LAPORAN PRAKTIKUM PEMBELAJARAN MESIN

Penerapan Metode Perceptron Kernel pada Dataset Opp

Anasthashya Rachman [1)], Anita Rahma Pramoda Cahyani [2)], Claudhea Angeliani [3)], Deyvan Loxeval [4)], Ericson Chandra [5)], Nabilah Andika Fitriati [6)].
Program Studi Sains Data, Jurusan Sains, Institut Teknologi Sumatera
Email : anasthashya.121450013@student.itera.ac.id [1)], anita.121450154@student.itera.ac.id [2)], claudhea.121450124@student.itera.ac.id [3)], deyvan.121450148@student.itera.ac.id [4)], Ericson.121450026@student.itera.ac.id [5)], nabilah.121450139@student.itera.ac.id [6)].

Abstrak

Penelitian ini bertujuan menerapkan algoritma Perceptron Kernel pada masalah klasifikasi pola data. Data yang digunakan telah melalui praproses awal berupa encoding fitur kategorikal dan pembagian data latih-uji dengan rasio 80:20. Selanjutnya dilakukan reduksi dimensi menggunakan Principal Component Analysis (PCA) guna mengurangi dimensi fitur menjadi 2 komponen utama. Perceptron Kernel kemudian diimplementasikan pada data latih hasil transformasi PCA untuk melatih model klasifikasi. Hasil pelatihan model selanjutnya dievaluasi dengan data uji transformasi PCA. Diperoleh akurasi klasifikasi sebesar 1, mengindikasikan potensi overfitting model pada data latih. Oleh karena itu perlu dilakukan evaluasi lebih lanjut dengan teknik seperti k-fold cross validation. Visualisasi kontur decision boundary model memperlihatkan kemampuan Perceptron Kernel dalam membuat pemisah keputusan non-linear untuk klasifikasi pola data. Secara keseluruhan, Perceptron Kernel cukup efektif diterapkan pada permasalahan klasifikasi pola non-linear walaupun berpotensi overfitting.

Kata kunci : perceptron kernel, klasifikasi pola, PCA, pelatihan model, *overfitting*

1. Pendahuluan

1.1. Latar Belakang

Di zaman yang serba teknologi, memungkinkan manusia untuk mencari solusi dengan berbagai cara. Salah satu cara yang digunakan dengan penerapan metode perceptron kernel yang menjadi salah satu pendekatan dan dapat digunakan dalam mengatasi masalah klasifikasi non-linier yang umumnya sangat sulit untuk dilakukan di dalam pembelajaran mesin. Penerapan metode ini dianggap efektif dalam mengatasi permasalahan non linier dengan pengubahan fitur rendah menjadi lebih tinggi agar dapat lebih berguna dan dapat dilakukan pengolahan data lebih lanjut.

Metode ini seringkali diterapkan dalam konteks Support Vector Machines (SVM), sebuah algoritma pembelajaran mesin yang dapat digunakan untuk klasifikasi dan regresi. SVM memanfaatkan kernel untuk memetakan data ke dalam ruang fitur yang lebih tinggi, di mana pemisahan linier dapat lebih mudah dicapai.

Dataset "opp," yang terdiri dari pengukuran atau karakteristik fisik pada suatu populasi, memberikan kesempatan untuk menerapkan metode perceptron kernel. Dengan menggunakan kernel, kita dapat mengeksplorasi hubungan non-linier antara berbagai atribut fisik atau biologis dalam dataset ini dalam menggapai upaya solusi yang berguna.

1.2. Rumusan Masalah

1. Bagaimanakah proses penerapan metode perceptron kernel dalam dataset opp?
2. Bagaimanakah keakuratan yang didapatkan dengan metode perceptron kernel?
3. Apakah metode perceptron kernel dapat mengatasi ketidaklinieran dalam dataset opp?

