Setiap baris dalam dataset ini mewakili satu individu dalam populasi, sedangkan kolom-kolomnya mendefinisikan berbagai karakteristik pada setiap individunya. Berikut nama 11 kolom dalam dataset ini: sex, age, hdlngth, skullw, totlngth, taill, footlgth, earconch, eye, chest, dan belly.

Di bawah ini adalah penjelasan mengenai setiap kolom pada dataset:

- sex: Menunjukkan jenis kelamin (male/female) dari subjek dalam pengukuran.
- age: Merupakan variabel numerik yang mewakili usia subjek dalam pengukuran.
- hdlngth: Mengacu pada panjang kepala.
- skullw: Merupakan lebar tengkorak.
- totlngth: Panjang total subjek, mungkin mencakup panjang kepala, tubuh, dan ekor.
- taill: Panjang ekor subjek.
- footlgth: Panjang kaki atau kaki belakang subjek.
- earconch: Mungkin merujuk pada panjang atau dimensi kuping atau cangkang telinga.
- eye: Mengacu pada panjang mata atau dimensi mata subjek.
- chest: Ukuran dada subjek.
- belly: Ukuran perut subjek.

2.2 Metode

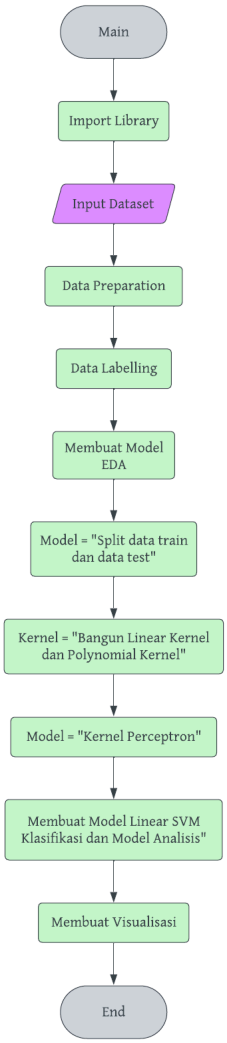
Sebelum melakukan , kami melakukan data understanding and preparation terlebih dahulu untuk mengetahui sekaligus meningkatkan kualitas data, selanjutnya kami memuat beberapa langkah dalam penelitian kami sebagai berikut:

1. Preprocessing Data
   Preprocessing Data melibatkan pembersihan dan mempersiapkan data untuk analisis. Hal ini melibatkan penanganan missing value, transformasi variabel, penanganan outlier dan langkah-langkah yang lainnya.
2. Exploratory Data Analysis (EDA)

EDA melibatkan perhitungan dan pengevaluasian statistik dasar untuk mendapatkan gambaran tentang kecenderungan pusat dan variasi pada data. Di dalam EDA juga melibatkan data visualization yang dimana bertujuan untuk mengeksplorasi distribusi dan hubungan antar variabel pada data.

3. Split Data dan Testing Data
   Pada metode ini kita membagi dataset menjadi 2 bagian yaitu train data dan test data. Tujuannya adalah membantu mengukur sejauh mana model yang dikembangkan dapat diterapkan pada data baru yang tidak terlihat. Pada penelitian yang kami buat kita membagi data dengan rasio 80/20 dengan porsi lebih besar untuk training data. Tujuannya untuk memastikan model tidak overfitting pada data training saat dijalankan.
4. Fungsi Kernel
   Fungsi kernel adalah metode untuk mengukur seberapa mirip dua data point dalam bentuk fungsi matematis. Dalam konteks metode SVM ini digunakan untuk memungkinkan pembelajaran di ruang fitur yang lebih tinggi.
5. Kernel Perceptron Model
   Perceptron adalah algoritma pembelajaran mesin untuk klasifikasi biner. Dalam konteks kernel perceptron, fungsi kernel digunakan untuk mengubah data ke dalam ruang dimensi yang lebih tinggi untuk memungkinkan pemisahan yang lebih baik antara kelas.
6. Linear SVM Classification
   Support Vector Machine (SVM) adalah algoritma pembelajaran mesin untuk klasifikasi dan regresi. Dalam kasus SVM linear, model mencoba menemukan garis pemisah linear terbaik antara dua kelas.
7. Visualize Results:
   Membuat visualisasi untuk mengevaluasi dan memahami hasil dari model yang telah dilatih. Ini bisa mencakup plot keputusan (decision boundaries), confusion matrix, atau visualisasi lainnya untuk menganalisis kinerja model.

