**Pokok Bahasan XII**

**Support Vector Regression**

**Kode Pokok Bahasan**: TIK.RPL03.001.009.01

**Deskripsi Pokok Bahasan**:

Membahas bagaimana melakukan proses penentuan keputusan prediksi dengan Support Vector Regression (SVR) .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Elemen Kompetensi | Indikator Kinerja | Jml Jam | Hal |
| 1 | Mengimplementasikan SVR dalam kasus yang diberikan | 1.1 Menerapkan SVR untuk data beras yang tersimpan dalam database.  1.2 Membandingkan error metode SVR dan MLP | 1 | 5 |

**TUGAS PENDAHULUAN**

Hal yang harus dilakukan dan acuan yang harus dibaca sebelum praktikum :

1. Menginstal R pada PC masing-masing praktikan.

2. Menginstal R Studio pada PC masing-masing praktikan.

**DAFTAR PERTANYAAN**

1. Apa yang dimaksud dengan Support Vector Regression (SVR)?

Support Vector Regression (SVR) adalah salah satu metode yang bisa digunakan dalam melakukan peramalan. Data yang digunakan adalah data penjualan roti manis, cake dan tawar dengan tipe data time series dan menggunakan 4 fitur.

2. Apa manfaat menggunakan Support Vector Regression (SVR)?

SVR memiliki kemampuan untuk mengatasi permasalahan overfitting, sehingga bisa mendapatkan suatu fungsi dengan tingkat kesalahan yang kecil dan menghasilkan prediksi yang bagus dengan cara menemukan Hyperplane (garis pemisah) terbaik.

3. Sebutkan kelebihan Support Vector Regression (SVR) dalam memprediksi data?

Dapat menggunakan beberapa classifier yang dilatih pada berbagai jenis data menggunakan aturan probabilitas. SVR melakukan komputasi yang lebih rendah dibandingkan dengan teknik regresi lainnya. Memiliki kemampuan generalisasi yang sangat baik, dengan akurasi prediksi yang tinggi.

**TEORI SINGKAT**

Algoritma SVR (Support Vector Regression) adalah salah satu algoritma yang digunakan untuk pengambilan keputusan. Contoh yang dibahas kali ini adalah menentukan hasil jual tipe sepeda motor baru berdasarkan kelompok data yang sudah ada.

Algoritma ini dapat dikatakan sebagai perbaikan dari Regresi Linier / Analisis Regresi. Jika regresi linier menghasilkan sebuah fungsi dengan hasil linier / garis lurus, maka algoritma ini dapat menghasilkan sebuah fungsi dengan hasil yang bergelombang mengikuti jalur data yang terbentuk, sehingga prediksi yang didapatkan menjadi lebih akurat dibandingkan dengan regresi linier.

**LAB SETUP**

Hal yang harus disiapkan dan dilakukan oleh praktikan untuk menjalankan praktikum modul ini.

1. Menginstal library yang dibutuhkan untuk mengerjakan modul.

2. Menjalankan R Studio.

3. XAMPP yang sudah dijalankan.

4. Membuat database pada phpmyadmin.

**ELEMEN KOMPETENSI I**

**Deskripsi:**

Mengimplementasikan SVR dalam kasus yang diberikan.

**Kompetensi Dasar**:

1 Menerapkan SVR untuk data beras yang tersimpan dalam database.

2 Membandingkan error metode SVR dan MLP.

**Latihan 1.1.1**

**Penjelasan Singkat :**

Pada latihan ini anda akan diminta untuk melakukan analisis regresi dengan SVR terhadap data beras yang diakses melalui database.