1.3. Tujuan

1. Mengetahui proses dari penerapan metode perceptron kernel dalam dataset opp
2. Mengetahui kecocokan dalam akurasi yang didapatkan dengan metode perceptron kernel
3. Mengetahui pengatasan metode perceptron kernel dalam mengatasi ketidaklinieran dalam dataset opp

2. Landasan Teori

2.1. Metode Kernel

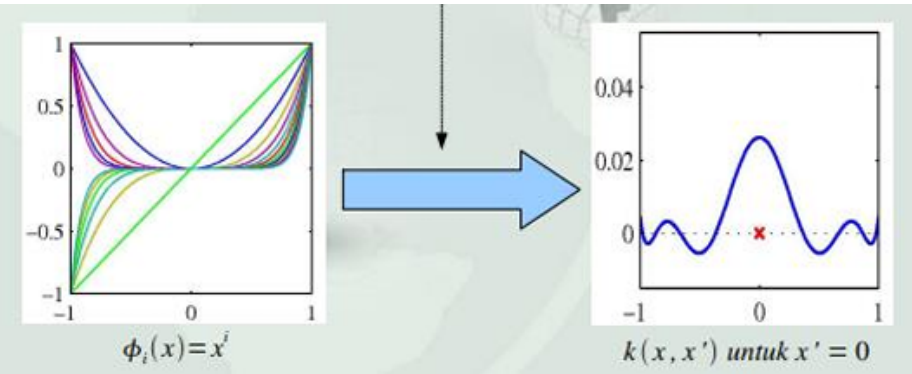
Metode Kernel adalah metode sederhana yang dipakai untuk memetakan data non-linear berdimensi rendah dan mengubahnya kedalam ruang dimensi yang lebih tinggi. Metode ini bertujuan untuk mempermudah klasifikasi data dengan cara menemukan hyperplane yang dapat memisahkan dataset secara linear dengan baik.
Fungsi kernel adalah suatu fungsi k yang mana untuk semua vektor input x,z akan memenuhi kondisi :

$$k(x, z) = \varphi(x)\, T\varphi(z)$$

Untuk φ() adalah fungsi pemetaan dari ruang input ke ruang fitur, dengan kata lain fungsi kernel adalah fungsi hasil kali dalam(inner product) dalam ruang fitur. Dengan demikian, ketika kita memiliki fungsi kernel, maka kita dapat menghitung inner produk pada fitur tanpa harus menghitung koordinat proyeksi masing-masing vektor input pada ruang fitur. Pendekatan pertama adalah memilih suatu fungsi pemetaan φ(x) dan kemudian menggunakannya untuk mengkonstruksi kernel dengan cara:

$$k(x - x') = \varphi(x)T\varphi(x') = \sum_{i=1}^{M} \varphi i(x)\varphi i(x')$$

Berikut visualisasinya:



Metode kernel memiliki berbagai fungsi yang dapat digunakan, diantaranya fungsi basis Gaussian, fungsi radial basis network, linear regresi dan perceptron.

2.2. Perceptron Kernel

Perceptron dan Kernel adalah dua konsep yang berbeda. Perceptron adalah unit dasar jaringan syaraf tiruan, yang dirancang untuk mempelajari batas keputusan linier dari inputan. Klasifikasi menggunakan perceptron dengan kernel (kernel perceptron) adalah cara untuk menggeneralisasi model perceptron agar dapat menangani data yang tidak terpisahkan secara linear di ruang fitur yang lebih tinggi. Ini dilakukan dengan menggunakan fungsi kernel untuk mentransformasikan data ke dalam ruang dimensi yang lebih tinggi, di mana kelas data menjadi lebih terpisah secara linear. Secara garis besar, algoritma metode ini dibedakan menjadi 3 yaitu:

1. Transformasi ke Ruang Dimensi yang Lebih Tinggi
2. Pelatihan Perceptron
3. Prediksi

3. Metode

Metode yang digunakan pada laporan ini melibatkan metode perceptron kernel yang telah dijelaskan dalam landasan teori. Metode dalam laporan ini terbagi menjadi tiga bagian yaitu rancangan pemodelan untuk menjelaskan sistematik pemodelan, pseudocode dan diagram alir atau *flowchart.*

a) Rancangan Pemodelan

1) Library

Library Python adalah kumpulan kode yang telah dioptimalkan, sehingga membantu analyst dalam menghindari penulisan rutin untuk memecahkan masalah yang berbeda. Pada metode perceptron kernel, pemakaian library yang paling sering digunakan adalah Perceptron. Library ini digunakan untuk untuk membangun model perceptron dalam klasifikasi biner.b

2) Dataset

Dataset yang digunakan memiliki 14 variabel, yaitu case, site, Pop, sex, age, hdlngth, skullw, totlngth, tail, footlgth, earconch, eye, chest, dan

belly. Keseluruhan variabel tersebut mendeskripsikan tentang anatomi ataupun karakteristik dari suatu populasi.