2.3 Flowchart



2.3 Pseudocode

Berikut pseudocode pada flowchart diatas:
Fungsi main()

// Import library
Import library yang dibutuhkan

// Input dataset
Input dataset dari sumber yang diinginkan

// Data preparation
Lakukan persiapan data, seperti pembersihan data dan transformasi data

// Data labelling
Lakukan pelabelan data, jika diperlukan

// Membuat model
Buat model pembelajaran mesin yang diinginkan, seperti SVM, Linear Regression, atau Random Forest

// EDA
Lakukan analisis data eksploratif untuk memahami data

// Split data
Bagi data menjadi data pelatihan dan data uji

// Kernel
Bangun kernel yang diinginkan, seperti linear kernel atau polynomial kernel

// Membuat model
Buat model SVM dengan kernel yang diinginkan

// Klasifikasi dan model analisis
Lakukan klasifikasi data dan analisis model

// Membuat visualisasi
Buat visualisasi dari hasil klasifikasi dan analisis model

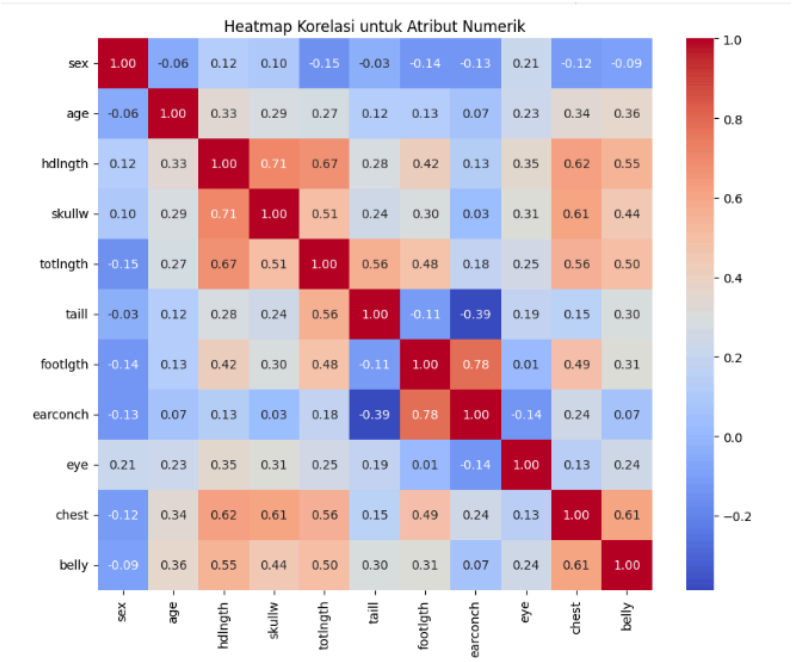
Akhir fungsi main()