**Langkah Praktikum :**

**1. Buatlah sebuah database pada phpmyadmin untuk menyimpan data beras.**

**2. Import data beras yang akan tersimpan dalam bentuk beberapa tabel.**

**3. Panggil library yang dibutuhkan**

library(RMySQL)

library(nnfor)

library(dplyr)

library(hydroGOF)

library(e1071)

**4. Lakukan koneksi ke database**

con = dbConnect(MySQL(), user = 'root', password = '', dbname = 'db\_pibc\_olap', host = 'localhost')

dbListTables(con)

**5. Lakukan query “select” untuk mengambil data yang terpilih.**

myQuery <- "select \* from fact\_harga;"

**6. Simpan data ke dalam dataframe.**

df <- dbGetQuery(con, myQuery)

df1<-filter(df,SK\_RICE\_TYPE==9, SK\_DATE>=20180101, SK\_DATE<=20181231, SK\_MARKET==0)

View(df1)

dim(df1)

**7. Ubah dataframe menjadi time series.**

myts=ts(df1$PRICE)

**8. Lakukan prediksi SVM sebagai berikut:**

X=c(1:243) #sesuaikan jumlah observasinya

#Regression with SVM

modelsvm = svm(df1$PRICE~X,data=df1)

#Predict using SVM regression

predYsvm = predict(modelsvm, data=df1)

#Overlay SVM Predictions on Scatter Plot

plot(X, predYsvm, col = "red", pch=16)

points(X,df1$PRICE, col="blue", pch=16)

**9. Hitung error RMSE pada model, dengan menggunakan fungsi rmse**

#Calculate RMSE

RMSEsvm=rmse(predYsvm,df1$PRICE)

RMSEsvm

#Overlay SVM Predictions on Scatter Plot

X=c(1:273)

**10. Membuat plot pada data model fit**

fit2 <- mlp(myts)

plot(fit2, col = "red", pch=16)

points(df1$PRICE, col="blue", pch=16)

Output :

|  |
| --- |
| Chart, line chart, histogram  Description automatically generated  Diagram, schematic  Description automatically generated |

Penjelasan :

|  |
| --- |
| Plot SVM memuat harga asli dan prediksi dari data Fact\_Harga, yang bergaris merah menunjukan data prediksi, yang bergaris biru menunjukan data asli. MLP mendapatkan data input kemudian diolah di Hidden kemudian jadi output yang diinginkan |

**Latihan 1.1.2**

**Penjelasan Singkat :**

Pada latihan ini anda akan diminta untuk melakukan peramalan dengan metode MLP terhadap data beras yang diakses melalui database. Lalu dibandingkan dengan metode SVR.

**Tugas:**

Gunakan data pada database “db\_pasokan\_beras”. Kemudian Lakukan Langkah Langkah seperti di Latihan 1.1.1. kemudian bandingkan errornya antara MLP dan SVM dengan metode SVR. Manakah hasilnya yang lebih akurat.

Catatan : Semua output dalam bentuk screenshot dimana nama object model peramalannya ditulis sesuai nama praktikan (contoh modelsvm diganti dengan svm\_namapraktikan). Berikan penjelasan atas output yang didapatkan.

|  |
| --- |
| #Latihan 1.1.2  library(statsr)  library(RMySQL)  library(dplyr)  con = dbConnect(MySQL(), user = 'root', password = '', dbname = 'dbpasokanberas', host = 'localhost')  dbListTables(con)  myQuery <- "select \* from fact\_price;"  df <- dbGetQuery(con, myQuery)  df1<-filter(df,id\_rice\_type==10, id\_date>=20170101,id\_date<=20171231)  df2<- df1[order(df1$id\_date),]  View(df1)  myts=ts(df1$harga)  myts  X=c(1:365) #sesuaikan jumlah observasinya  #Regression with SVM  modelsvm = svm(df1$harga~X,data=df1)  modelsvm  #Predict using SVM regression  predYsvm = predict(modelsvm, data=df1)  #Overlay SVM Predictions on Scatter Plot  plot(X, predYsvm, col = "red", pch=16)  points(X,df1$harga, col="blue", pch=16)  #Calculate RMSE  RMSEsvm=rmse(predYsvm,df1$harga)  RMSEsvm  #Overlay SVM Predictions on Scatter Plot  X=c(1:273)  fit2 <- mlp(myts)  plot(fit2, col = "red", pch=16)  points(df1$harga, col="blue", pch=16) |

Output

|  |
| --- |
| Chart, line chart  Description automatically generated  Chart, line chart  Description automatically generated  Diagram, schematic  Description automatically generated |

**Penjelasan**

|  |
| --- |
| Plot SVM memuat harga asli dan prediksi dari data Pasokan Beras, yang bergaris merah menunjukan data prediksi, yang bergaris biru menunjukan data asli. MLP mendapatkan data input kemudian diolah di Hidden kemudian jadi output yang diinginkan |

**Latihan 1.2.1**

**Penjelasan Singkat :**

Pada latihan ini akan diminta menggunakan SVR dengan python.