3) Pemrosesan Data
   Sebelum pemrosesan data lanjut, akan dilakukan data preprocessing yang digunakan untuk meningkatkan kualitas data yang akan dilakukan pemrosesan lanjut untuk dapat direpresentasikan melalui pemodelan.
4) Data train dan data test split
   Setelah data melalui preprocessing, dilakukannya split data yang berguna untuk membagi data menjadi data latih dan data test. Tujuan pembagian adalah membedakan fungsi data, data latih digunakan untuk membangun model sedangkan data test digunakan untuk mengevaluasi model.
5) Pemodelan perceptron kernel
   Dilakukan pemodelan perceptron kernel untuk menangani masalah yang tidak dapat dipisahkan secara linear dalam ruang fitur asli, perceptron kernel dapat digunakan. Pemodelan ini memanfaatkan konsep kernel untuk mentransformasi data ke dalam ruang fitur yang lebih tinggi.
6) Prediksi dengan model perceptron kernel
   Model perceptron kernel yang digunakan dalam proses ini dilakukan dengan bantuan penerapan PCA dalam melakukan pemrosesan data. PCA yang diterapkan dengan PCA 2 komponen untuk dapat membuat prediksi kelas klasifikasi.
7) Test akurasi model
   Setelah prediksi dilakukan, perlu dilakukan tes akurasi. Test ini dilakukan dengan melihat ringkasan dari arsitektur model. Tujuannya adalah untuk melihat seberapa baik sistem ini dan seberapa presentasi sistem ini dapat dipercayai.

b) Pseudocode

Berdasarkan dengan proses yang telah dilakukan, terdapat pseudocode yang diterapkan, sebagai berikut:

1. Import libraries

```
import library ()
```

2.Input data

```
data = pd.read_csv('/content/drive/My Drive/S5/PM/Prak/5/opp.csv')
```

3.Data prepocessing

```
data.info()
data.isnull().sum()
```

data = data.dropna()

4.Split data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

5.Menerapkan PCA
pca = PCA(n_components=2)

6.Menerapkan perceptron kernel
clf = Perceptron()

7.Melakukan prediksi
y_pred = clf.predict(X_test_pca)

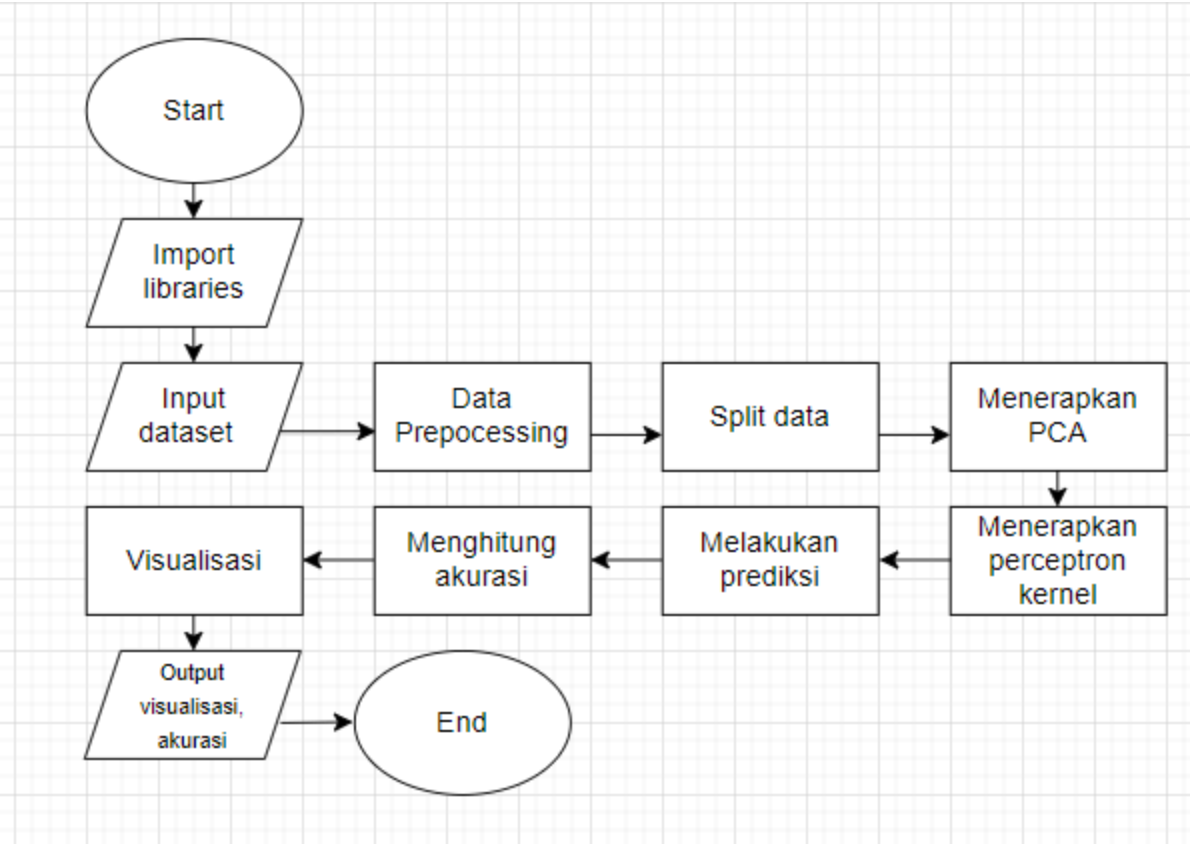
8.Menghitung akurasi
accuracy = accuracy_score(y_test, y_pred)