3. Hasil dan pembahasan

3.1 Data Understanding & Processing

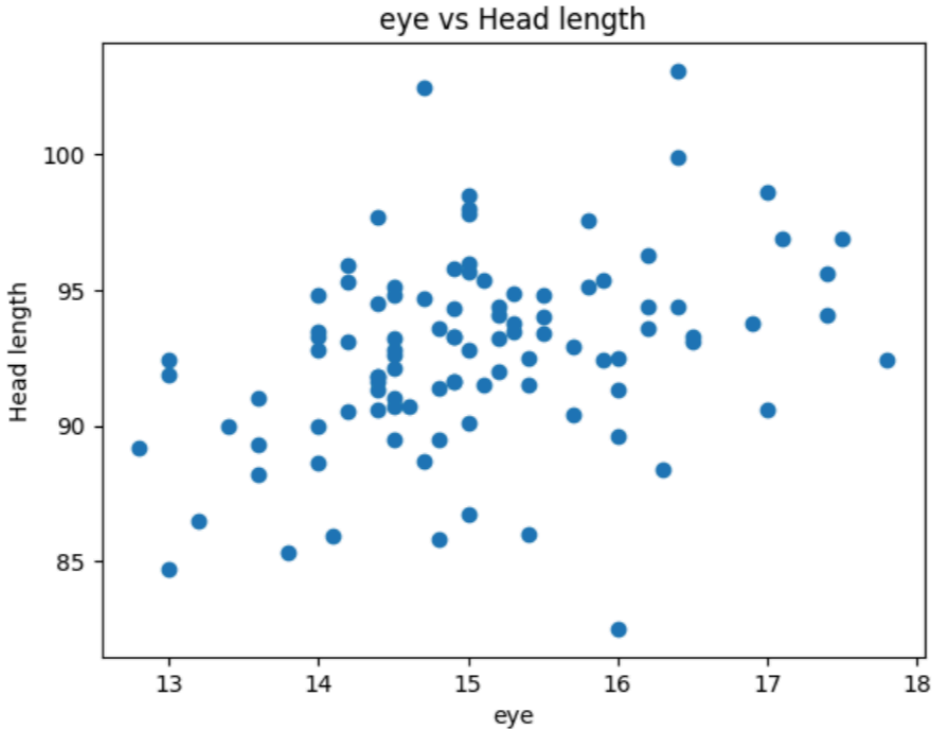
Data Cleaning & Feature Engineering

- Visualisasi heatmap



Dari korelasi antar variabel yang sebelumnya kita analisis, dapat dilihat ada banyak atribut yang saling berkorelasi dan yang paling tidak berkorelasi ada pada atribut earconch dan tail.

- membuat scatter plot 'eye' dan 'hdlngth'



Dari scatterplot kolom eye dan head length diatas, dapat dilihat bahwa semakin besar nilai eye maka nilai head length cenderung ikut naik.

3.3 Kernel Perceptron Modelling

3.3.1 Split Train dan Test Data

```
(80, 2) (80,)
(21, 2) (21,)
<ipython-input-13-01bfc70e889c>:5: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, u
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  X = np.asarray(X,dtype=np.float)
<ipython-input-13-01bfc70e889c>:6: DeprecationWarning: `np.float` is a deprecated alias for the builtin `float`. To silence this warning, u
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  y = np.asarray(y,dtype=np.float)
```

Hasil dari kode tersebut memiliki keterkaitan. Untuk (80, 2) dan (80,) untuk pasangan pertama, dan (21, 2) dan (21,) untuk pasangan kedua. Artinya ada dua kumpulan data, kumpulan pertama terdiri dari 80 sampel dengan masing-masing 2 fitur dan label yang sesuai, dan kumpulan kedua terdiri dari 21 sampel dengan masing-masing 2 fitur dan label yang sesuai.

3.3.2 Linear Kernel & Polynomial Kernel

- Fungsi kernel linier

  Fungsi kernel linier digunakan untuk mengklasifikasikan data yang linier. Fungsi ini dapat ditulis sebagai berikut:

  $K(x, y) = x \cdot y$

  Dimana x dan y adalah dua vektor. Nilai kernel linier akan bernilai 1 jika x dan y sejajar, 0 jika x dan y tegak lurus, dan nilai antara 0 dan 1 jika x dan y memiliki sudut tertentu.