**Langkah Praktikum :**

1. Import libraries

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import matplotlib.pyplot as plt # for data visualization

import seaborn as sns # for statistical data visualization

%matplotlib inline

import warnings

warnings.filterwarnings('ignore')

1. Import dataset

df = pd.read\_csv("fact\_harga.csv")

1. Explorasi data

# melihat dimensi dataset

df.shape

|  |
| --- |
| output: |

# melihat dataset

df.head()

|  |
| --- |
| output: |

# melihat nama kolom pada dataframe

col\_names = df.columns

col\_names

|  |
| --- |
| output: |

# melihat summary dari dataset

df.info()

|  |
| --- |
| output: |

# cek adanya missing value

df.isnull().sum()

|  |
| --- |
| output: |

# melihat statistik summary pada variabel numerik

round(df.describe(),2)

|  |
| --- |
| output: |

1. Visualisasi dengan boxplot dan histogram

# menggambarkan boxplots untuk visualisasi outlier

plt.figure(figsize=(24,20))

plt.subplot(4, 2, 3)

fig = df.boxplot(column='SK\_RICE\_TYPE')

fig.set\_title('')

fig.set\_ylabel('SK\_RICE\_TYPE')

plt.subplot(4, 2, 4)

fig = df.boxplot(column='SK\_MARKET')

fig.set\_title('')

fig.set\_ylabel('SK\_MARKET')

plt.subplot(4, 2, 5)

fig = df.boxplot(column='PRICE')

fig.set\_title('')

fig.set\_ylabel('PRICE')

|  |
| --- |
| output: |

# plot histogram untuk mengecek distribusi

plt.figure(figsize=(24,20))

plt.subplot(4, 2, 3)

fig = df['SK\_RICE\_TYPE'].hist(bins=20)

fig.set\_xlabel('SK\_RICE\_TYPE')

fig.set\_ylabel('Number of ...')

plt.subplot(4, 2, 4)

fig = df['SK\_MARKET'].hist(bins=20)

fig.set\_xlabel('SK\_MARKET')

fig.set\_ylabel('Number of ...')

plt.subplot(4, 2, 5)

fig = df['PRICE'].hist(bins=20)

fig.set\_xlabel('PRICE')

fig.set\_ylabel('Number of ...')

|  |
| --- |
| output: |

1. Split data

X = df.drop(['PRICE'], axis=1)

y = df['PRICE']

# split X and y into training and testing sets

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# check the shape of X\_train and X\_test

X\_train.shape, X\_test.shape

|  |
| --- |
| output: |

1. Feature Scaling

cols = X\_train.columns

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

X\_train = pd.DataFrame(X\_train, columns=[cols])

X\_test = pd.DataFrame(X\_test, columns=[cols])

X\_train.describe()

|  |
| --- |
| output: |

1. Menjalankan SVM dengan default hyperparameters

SVM dengan C = 1.0

# import SVC classifier

from sklearn.svm import SVC

# import metrics to compute accuracy

from sklearn.metrics import accuracy\_score

# instantiate classifier with default hyperparameters

svc=SVC()

# fit classifier to training set

svc.fit(X\_train,y\_train)

# make predictions on test set

y\_pred=svc.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with default hyperparameters: {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

SVM dengan C = 1000.0

# instantiate classifier with rbf kernel and C=1000

svc=SVC(C=1000.0)

# fit classifier to training set

svc.fit(X\_train,y\_train)

# make predictions on test set

y\_pred=svc.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with rbf kernel and C=1000.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

**Latihan 1.2.2**

**Penjelasan singkat :**

Pada latihan ini akan diminta menjalankan SVM dengan polynomial kernel, sigmoid kernel, dan linear kernel menggunakan python.