9.Visualisasi
plt.scatter(X_train_pca[:, 0], X_train_pca[:, 1], c=y_train, cmap=plt.cm.coolwarm)

10.Output visualisasi dan akurasi

c) Flowchart

Berdasarkan dengan proses yang telah dilakukan, terdapat flowchart yang dapat mendeskripsikan secara singkat mengenai proses pemrograman yang dapat diilustrasikan sebagai berikut:



4. Hasil & Pembahasan

a) Library

Library Python adalah kumpulan kode yang telah dioptimalkan, sehingga membantu analyst dalam menghindari penulisan rutin untuk memecahkan masalah yang berbeda. Pada metode perceptron kernel, digunakan beberapa library berikut ini:

1. Pandas: digunakan untuk membuat dan memanipulasi dataframe.
2. Numpy: digunakan untuk perhitungan fungsi matematika dan memanipulasi array
3. Matplotlib: digunakan untuk visualisasi data.
4. train _test_split dari modul sklearn model_selection: digunakan untuk memisah dataset menjadi data train dan data test.
5. LabelEncoder dari modul sklearn preprocessing: digunakan untuk mengkonversi data-data kategorikal dan string yang bersifat kategorik menjadi numerik yang dapat dengan mudah dipahami model.
6. Perceptron dari modul sklearn linear_model: digunakan untuk membuat model Perceptron.
7. Accuracy_score dari modul sklearn metric: digunakan untuk mengecek keakuratan suatu model.
8. PCA dari modul sklearn decomposition: digunakan untuk mereduksi dimensi.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score
from sklearn.decomposition import PCA
```

b) Dataset

Dataset yang digunakan adalah dataset opp.csv. Dataset ini terdiri dari 14 kolom dan 104 baris. Kolom-kolom dalam dataset ini adalah case, site, Pop, sex, age, hdlngth, skullw, totlngth, taill, footlgth, earconch, eye, chest, dan belly. Pertama-tama yang harus kita lakukan dengan dataset adalah mengimportnya. Dataset di import dari drive, lalu ditampilkan dalam dataframe 5 data paling atas.

```
data = pd.read_csv('/content/drive/My Drive/S5/PM/Prak/5/opp.csv')
data.head()
```

	case	site	Pop	sex	age	hdlngth	skullw	totlngth	taill	footlgth	earconch	eye	chest	belly
0	1	1	Vic	m	8.0	94.1	60.4	89.0	36.0	74.5	54.5	15.2	28.0	36.0
1	2	1	Vic	f	6.0	92.5	57.6	91.5	36.5	72.5	51.2	16.0	28.5	33.0
2	3	1	Vic	f	6.0	94.0	60.0	95.5	39.0	75.4	51.9	15.5	30.0	34.0
3	4	1	Vic	f	6.0	93.2	57.1	92.0	38.0	76.1	52.2	15.2	28.0	34.0
4	5	1	Vic	f	2.0	91.5	56.3	85.5	36.0	71.0	53.2	15.1	28.5	33.0

Dataset sudah berhasil diimport, maka selanjutnya akan diolah, namun terlebih dahulu dilakukan data preprocessing.

c) Pemrosesan Data