- Fungsi kernel polynomial

  Fungsi kernel polinomial digunakan untuk mengklasifikasikan data yang nonlinier. Fungsi ini dapat ditulis sebagai berikut:

  $K(x, y) = (\text{gamma} \cdot x \cdot y + a) \text{ ^ degree}$

  Dimana x dan y adalah dua vektor, gamma adalah parameter yang menentukan kelengkungan permukaan keputusan, a adalah parameter yang menentukan posisi permukaan keputusan, dan degree adalah derajat polinomial. Nilai kernel polinomial akan bernilai 0 jika x dan y sejajar, dan nilai yang lebih besar dari 0 jika x dan y memiliki sudut tertentu. Nilai kernel polinomial akan semakin besar seiring dengan meningkatnya derajat polinomial.

3.3.3 Kernel Perceptron

```
iter:61,updates:35
iter:62,updates:35
iter:63,updates:35
iter:64,updates:35
iter:65,updates:35
iter:66,updates:35
iter:67,updates:35
iter:68,updates:35
iter:69,updates:35
iter:70,updates:35
iter:71,updates:35
iter:72,updates:35
iter:73,updates:35
iter:74,updates:35
iter:75,updates:35
iter:76,updates:35
iter:77,updates:35
iter:78,updates:35
iter:79,updates:35
iter:80,updates:35
iter:81,updates:35
iter:82,updates:35
iter:83,updates:35
iter:84,updates:35
iter:85,updates:35
iter:86,updates:35
iter:87,updates:35
iter:88,updates:35
iter:89,updates:35
iter:90,updates:35
iter:91,updates:35
iter:92,updates:35
iter:93,updates:35
iter:94,updates:35
iter:95,updates:35
iter:96,updates:35
iter:97,updates:35
iter:98,updates:35
iter:99,updates:35
iter:100,updates:35
test accuracy: 0.6666666666666666, train accuracy: 0.5625
```

Berdasarkan hasil diatas maka :
"iter:N, update:M": Menunjukkan nomor iterasi N dan jumlah pembaruan yang dilakukan dalam iterasi tersebut, yaitu M. "akurasi pengujian: X, akurasi pelatihan: Y": Ini menunjukkan akurasi

pengujian (seberapa baik performa model pada kumpulan data pengujian terpisah) dan akurasi pelatihan (seberapa baik performa model pada kumpulan data pelatihan) setelah menyelesaikan iterasi.

Proses pelatihan dilakukan sebanyak 100 iterasi. Dalam setiap iterasi, dilakukan 35 pembaruan. Akurasi pengujian yang dihasilkan setelah menyelesaikan 100 iterasi adalah 0,6666666666666666 (sekitar 66,67%), dan akurasi pelatihan adalah 0,5625 (sekitar 56,25%). Nilai akurasi ini menunjukkan bahwa model berperforma lebih baik pada set pengujian dibandingkan pada set pelatihan, karena akurasi pengujian lebih tinggi.

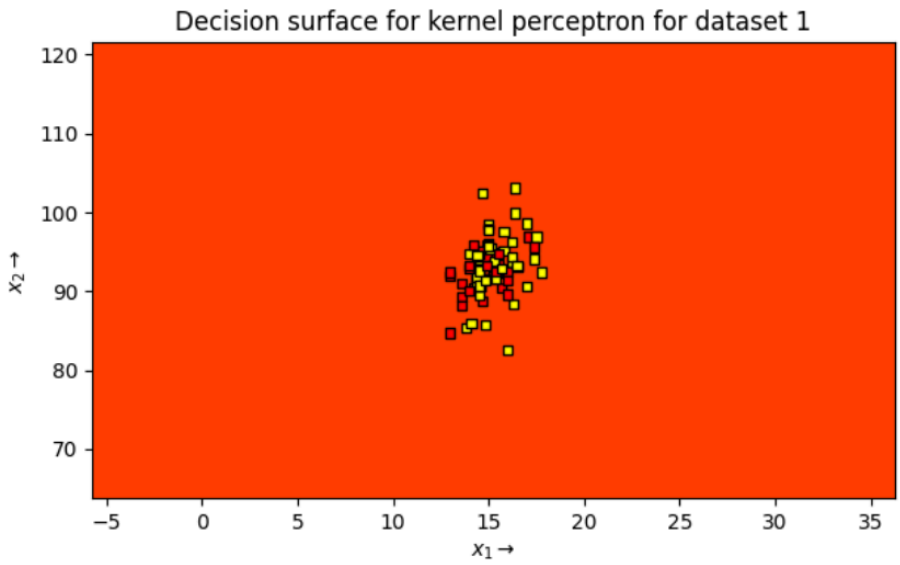
3.3.4 Linear SVM Classification and Model Analysis


```
Train Accuracy: 0.55
Test Accuracy: 0.5714285714285714
```