1. Menjalankan SVM dengan polynomial kernel

SVM dengan C = 1.0

# instantiate classifier with polynomial kernel and C=1.0

poly\_svc=SVC(kernel='poly', C=1.0)

# fit classifier to training set

poly\_svc.fit(X\_train,y\_train)

# make predictions on test set

y\_pred=poly\_svc.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with polynomial kernel and C=1.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

SVM dengan C = 100.0

# instantiate classifier with polynomial kernel and C=100.0

poly\_svc100=SVC(kernel='poly', C=100.0)

# fit classifier to training set

poly\_svc100.fit(X\_train, y\_train)

# make predictions on test set

y\_pred=poly\_svc100.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with polynomial kernel and C=1.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

1. Menjalankan SVM dengan sigmoid kernel

SVM dengan C = 1.0

# instantiate classifier with sigmoid kernel and C=1.0

sigmoid\_svc=SVC(kernel='sigmoid', C=1.0)

# fit classifier to training set

sigmoid\_svc.fit(X\_train,y\_train)

# make predictions on test set

y\_pred=sigmoid\_svc.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with sigmoid kernel and C=1.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

SVM dengan C = 100.0

# instantiate classifier with sigmoid kernel and C=100.0

sigmoid\_svc100=SVC(kernel='sigmoid', C=100.0)

# fit classifier to training set

sigmoid\_svc100.fit(X\_train,y\_train)

# make predictions on test set

y\_pred=sigmoid\_svc100.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with sigmoid kernel and C=100.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

1. Menjalankan SVM dengan Linear Kernel

SVM dengan C = 1.0

# instantiate classifier with linear kernel and C=1.0

linear\_svc=SVC(kernel='linear', C=1.0)

# fit classifier to training set

linear\_svc.fit(X\_train,y\_train)

# make predictions on test set

y\_pred\_test=linear\_svc.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with linear kernel and C=1.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred\_test)))

|  |
| --- |
| output: |

SVM dengan C = 100.0

# instantiate classifier with linear kernel and C=100.0

linear\_svc100=SVC(kernel='linear', C=100.0)

# fit classifier to training set

linear\_svc100.fit(X\_train, y\_train)

# make predictions on test set

y\_pred=linear\_svc100.predict(X\_test)

# compute and print accuracy score

print('Model accuracy score with linear kernel and C=100.0 : {0:0.4f}'. format(accuracy\_score(y\_test, y\_pred)))

|  |
| --- |
| output: |

Membandingkan akurasi train-set dan test\_set

y\_pred\_train = linear\_svc.predict(X\_train)

y\_pred\_train

|  |
| --- |
| output: |

print('Training-set accuracy score: {0:0.4f}'. format(accuracy\_score(y\_train, y\_pred\_train)))

|  |
| --- |
| output: |

Mengecek adanya overfitting dan underfitting

# print the scores on training and test set

print('Training set score: {:.4f}'.format(linear\_svc.score(X\_train, y\_train)))

print('Test set score: {:.4f}'.format(linear\_svc.score(X\_test, y\_test)))

|  |
| --- |
| output: |

Membandingkan akurasi model dan akurasi null

# check null accuracy score

null\_accuracy = (3306/(3306+274))

print('Null accuracy score: {0:0.4f}'. format(null\_accuracy))

|  |
| --- |
| output: |

**CEK LIST**

|  |  |  |  |
| --- | --- | --- | --- |
| Elemen Kompetensi | No Latihan | Penyelesaian | |
| Selesai | Tidak selesai |
| 1 | 1.1.1 |  |  |
| 1.1.2 |  |  |
| 1.2.1 |  |  |
| 1.2.2 |  |  |

**FORM UMPAN BALIK**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Elemen Kompetensi** | **Tingkat Kesulitan** | | | **Tingkat Ketertarikan** | | | **Waktu Penyelesaian dalam menit** |
| Mengimplementasikan SVR dalam kasus yang diberikan |  |  |  |  |  |  |  |
|  |  | Sangat Mudah |  |  | Tidak Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Mudah |  |  | Cukup Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | * Biasa |  |  | Tertarik | 70 |
|  |  |  |  |  |  |  |
|  |  | Sulit |  |  | * Sangat Tertarik |  |
|  |  |  |  |  |  |  |
|  |  | Sangat Sulit |  |  |  |  |
|  |  |  |  |  |  |  |