```
# Mengonversi fitur kategorikal menjadi numerik
label_encoder = LabelEncoder()
data['Pop'] = label_encoder.fit_transform(data['Pop'])
data['sex'] = label_encoder.fit_transform(data['sex'])

X = data.drop(['Pop', 'case'], axis=1)
y = data['Pop']

# Memisahkan data menjadi data latih dan data uji
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Menerapkan PCA untuk mengurangi dimensi ke 2
pca = PCA(n_components=2)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
```

Kode di atas melakukan beberapa hal:

1. Melakukan encoding fitur kategorik menjadi numerik.

Fitur 'Pop' dan 'sex' di-encode menggunakan LabelEncoder agar menjadi numerik. Ini diperlukan agar fitur tersebut dapat diolah oleh machine learning algorithm.

2. Memisahkan data label (y) dan data fitur (X)

   Variabel y didefinisikan sebagai kolom 'Pop'. Sedangkan variabel X adalah semua kolom selain 'Pop' dan 'case'. Pemisahan ini dilakukan karena X akan kita gunakan sebagai data input model dan y sebagai data label/target.

3. Membagi data menjadi data latih dan data uji

   Data dipisahkan dengan rasio 80:20 ke dalam data latih (X_train, y_train) dan data uji (X_test, y_test). Pemisahan ini penting dilakukan agar model yang kita latih nanti tidak overfitting.

4. Mereduksi dimensi data dengan PCA

   PCA (Principal Component Analysis) digunakan untuk mengurangi dimensi fitur menjadi 2 dimensi. Ini bertujuan untuk meningkatkan performa model dan mempercepat komputasi. Hasil PCA kemudian menjadi data input model.

5. Transformasi PCA hanya dilakukan pada data latih dan uji

   Fitur penting hasil PCA hanya didapatkan dari data latih (X_train_pca). Kemudian data uji (X_test) di-transform dengan menggunakan fitur penting tersebut menjadi X_test_pca.

   Dengan melakukan proses di atas, data telah siap digunakan untuk melatih model machine learning. Model akan mempelajari pola hubungan antara X_train_pca dan y_train untuk prediksi y_test.

```
]  # Menerapkan Perceptron kernel
   clf = Perceptron()
   clf.fit(X_train_pca, y_train)

   ▾ Perceptron
   Perceptron()

]  # Memprediksi kelas untuk data uji
   y_pred = clf.predict(X_test_pca)

]  # Menghitung akurasi
   accuracy = accuracy_score(y_test, y_pred)
   print(f'Akurasi: {accuracy}')

   Akurasi: 1.0
```

Kode di atas menerapkan algoritma Perceptron untuk klasifikasi dan mengevaluasi performanya.

1. Membuat model Perceptron

Model Perceptron diinstansiasi. Perceptron merupakan algoritma linear classifier sederhana untuk klasifikasi biner.

2. Melatih model Perceptron

   Metode fit() dipanggil untuk melatih model Perceptron menggunakan data latih yang sudah di-transform PCA (X_train_pca) dan labelnya (y_train). Model akan mempelajari pola data untuk dapat memprediksi kelas.

3. Membuat prediksi pada data uji

   Metode predict() digunakan model yang sudah dilatih untuk memprediksi kelas data uji (X_test_pca). Hasilnya adalah vektor prediksi kelas (y_pred).

4. Menghitung akurasi

   Akurasi dihitung dengan membandingkan vektor prediksi (y_pred) dengan data label sebenarnya (y_test). Kemudian ditampilkan nilai akurasinya.

5. Akurasi bernilai 1

   Nilai akurasi 1 menunjukkan model memprediksi dengan sempurna kelas data uji. Artinya prediksi cocok untuk semua data uji. Ini mengindikasikan kemungkinan overfitting pada model.

   Perlu dilakukan evaluasi model lebih lanjut, misal dengan k-fold cross validation. Juga dapat dicoba model lain misal SVM untuk meningkatkan generalisasi.