menguji keakuratan model Support Vector Machine (SVM) dengan kernel linier pada kumpulan data pelatihan dan pengujian. Akurasi dihitung dengan membandingkan label prediksi dari model dengan label sebenarnya di dataset. Dengan menginisialisasi model SVM dengan kernel linier dan parameter regularisasi besar C=1000. Kemudian menyesuaikan model dengan data pelatihan (X_train, y_train) dan menghitung keakuratan model pada kumpulan data pelatihan dan pengujian menggunakan metode skor. Kode tersebut menunjukkan bahwa model memprediksi kelas dengan tepat sebesar 55% data pelatihan dan 57,14% data pengujian.

Selain itu, kode ini menghitung vektor bobot yang dinormalisasi w1 dan nilai gamma menggunakan koefisien model SVM. Keakuratan model pada kumpulan data pengujian sedikit lebih tinggi dibandingkan pada kumpulan data pelatihan sehingga menunjukkan bahwa model tersebut mungkin tidak secara efektif menangkap pola dasar data.

3.3.5 Visualisasi Hasil



Gambar tersebut menampilkan model menggunakan perceptron kernel yang membagi menjadi 2 bagian. Perceptron kernel menggunakan fungsi kernel untuk mengubah ruang fitur input menjadi ruang dimensi yang lebih tinggi di mana data dapat dipisahkan secara linier. Bentuk spesifik dan orientasi permukaan keputusan akan bergantung pada parameter fungsi kernel dan bobot yang dipelajari oleh algoritma perceptron kernel. Gambar tersebut masih belum menunjukkan metode kernel dengan baik, mungkin karena ada beberapa kesalahan dalam kode.

4. Kesimpulan

Model Kernel Perception yang digunakan pada dataset *opp* menghasilkan nilai akurasi sebesar 0.6666666666666666 pada Train Accuracy dan 0.5625 pada Test Accuracy. Lalu pada model SVM Classification menghasilkan nilai akurasi sebesar 55% pada Train Accuracy dan 57,14%. Keakuratan model pada kumpulan data pengujian sedikit lebih tinggi dibandingkan pada kumpulan data pelatihan sehingga menunjukkan bahwa model tersebut mungkin tidak secara efektif menangkap pola dasar data. Perceptron kernel menggunakan fungsi kernel untuk mengubah ruang fitur input menjadi ruang dimensi yang lebih tinggi di mana data dapat dipisahkan secara linier sehingga dapat disimpulkan bahwa model Kernel Perception dan SVM Classification yang dikembangkan sangat baik dalam mengolah dataset *opp*, dengan tingkat akurasi yang tinggi. Model ini mampu mengenali pola dengan baik, dan dapat diandalkan untuk tugas menangkap pola dan relasi kompleks pada data baru.

Referensi

MODUL 5 PEMBELAJARAN MESIN

Hofmann, Thomas, Bernhard Schölkopf, and Alexander J. Smola. "Kernel methods in machine learning." (2008): 1171-1220.

Lampiran

Link code : PM_M5_KEL.ipynb
Link Presentasi Link Youtube