



**LAMPIRAN**

## Lampiran 1. Code SVM

```
from sklearn.feature_extraction.text import
TfidfVectorizer, CountVectorizer

from sklearn.preprocessing import LabelEncoder

from sklearn.model_selection import cross_val_score,
KFold

from sklearn import svm

import numpy as np

from sklearn.metrics import confusion_matrix,
ConfusionMatrixDisplay

import matplotlib.pyplot as plt

import pandas as pd

kf = KFold(n_splits=10, shuffle=True,
random_state=42)

acc_scores, pre_scores, rec_scores, f1_scores = [],
[], [], []

for train_index, test_index in kf.split(X):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = Y[train_index], Y[test_index]
```

```
SVM = svm.SVC(kernel='rbf') #Jika dengan
Kernel RBF

SVM = svm.SVC(kernel='linear') #Jika dengan
Kernel Linear

SVM.fit(X_train,y_train)

acc_score = SVM.score(X_test, y_test)

pre_score = np.mean(cross_val_score(SVM, X_train,
y_train, cv=kf, scoring='precision_macro'))

rec_score = np.mean(cross_val_score(SVM, X_train,
y_train, cv=kf, scoring='recall_macro'))

f1_score = np.mean(cross_val_score(SVM, X_train,
y_train, cv=kf, scoring='f1_macro'))

acc_scores.append(acc_score)
pre_scores.append(pre_score)
rec_scores.append(rec_score)
f1_scores.append(f1_score)

# Memprediksi kelas pada data uji
y_pred = SVM.predict(X_test)

# Menghitung confusion matrix
cm = confusion_matrix(y_test, y_pred)
```

```
print("Fold Confusion Matrix:")  
  
print(cm)  
  
# Menampilkan map confusion matrix  
  
disp =  
ConfusionMatrixDisplay(confusion_matrix=cm,  
display_labels=Encoder.classes_)  
  
disp.plot()  
  
plt.title("Confusion Matrix")  
  
plt.show()
```



## Lampiran 2. Code Naive Bayes

```
import numpy as np

import pandas as pd

from sklearn.feature_extraction.text import
CountVectorizer

from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import StratifiedKFold

from sklearn.metrics import confusion_matrix,
accuracy_score, precision_score, recall_score,
f1_score

import seaborn as sns

import matplotlib.pyplot as plt

# Bagi dataset menjadi fitur dan label
X = df['Text Tweet']
y = df['Sentiment']

# Inisialisasi CountVectorizer
vectorizer = CountVectorizer()

# Inisialisasi naive bayes classifier
nb_classifier = MultinomialNB()
```

```
# Inisialisasi variabel untuk menyimpan metrik
evaluasi

accuracy_scores = []

precision_scores = []

recall_scores = []

f1_scores = []

confusion_matrices = []

# K-fold cross validation
k = 5
skf = StratifiedKFold(n_splits=k, shuffle=True,
random_state=42)
fold = 1
for train_index, test_index in skf.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    # Preprocessing teks (mengubah teks menjadi
vektor)

    X_train_vectors =
vectorizer.fit_transform(X_train)

    X_test_vectors = vectorizer.transform(X_test)

# Melatih model Naive Bayes
```

```
nb_classifier.fit(X_train_vectors, y_train)

# Melakukan prediksi pada data uji
y_pred = nb_classifier.predict(X_test_vectors)

# Menghitung metrik evaluasi
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred,
average='weighted', zero_division=1)
recall = recall_score(y_test, y_pred,
average='weighted', zero_division=1)
f1 = f1_score(y_test, y_pred, average='weighted')

# Menghitung matriks confusion
cm = confusion_matrix(y_test, y_pred,
labels=['Negatif', 'Netral', 'Positif'])
confusion_matrices.append(cm)

# Menyimpan metrik evaluasi pada setiap iterasi
accuracy_scores.append(accuracy)
precision_scores.append(precision)
recall_scores.append(recall)
f1_scores.append(f1)
```

```
# Menyimpan data training dan testing dalam
format CSV

train_df = pd.DataFrame({'Text Tweet': X_train,
'Sentiment': y_train})

test_df = pd.DataFrame({'Text Tweet': X_test,
'Sentiment': y_test})

train_df.to_csv(f'training_fold{fold}.csv',
index=False)

test_df.to_csv(f'testing_fold{fold}.csv',
index=False)

fold += 1

# Menampilkan matriks confusion
print("Confusion Matrix:")
print(cm)
print()

# Menampilkan heatmap rata-rata matriks confusion
average_cm = np.mean(confusion_matrices, axis=0)
class_names = ['Negatif', 'Netral', 'Positif']
plt.figure(figsize=(8, 6))
sns.heatmap(average_cm, annot=True, fmt='g',
cmap='Blues', xticklabels=class_names,
yticklabels=class_names)
```



```
plt.xlabel('Prediksi')
plt.ylabel('Aktual')
plt.title('Confusion Matrix')
plt.show()

# Menampilkan metrik evaluasi rata-rata
print("hasil akurasi masing masing fold : %s" %
      (accuracy_scores))
print("Average Accuracy:", np.mean(accuracy_scores))
print("Average Precision:",
      np.mean(precision_scores))
print("Average Recall:", np.mean(recall_scores))
print("Average F1-score:", np.mean(f1_scores))
```



## RIWAYAT HIDUP

I Kadek Diksa Sukmadinata lahir di Bangli pada tanggal 4 september. Penulis lahir dari pasangan suami istri Bapak I Komang Wikerta dan Ibu Ni Kadek Sutarini. Penulis berkebangsaan Indonesia dan beragama Hindu. Kini penulis beralamat di Banjar Karuna Gunung Sari. Kecamatan Kintamani, Kabupaten Bangli, Provinsi Bali. Penulis menyelesaikan pendidikan dasar di SDN 3 Batur dan lulus pada 2013. Kemudian penulis melanjutkan di SMPN Negeri 1 Kintamani dan lulus pada tahun 2016. Pada tahun 2019, penulis lulus dari SMAN 1 Kintamani dengan jurusan MIPA. Penulis terdaftar sebagai mahasiswa Program Studi S1 Ilmu Komputer di Universitas Pendidikan Ganesha pada tahun 2019.