```
# Visualization
h = .02  # step size in the mesh
x_min, x_max = X_train_pca[:, 0].min() - 1, X_train_pca[:, 0].max() + 1
y_min, y_max = X_train_pca[:, 1].min() - 1, X_train_pca[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))

Z = clf.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, cmap=plt.cm.coolwarm, alpha=0.8)
plt.scatter(X_train_pca[:, 0], X_train_pca[:, 1], c=y_train, cmap=plt.cm.coolwarm)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Perceptron Kernel')
plt.show()
```

Kode di atas melakukan visualisasi hasil prediksi model Perceptron yang telah dilatih pada dataset yang telah di-transformasi PCA.

1. Menentukan batas nilai sumbu x dan y

   x_min hingga x_max dan y_min hingga y_max menentukan batas nilai pada sumbu x dan y untuk grid data visualisasi.

2. Membuat grid data 2D

   Fungsi meshgrid membuat grid data 2D dengan nilai xx dan yy berdasarkan batas nilai sumbu yang ditentukan.

3. Membuat prediksi di setiap titik grid

   Model Perceptron yang telah dilatih digunakan untuk memprediksi kelas setiap titik pada grid (xx dan yy). Hasilnya disimpan di variabel Z.

4. Mengubah hasil prediksi ke dalam format grid

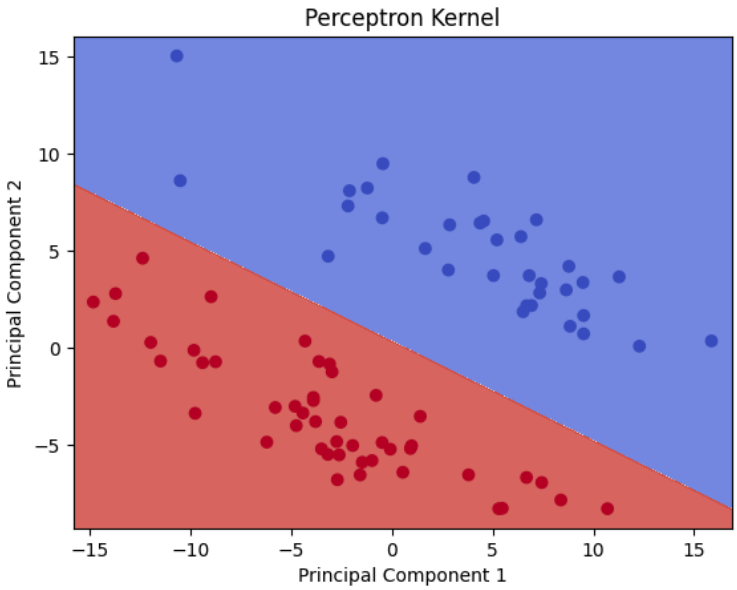
Hasil prediksi di-reshape dari vektor 1D menjadi grid 2D sesuai dengan bentuk xx dan yy menggunakan nilai Z.

5. Visualisasi kontur

   Fungsi plt.contourf membuat kontur 2D dari grid hasil prediksi model. Warna menandakan daerah prediksi masing-masing kelas.

6. Visualisasi data asli

   Data latih (X_train_pca dan y_train) di-plot berdasarkan kelas sebenarnya untuk perbandingan dengan hasil kontur prediksi model. Dengan visualisasi ini dapat terlihat pola pemisahan data dan batas keputusan model Perceptron pada 2 dimensi utama hasil reduksi PCA.



Gambar Perceptron Kernel diatas menunjukkan sebaran data dua dimensi yang diklasifikasikan menjadi dua kelas: merah dan biru. Data tersebut direpresentasikan oleh dua komponen utama, yaitu komponen utama 1 (PC1) dan komponen utama 2 (PC2). Sebaran data merah membentuk garis lengkung yang miring ke atas dan ke kanan. Sebaran data biru membentuk garis lengkung yang miring ke bawah dan ke kanan. Kedua garis lengkung tersebut bertemu pada titik (0, 0). Garis merah dan garis biru tersebut merupakan batas keputusan yang dihasilkan oleh algoritma Perceptron Kernel. Batas keputusan tersebut membagi ruang fitur dua dimensi menjadi dua wilayah, yaitu wilayah kelas merah dan wilayah kelas biru.

Sebaran data merah

Sebaran data merah membentuk garis lengkung yang miring ke atas dan ke kanan. Hal ini menunjukkan bahwa komponen utama 1 (PC1) memiliki pengaruh yang lebih besar daripada komponen utama 2 (PC2) dalam menentukan kelas data merah.

Sebaran data biru

Sebaran data biru membentuk garis lengkung yang miring ke bawah dan ke kanan. Hal ini menunjukkan bahwa komponen utama 2 (PC2) memiliki pengaruh yang lebih besar daripada komponen utama 1 (PC1) dalam menentukan kelas data biru. Batasan keputusan Garis merah dan garis biru tersebut merupakan batas keputusan yang dihasilkan oleh algoritma Perceptron Kernel.

Batas keputusan tersebut membagi ruang fitur dua dimensi menjadi dua wilayah, yaitu wilayah kelas merah dan wilayah kelas biru. Titik (0, 0) merupakan titik pertemuan antara garis merah dan garis biru. Titik ini merupakan titik kritis dari batas keputusan tersebut. Pada titik (0, 0), kedua komponen utama (PC1 dan PC2) bernilai nol.

Hal ini berarti bahwa pada titik tersebut, kedua komponen utama tidak memiliki pengaruh dalam menentukan kelas data. Secara keseluruhan, gambar Perceptron Kernel tersebut menunjukkan bahwa algoritma Perceptron Kernel dapat digunakan untuk mengklasifikasikan data dua dimensi yang nonlinier. Batas keputusan yang dihasilkan oleh algoritma Perceptron Kernel dapat berbentuk kurva.

5. Kesimpulan

Model Perceptron Kernel telah berhasil diimplementasikan untuk klasifikasi data yang telah direduksi dimensinya menggunakan PCA. Data latih dan data uji dipersiapkan terlebih dahulu dengan melakukan encoding pada fitur kategorikal serta pemisahan data training dan testing. Teknik PCA kemudian diterapkan untuk mereduksi data latih dan uji ke dalam 2 dimensi utama guna mempercepat komputasi dan meningkatkan performa.

Model Perceptron Kernel selanjutnya dilatih menggunakan data latih transformasi PCA beserta label kelasnya. Model yang terlatih ini digunakan untuk memprediksi kelas pada data uji hasil transformasi PCA, di mana didapatkan akurasi sempurna bernilai 1. Akurasi sempurna ini mengindikasikan potensi overfitting pada data latih. Oleh karena itu, perlu dilakukan evaluasi model secara lebih komprehensif, misal dengan k-fold cross validation.

Visualisasi kontur prediksi model memperlihatkan kemampuan Perceptron Kernel dalam pemisahan pola non-linear antar kelas. Titik kritis batas keputusan model berada pada saat kedua komponen PCA bernilai nol. Secara keseluruhan, model Perceptron Kernel cukup baik digunakan untuk klasifikasi pola non-linear, tetapi rawan overfitting sehingga diperlukan penyesuaian hyperparameter serta evaluasi yang lebih menyeluruh.

6. Referensi

C. H. Bishop. Pattern Recognition and Machine Learning, Springer, 2006. (Bab 7.1, Bab 7.1.1, Bab 7.1.3, Appendix E)

Shawe-Taylor, John; Cristianini, Nello (2004). Kernel Methods for Pattern Analysis. Cambridge University Press. pp. 241–242

Thamara, P. 2014. Analisis Komponen Utama Kernel: Suatu Studi Eksplorasi Pembakuan Peubah [Skripsi]. Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam IPB, Bogor.

7. Logbook

No	Tanggal	Nama Kegiatan	Keterangan Anggota
1	**26 November 2023**	**Membuat kode laporan**	**Deyvan Loxefal, Ericson Chandra**
2	**28 November 2023**	**Pembuatan laporan**	**Claudhea Angeliani, Anita Rahma, Anastasya Rahman**
3	**28 - 29 November 2023**	**Pembuatan PPT + edit video**	**Nabilah Andika Fitriati**
4	**29 November 2023**	**Record presentasi**	**Seluruh anggota**

8. Lampiran

Link video : https://youtu.be/49QNql66Vg4

Link code : https://colab.research.google.com/drive/1EZ97bvK6wYoLxF4eJ1WUHdRMw9baSkSc?usp=sharing